Learning from Limited Labeled Data

Zero-Shot and Few-Shot Learning

A dissertation submitted towards the degree
Doctor of Engineering
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Yongqin Xian

Saarbrücken
2020

Day of Colloquium	7^{th} of July, 2020
Dean of the Faculty	Prof. Dr. Thomas Schuster Saarland University, Germany

Examination Committee

Chair	Prof. Dr. Antonio Krüger
Reviewer, Advisor	Prof. Dr. Zeynep Akata
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer	Prof. Trevor Darrell, Ph.D.
Reviewer	Prof. Barbara Caputo, Ph.D.
Academic Assistant	Dr. Paul Swoboda

ABSTRACT

Human beings have the remarkable ability to recognize novel visual concepts after observing only few or zero examples of them. Deep learning, however, often requires a large amount of labeled data to achieve a good performance. Labeled instances are expensive, difficult and even infeasible to obtain because the distribution of training instances among labels naturally exhibits a long tail. Therefore, it is of great interest to investigate how to learn efficiently from limited labeled data.

This thesis concerns an important subfield of learning from limited labeled data, namely, low-shot learning. The setting assumes the availability of many labeled examples from known classes and the goal is to learn novel classes from only a few (few-shot learning) or zero (zero-shot learning) training examples of them. To this end, we have developed a series of multi-modal learning approaches to facilitate the knowledge transfer from known classes to novel classes for a wide range of visual recognition tasks including image classification, semantic image segmentation and video action recognition.More specifically, this thesis mainly makes the following contributions. First, as there is no agreed upon zero-shot image classification benchmark, we define a new benchmark by unifying both the evaluation protocols and data splits of publicly available datasets. Second, in order to tackle the labeled data scarcity, we propose feature generation frameworks that synthesize data in the visual feature space for novel classes. Third, we extend zero-shot learning and few-shot learning to the semantic segmentation task and propose a challenging benchmark for it. We show that incorporating semantic information into a semantic segmentation network is effective in segmenting novel classes. Finally, we develop better video representation for the few-shot video classification task and leverage weakly-labeled videos by an efficient retrieval method.

ZUSAMMENFASSUNG

Menschen haben die bemerkenswerte Fähigkeit, neuartige visuelle Konzepte zu erkennen, nachdem sie nur wenige oder gar keine Beispiele davon beobachtet haben. Tiefes Lernen erfordert jedoch oft eine große Menge an beschrifteten Daten, um eine gute Leistung zu erzielen. Etikettierte Instanzen sind teuer, schwierig und sogar undurchführbar, weil die Verteilung der Trainingsinstanzen auf die Etiketten naturgemäß einen langen Schwanz aufweist. Daher ist es von großem Interesse zu untersuchen, wie man effizient aus begrenzten gelabelten Daten lernen kann.

Diese These betrifft einen wichtigen Teilbereich des Lernens aus begrenzt gelabelten Daten, nämlich das Low-Shot-Lernen. Das Setting setzt die Verfügbarkeit vieler gelabelter Beispiele aus bekannten Klassen voraus, und das Ziel ist es, neuartige Klassen aus nur wenigen (few-shot learning) oder null (zero-shot learning) Trainingsbeispielen davon zu lernen. Zu diesem Zweck haben wir eine Reihe von multimodalen Lernansätzen entwickelt, um den Wissenstransfer von bekannten Klassen zu neuartigen Klassen für ein breites Spektrum von visuellen Erkennungsaufgaben zu erleichtern, darunter Bildklassifizierung, semantische Bildsegmentierung und Videoaktionserkennung. Genauer gesagt, leistet diese Arbeit hauptsächlich die folgenden Beiträge. Da es keinen vereinbarten Benchmark für die Zero-Shot-Bildklassifikation gibt, definieren wir zunächst einen neuen Benchmark, indem wir sowohl die Evaluierungsprotokolle als auch die Datensplits öffentlich zugänglicher Datensätze vereinheitlichen. Zweitens schlagen wir zur Bewältigung der etikettierten Datenknappheit einen Rahmen für die Generierung von Merkmalen vor, der Daten im visuellen Merkmalsraum für neuartige Klassen synthetisiert. Drittens dehnen wir das Zero-Shot-Lernen und das few-Shot-Lernen auf die semantische Segmentierungsaufgabe aus und schlagen dafür einen anspruchsvollen Benchmark vor. Wir zeigen, dass die Einbindung semantischer Informationen in ein semantisches Segmentierungsnetz bei der Segmentierung neuartiger Klassen effektiv ist. Schließlich entwickeln wir eine bessere Videodarstellung für die Klassifizierungsaufgabe "few-shot video" und nutzen schwach markierte Videos durch eine effiziente Abrufmethode.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to Prof. Bernt Schiele and Prof. Zeynep Akata for supervising my PhD thesis. Both of them have been great advisors. I am grateful to Bernt for his constant supports and inspiration throughout the time. He has not only provided me invaluable advices in computer vision research, but also taught me how to be a good scientist as well as a good father by setting a role model himself. Likewise, I would like to thank Zeynep for guiding me to the wonderful journey of computer vision research. She has been extremely helpful because she gave me a lot of critical hands-on supervision and encouragement. None of my research presented in the thesis would be possible without her. I am fortunate and thankful for having both of them as my advisors.

I am also truly thankful to the other members in my dissertation committee. Thanks Prof. Trevor Darrell and Prof. Barbara Caputo for serving as external reviewers and attending my defense at those difficult times caused by the COVID-19 virus. Thanks Prof. Antonio Krüger for his quick responses and agreeing to chair the defense. Thanks Dr. Paul Swoboda for being the academic assistant. Their invaluable feedback and discussion on my thesis have helped and inspired me a lot.

I also would like to thank my lovely colleagues at MPII, not only for the inspiring discussion and collaboration concerning research but also for sharing a lot of happy moments outside of the work: Connie Balzert, Apratim Bhattacharyya, Rakshith Shetty, Philipp Müller, Eldar Insafutdinov, Anna Kukleva, Dr. Mykhaylo Andriluka, Dr. Gerard Pons-Moll, Prof. Wei-Chen Chiu, Dr. Anna Khoreva, Dr. Jan-Hendrik Lange, Dr. Wenbin Li, Prof. Siyu Tang, Prof. Shanshan Zhang, and Dr. Xucong Zhang. I owe particular thanks to Connie for helping me to handle a lot of difficult matters regarding my life in Germany. Thanks Jan-Hendrik and Philipp for helping me to translate many German letters into English. I also shared an office with Rakshith and we had many fruitful discussions about research. Thank you, Rakshith! It was a great pleasure to work with these talented people.

Furthermore, I would like to thank my collaborators, without whom I would have no chance to complete the thesis: Saurabh Sharma, Dr. Gaurav Sharma, Dr. Yang He, Prof. Matthias Hein, Dr. Quynh Nguyen Ngoc, Prof. Christoph H. Lampert, Prof. Lorenzo Torresani, Bruno Korbar and Dr. Matthijs Douze. I am particularly grateful to Lorenzo for supervising my internship at Facebook AI in Boston. Similarly, my thanks go to the students that I had the chance to supervise or work with: Yue Fan, Subhabrata Choudhury, Tobias Lorenz, Wenjia Xu and Miaoran Zhang.

Last but not the least, I am deeply thankful to my family and friends who have been constantly loving and supporting me. I would like to especially thank my wife Dr. Yijuan Qiao for her encouragement and sacrifice. My deepest gratitude also goes to my parents and brother who always love me without any condition. This thesis is dedicated to my beloved daughter Odelia who was born in the end of my PhD.

CONTENTS

1 INTRODUCTION

Contents

The demand for automated understanding of visual data (videos and images) has became more urgent than ever. Billions of images and videos uploaded on the internet demand autonomous analysis and understanding. Self-driving vehicles need a visual perception system to detect pedestrians, traffic signs and other obstacles. Hospitals need automated analysis of medical imaging data to improve the clinical efficiency. Robotics need to understand complex visual scenes for interacting with the environment.

In general, solving a computer vision task consists of two necessary steps: encoding and decoding. Given an image or video as input, the encoding step extracts features from the input and represents them as a compact vector. A lot of previous computer vision studies focus on designing hand-crafted features to encode an image or video. The decoding step extracts "patterns" from the feature vector and produces a decision depending on what the end task is. Machine learning is often applied in this step to learn the patterns in a principled way. Recent advances in computer vision are mainly due to the success of deep learning, which proposes to learn encoding and decoding simultaneously by a deep neural network optimized with task-specific losses. Despite the substantial progress, current computer vision algorithms still fail to generalize to the variety of visual environments in real-world applications.

A limitation of deep learning is that it requires massive amounts of labeled data to achieve high performance. However, labeled instances are expensive, difficult and even infeasible to obtain. As shown in Figure 1.1, in almost all scenarios, there is an exponential decay in terms of number of samples per class i.e., only a few classes contain a large number of samples whereas most classes are sparsely

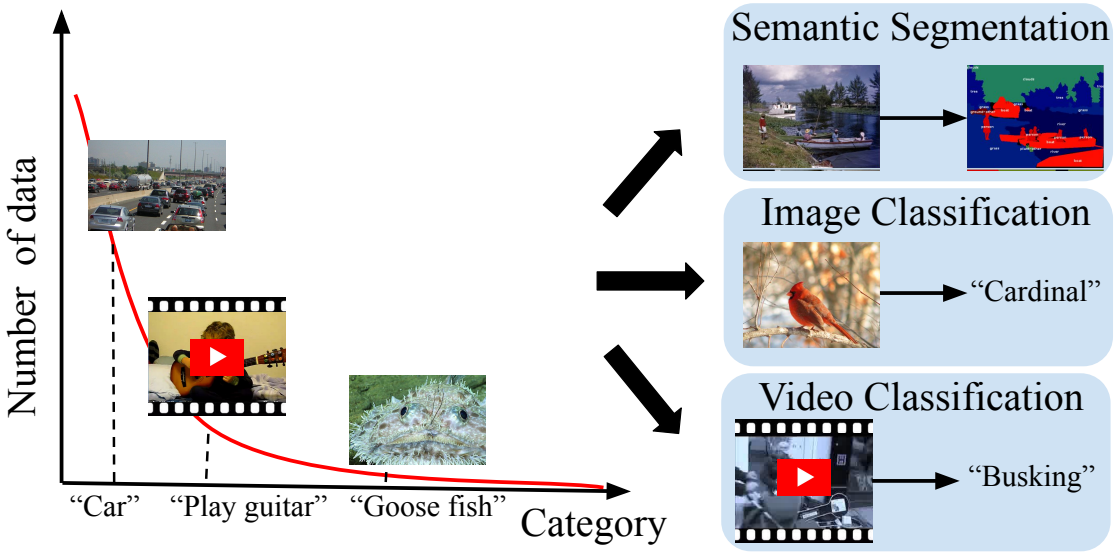

Figure 1.1: In almost all real-wold settings, the number of samples per category follows a skewed distribution i.e. a few categories have a large number of samples while most of categories have only a small number of samples (as shown in the left figure). The scarcity of samples results in poor generalization performance of the powerful deep learning methods which often require a huge number of labeled data to train. In this thesis, we address the challenges when learning with limited labeled data in the scenarios of image classification (e.g. He *et al.*, 2016), semantic segmentation (e.g. Long *et al.*, 2015) and video classification (e.g. Tran *et al.*, 2018).

populated. It becomes almost impossible to collect enough training examples for every class, leading to the inferior performance of deep neural networks. Consider a real-world example in the autonomous driving field. In order to train a reliable visual perception system for self-driving cars, current algorithms need to collect a vast amount of labeled examples that cover all the road condition, weather condition, time of driving, and obstacles. This is obviously infeasible because there are many circumstances that rarely occur e.g., big rocks on the snowy roads. As a consequence, the self-driving car is very likely to make wrong decisions when it encounters the rare circumstances. On the contrary, humans naturally possess the ability of learning novel concepts from a small number of examples. This is not only attributed to the computational power of the human brain, but also to its ability of re-using previous learned knowledge. Attaining such ability of rapid learning is particularly appealing for artificial intelligence (AI) and will push AI one step further towards human-level intelligence.

The goal of this thesis is thus to address the labeled data scarcity by developing machine learning methods that can be trained with limited labeled data. Our key idea is to re-use information from related tasks, transfer knowledge across different modalities , and leverage unlabeled data to minimize the human supervision on

novel tasks. More specifically, we aim to enable deep neural networks to generalize to novel concepts with as few labeled examples as possible. In order to mimic the way that human learns new concepts i.e., by re-using previous gained knowledge, we divide classes of interests into disjoint base and novel classes. Each of the base classes has enough training examples and plays a role as previous learned knowledge. On the contrary, the novel classes have only limited training examples and the task is to develop methods that generalize well to unseen examples from those novel classes. This thesis concerns both few-shot learning where each novel class possesses a few examples (up to 10 examples per class), and zero-shot learning where novel class has no labeled example at all. In this section, we will discuss the challenges in zero-shot image classification, few-shot image classification, and their applications in other computer vision tasks e.g., semantic segmentation and video action recognition. Finally, we summarize how this thesis contributes to the fields of zero-shot and few-shot learning.

1.1 CHALLENGES OF LEARNING FROM LIMITED LABELED DATA

Machine learning methods, typically deep neural networks, rely on a large labeled dataset for achieving a good performance, which makes it difficult to apply AI into real-world settings because collecting labeled data is not always possible (e.g., the skewed distribution for number of available samples in Figure 1.1). It is thus of great importance to develop machine learning methods that can learn from limited labeled data.

A fundamental problem of learning from a small dataset is the risk of overfitting i.e., a model fits too closely to the limited training examples and fails to generalize to unseen test samples. When the training data is limited, smart sampling of training data, regularization and data augmentation are three classical ways to improve the generalization performance according to the statistical learning theory (Bishop, 2006). While conventional machine learning methods draw training examples uniformly, smart sampling aims to select the "best" instances to reduce the amount of required training data. An example of smart sampling is active learning where the learning algorithms select the most uncertain samples to annotate given a fixed budget of labeling cost. Recent advancements in active learning show that deep learning models can be built with limited labeled data if training examples are smartly selected. However, active learning still requires a huge pool of data to select training examples. Another principled way to reduce overfitting is regularization, which refers to technics that prevent learning algorithms from fitting too closely to the training examples. Typical regularization techniques achieve this by reducing the model complexity e.g., L2 regularizer. For deep neural networks, popular regularizers include dropout that averages multiple models, pretraining on ImageNet that provides good initialization and early stopping of optimization that avoids fitting the noise in the dataset. In addition, data augmentation addresses the labeled data scarcity by automatically generating more training data without manually collecting

them. For visual understanding tasks of images, it has been shown that simple horizontal flipping and cropping of images can successfully increase the diversity of the training data and significantly improve the performance. Unfortunately, those simple techniques are still insufficient to obtain a good performance in the extreme case of lacking labeled data e.g., there is only 1 example per class.

In addition to those classical approaches, emerging directions for learning with limited labeled data include weakly supervised learning and self-supervised learning. Those directions do not directly tackle the overfitting issue on the small training set like the classical approaches. Instead, they aim to learn from a big dataset that is weakly annotated or not annotated such that human supervision is reduced. For example, (Oquab *et al.*, 2015) proposes an object detection approach with image-level labels, avoiding the expensive bounding box annotation. Self-supervised learning (Chen *et al.*, 2020) completely eliminate human supervision by learning from an unlabeled dataset.

In this thesis, we are mainly focusing on data augmentation and regularization approaches. Weakly supervised and self-supervised learning are promising directions to explore in the future but not the scope of this thesis. In the following subsections we identify the specific challenges of the tasks we want to solve and also discuss how we tackle those challenges in this thesis.

1.1.1 Zero-shot image classification

Zero-shot learning refers to the ability to predict novel classes without accessing any of their training examples. In the context of image classification, the task is to predict the class label of a given image from one of the novel classes. For simplicity, this thesis will only study the case where each image consists of only one object class. This problem can be highly valuable in the fine-grained classification where annotating labeled data requires expert knowledge. Here are a few challenges we aim to address in the thesis.

Multi-modal learning. In order to associate novel classes with base classes, we assume every class has some semantic information available e.g., attributes and textual description. Therefore, zero-shot learning is naturally a multi-modal learning problem. How to learn the correlation between two or even more modalities becomes a challenging research topic. Previous works (Akata *et al.*, 2015b, 2013) often learn a bilinear compatibility function which is limited to capture the complex correlation between vision and language modalities. The zero-shot learning performance will rely on the efficiency of knowledge transfer via multi-modal learning methods.

Limitation of current zero-shot benchmarks. Although the number of publication in zero-shot learning is steadily increasing, there is no agreed evaluation protocol, leading to incomparable results. In addition, novel classes in existing benchmarks are present in ImageNet which is used for feature pretraining, violating the principle of zero-shot learning. Finally, current benchmark only evaluates on novel classes and ignores base classes at the testing time, which is unrealistic. Real-world applications require the models to perform well on both base and novel

classes. There is an urgent demand for a better zero-shot learning benchmark.

Domain shift. Zero-shot learning models are trained on the examples of base classes and evaluated on novel classes without any training examples. Therefore, there is no generalization guarantee on novel classes because their distribution is totally unknown. Zero-shot learning can be particularly challenging if there is a domain gap between distributions of novel and base classes. How to solve the domain shift issue becomes an important challenge in zero-shot learning.

Extreme class imbalance Zero-shot learning suffers from the extreme case of data imbalance i.e., base classes have a lot of training examples and novel classes have no training data at all. Existing zero-shot methods essentially fail when evaluated on both base and novel classes because classifiers have a strong tendency to predict seen classes. One way to address this class imbalance problem is to employ a cost-sensitive loss (Chawla *et al.*, 2004) or over-sampling (Chawla *et al.*, 2002) the minority classes. However, these prior solutions are fundamentally not in line with deep learning and zero-shot learning methods.

1.1.2 Few-shot image classification.

In zero-shot learning, there is no training example for novel classes, which might be too extreme. In real-world scenarios, it is often more realistic to consider few-shot learning where a few labeled examples are available for novel classes. Despite those additional training data, few-shot learning remains to be a difficult task because the number of training examples is still far from enough to learn a deep neural network. In addition to the classical regularization techniques, how can we encourage the models to share knowledge across related tasks?

Risk of overfitting. Due to the small number of training examples from novel classes, directly fine-tuning a deep neural network will result in overfitting i.e., the model fits exactly to the small training set of novel classes and fails to generalize to unseen examples of novel classes. Techniques that work well in supervised learning will probably fail in the few-shot learning setting because of the overfitting. How to regularize the networks to avoid overfitting when fine-tuning the deep neural networks remains an open problem.

Imbalanced classes. In few-shot learning, the number of training examples from base classes is much larger than that of novel classes, resulting in an imbalanced learning problem. Many few-shot learning papers avoid this issue by ignoring the base classes at the evaluation time. However, we argue that such evaluation setting is unrealistic and consider the imbalanced issue as one of the challenges we would like to tackle.

Representation learning for few-shot learning. In the supervised learning setting, the goal is to learn a model that generalizes well to unseen examples from the same training task. The underlying assumption is that the distribution of test data follows that of training data. Its generalization error is guaranteed theoretically. However, few-shot learning aims for a model that generalizes well to novel tasks with a few training examples. Although conventional representation learning framework

works well for the known tasks, it might not generalize well to novel tasks. How to develop efficient representation for few-shot learning remains unknown. What principles make the representation generalize better to novel tasks?

1.1.3 Zero-shot and few-shot learning tasks beyond image classification

The long-tail issue does not only occur in the image classification tasks but also in other computer vision tasks. In this thesis, we additionally study the semantic segmentation and video classification tasks in the context of zero-shot and few-shot learning.

Semantic segmentation. The image semantic segmentation task aims to predict a class label for every pixel in the image. This is a challenging structural output learning problem and requires expensive pixel-level labeling. Ordinary semantic segmentation methods fail to handle the images which contain novel classes. In order to tackle the long-tail issue, this thesis is interested in a semantic segmentation frame that can make zero-shot prediction on novel classes and few-shot learning on novel classes with limited labeled data. Since this is a new task, we face the challenge of how to formally define the problem. In addition, how to transfer knowledge from known classes to novel classes is another challenge as well.

Video classification. The task of the video classification is to assign an action class label to a trimmed video. The few-shot learning setting becomes practical in the video domain because annotating videos is more time-consuming and the class distribution is also skewed. In addition to learn the spatial information, we have to model temporal information which is particularly critical for some video applications. A common challenge in few-shot video learning as well as in ordinary video learning is how to learn representation that encodes both temporal and spatial information. In addition, the overfitting risk becomes higher comparing to the few-shot image classification task because the video models often have larger capacity than the image models.

1.2 CONTRIBUTIONS OF THE THESIS

In this section, we summarize the contributions of this thesis in three different fields.

1.2.1 Contributions to zero-shot image classification

To tackle the multi-modal learning challenges of zero-shot learning, we propose a novel compatibility learning framework by incorporating latent variables in the compatibility function. Instead of learning a single bilinear function like previous works, we propose to learn a collection of bilinear models while allowing each image-class pair to choose from them. This effectively makes our model non-linear, as in different local regions of the space the decision boundary, while being linear, is different. In addition, we propose a fast and effective method for model selection by

successive pruning of an over-complete initialization. We show that such a strategy is competitive compared to standard cross-validation based model selection, while being much faster to train. We extensively evaluate our novel piece-wise linear model for zero-shot and generalized zero-shot learning settings on various aspects such as stability, interpretability, generalizability to seen and unseen classes.

We define a new benchmark by unifying both the evaluation protocols and data splits of publicly available datasets used for this task. This is an important contribution as published results are often not comparable and sometimes even flawed due to, e.g. pre-training on zero-shot test classes. Our evaluation protocol emphasizes the necessity of tuning hyperparameters of the methods on a validation class split that is disjoint from training classes as improving zero-shot learning performance via tuning parameters on test classes violates the zero-shot assumption. We point out that extracting image features via a pre-trained deep neural network (DNN) on a large dataset that contains zero-shot test classes also violates the zero-shot learning idea as image feature extraction is a part of the training procedure. We recommend to abstract away from the restricted nature of zero-shot evaluation and make the task more practical by including training classes in the search space, i.e. generalized zero-shot learning setting. Moreover, we propose a new zero-shot learning dataset, the Animals with Attributes 2 (AWA2) dataset which we make publicly available both in terms of image features and the images themselves. We systematically evaluate zero-shot learning across a significant number of datasets and methods. The crux of the matter for all zero-shot learning methods is to associate observed and non observed classes through some form of auxiliary information which encodes visually distinguishing properties of objects. We thoroughly evaluate zero-shot learning approaches, by using multiple splits of several small, medium and large-scale datasets (Patterson and Hays, 2012; Welinder *et al.*, 2010; Lampert *et al.*, 2013; Farhadi *et al.*, 2009; Deng *et al.*, 2009). Therefore, we argue that our work plays an important role in advancing the zero-shot learning field by analyzing the good and bad aspects of the zero-shot learning task as well as proposing ways to eliminate the ugly ones.

Our benchmark paper demonstrates that almost all the zero-shot methods fail in the generalized zero-shot learning setting where the model has to predict both base and novel classes. In order to tackle the imbalance challenge in this setting, we propose a novel conditional generative model f-CLSWGAN that synthesizes CNN features of novel classes from their semantic embeddings. Once trained, the feature generator will be able to synthesize arbitrarily many features for any class which lacks training examples. We show that data generation in the feature space works much better than in the image space because generating realistic images from semantic embeddings is a much harder task. Across five datasets with varying granularity and sizes, we consistently improve upon the state of the art in both the ZSL and GZSL settings. We demonstrate a practical application for adversarial training and propose GZSL as a proxy task to evaluate the performance of generative models. Our model is generalizable to different deep CNN features, e.g., extracted from GoogleNet or ResNet, and may use different class-level auxiliary information,

e.g., sentence, attribute, and word2vec embeddings.

1.2.2 Contributions to few-shot image classification

The success of our feature generation approach encourages us to extend it to the few-shot learning setting, which also suffers from the imbalance issue. To this end, we propose the f-VAEGAN-D2 model that consists of a conditional encoder, a shared conditional decoder/generator, a conditional discriminator and a non-conditional discriminator. The first three networks aim to learn the conditional distribution of CNN image features given class embeddings optimizing VAE and WGAN losses on labeled data of seen classes. The last network learns the marginal distribution of CNN image features on the unlabeled features of novel classes. Once trained, our model synthesizes discriminative image features that can be used to augment softmax classifier training. Our empirical analysis on CUB, AWA2, SUN, FLO, and large-scale ImageNet shows that our generated features improve the state-of-the-art in low-shot regimes, i.e., (generalized) zero- and few shot learning in both the inductive and transductive settings. We demonstrate that our generated features are interpretable by inverting them back to the raw pixel space and by generating visual explanations.

1.2.3 Contributions to zero-shot and few-shot tasks beyond image classification

We introduce novel (generalized) zero-label and few-label semantic image segmentation tasks in a realistic settings inspired by zero-shot learning for image classification. In zero-label semantic segmentation (ZLSS), our aim is to segment previously unseen, i.e. novel, classes, in few-label semantic segmentation (FLSS) these novel classes have a small number of labeled training examples. In this work, we also aim for learning without forgetting the previously seen classes, i.e. generalized ZLSS and FLSS. To this end, we propose semantic projection network (SPNet), an end-to-end semantic segmentation model which maps each image pixel to a semantic word embedding space where it is projected with a fixed word embedding to class probabilities optimizing the cross-entropy loss. We create a benchmark for (generalized) zero- and few-label semantic image segmentation with two challenging datasets, i.e. COCO-Stuff and PASCAL-VOC. Our analysis shows that the SPNet model achieves impressive results both quantitatively and qualitatively in (generalized) zero-label and few-label tasks. Furthermore, as a side-product, our model improves the state of the art in zero-shot image classification demonstrating that it successfully generalizes to other tasks.

We push the progress of few-shot video classification in three aspects: 1) To learn the temporal information, we revisit spatiotemporal CNNs in the few-shot video classification regime. We develop a 3D CNN baseline that maintains significant temporal information within short clips; 2) We propose to retrieve relevant tag-labeled videos from a large video dataset, i.e. YFCC100M, to circumvent the need for class-labeled

videos of novel classes; 3) We extend current few-shot video classification evaluation by introducing two challenging experimental settings. In generalized few-shot video classification task, the search space has no restriction in terms of classes. In few-shot video classification with more ways, the search space goes beyond five towards all classes. Our extensive experimental results demonstrate that on existing settings spatiotemporal CNNs outperform the state-of-the-art by a large margin, and on our proposed settings weakly-labeled videos retrieved using tags successfully tackles both of our new few-shot video classification tasks.

1.3 OUTLINE OF THE THESIS

In this section, we provide an overview of the thesis by briefly summarizing each chapter and draw a connection between them. We also note the respective publications and collaborations with other researchers.

Chapter 2: Related work. This chapter surveys related work which tackles challenges of learning with limited labeled data with a focus on the three directions of the thesis i.e., zero-shot image classification, few-shot image classification and zero- and few-shot tasks beyond image classification. We discuss how these works relate to the approaches and contributions presented in this thesis. A discussion of related work specific to the following chapters is provided within each chapter.

Chapter 3: Latent Embedding for Zero-Shot Image Classification. In this chapter, we tackle the zero-shot image classification problem by developing a novel compatibility function that learns non-linear relationship between the image and semantic class embedding spaces.

The content of this chapter is an extension of Yongqin Xian's Master Thesis, which was published in CVPR 2016 with the title *Latent Embedding for Zero-Shot Image Classification* (Xian *et al.*, 2016). The following significant changes have been made in our extension: comparing with four other SOTA methods, evaluating in generalized zero-shot and few-shot settings, and combining multiple class embeddings for better performance. Yongqin Xian was the lead author of this paper. It is a collaboration with Gaurav Sharma, and the Machine Learning Group of Saarland University.

Chapter 4: Zero-Shot Learning: the Good, the Bad and the Ugly. In this chapter, we show that existing zero-shot learning evaluation protocols adopted by Chapter 3 and other works are limited. Therefore, we introduce a new zero-shot learning benchmark which resolves the issues of previous protocols. Our new benchmark involves 5 datasets and includes both zero-shot learning setting that only predicts novel classes and generalized zero-shot learning which predicts both base and novel classes. We provide a better summarization of existing approaches by classifying them into groups and evaluating them under the unified evaluation protocol.

The content of this chapter was published in TPAMI 2019 with the title *zero-shot learning - a comprehensive evaluation of the good the bad and the ugly* (Xian *et al.*, 2019b), which is an extension of our CVPR 2017 publication *Zero-Shot Learning-the Good, the Bad and the Ugly* (Xian *et al.*, 2017). Yongqin Xian was the lead author of both papers. It is also a collaboration with Christoph Lampert from IST Austria.

Chapter 5: Feature Generating Networks for Zero-Shot Image Classification. In this chapter, we tackle the issues we observe in Chapter 4. More specifically, we found that almost all the zero-shot learning approaches fail to achieve good performance on novel classes in the generalized zero-shot learning setting due to the extreme imbalanced dataset. To this end, we propose a novel generative model that synthesizes visual features for novel classes from their semantic class embeddings. The generative model is learned on base class data and can be used to synthesize arbitrarily many visual features for novel classes, alleviating the data imbalance issue.

The content of this chapter corresponds to the CVPR 2018 publication *Feature Generating Networks for Zero-Shot Learning* (Xian *et al.*, 2018). Yongqin Xian was the lead author of this paper, while Tobias Lorenz contributed the image generation part. Tobias Lorenz's bachelor thesis at MPI Informatics was co-supervised by Yongqin Xian and Bernt Schiele.

Chapter 6: Enhanced Feature Generation Frameworks for Low-Shot Learning. Based on the success of feature generation technique described in Chapter 5 on zero-shot learning tasks, we improve the generative model in Chapter 5 in two aspects. First, we combine GANs and VAE to obtain a stronger generative model that attains the strength of adversarial and non-adversarial learning. Second, we additionally add a discriminator that learns the marginal distribution of novel classes when their unlabeled data is available. We also propose to interpret generated features by inverting them back into the image pixel space.

The content of this chapter corresponds to the CVPR 2019 publication *f-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning* (Xian *et al.*, 2019c). Yongqin Xian was the lead author of this paper while Saurab Sharma contributed the feature explanation part.

Chapter 7: Zero-Label and Few-Label Semantic Segmentation. Previous chapters are all about image classification. In this chapter, we introduce a novel image semantic segmentation task that aims to segment novel classes that have zero or very few training examples. We propose an approach called SPNet that projects each pixel into a semantic embedding space such that knowledge can be transferred from base classes to novel classes. We show that our method can tackle both zero-label and few-label semantic segmentation tasks.

The content of this chapter corresponds to the CVPR 2019 publication *Sematic Projection Network for Zero-Label and Few-Label Semantic Segmentation* (Xian *et al.*,

2019a). Yongqin and Subhabrata Choudhury were the first co-authors of this paper. Yongqin Xian contributed to the main ideas, zero-shot image classification experiments, and writting of the paper. Subhabrata Choudhury implemented the approach and conducted most of the experiments. It is also a collaboration with Yang He.

Chapter 8: Generalized Many-Way Few-Shot Video Classification. In this chapter, we shift from image classification tasks to the video classification task which predict the action label of each video in the context of few-shot learning. We show that a simple linear classifier baseline with 3D CNNs as the backbone surpasses existing few-shot video classification benchmark. Therefore we propose a more realistic and challenging evaluation setting called generalized few-shot video classification involving more classes. We develop an efficient retrieval-based few-shot learning approach that leverages weakly-labeled videos from a large-scale video dataset.

The content of this chapter is still under review for a conference by the time of submitting this thesis. The lead author of this project was Yongqin Xian. This is his internship project done at Facebook AI together with Lorenzo Torresani, Bruno Korbar and Matthijs Douze

Chapter 9: Conclusions and future perspectives. This chapter concludes the thesis by summarizing the contributions and highlighting their current limitations and possible directions to overcome them. We provide an outlook on our ongoing and future work and discuss future directions for the field.

RELATED WORK 2

Contents

The field of learning with limited labeled data covers a wide range of topics including semi-supervised learning, unsupervised learning, self-supervised learning, weakly-supervised learning, few-shot learning and zero-shot learning. This thesis will mainly focus on few-shot and zero-shot learning tasks. In this chapter, we formally define the research problems chosen in this thesis. We present the most relevant and recent developments in the fields and relate them to the contributions of this thesis in the conclusion of each section. The following chapters also discuss related work, but targeted to the respective topic of the respective chapter.

2.1 ZERO-SHOT IMAGE CLASSIFICATION

The ability of predicting previously unseen classes, called zero-shot learning, is an extreme case of learning with limited labeled data. In object recognition or image classification, the task of zero-shot learning is to predict the label of an image belonging to one of novel object classes that do not appear during training time. The only available information on novel classes is the semantic information that describes those classes. Humans are able to predict unseen objects by combining their prior knowledge and textual description of novel classes. For instance, given an image of Scarlet Tanager (we probably have never seen before), we will have a high chance to make a correct prediction after reading the textual description of Scarlet Tanager. Inspired by the human brains, zero-shot object recognition can be addressed by

performing multi-modal learning from both image and semantic information. In the following, we will first formally define zero-shot learning. Different modalities of data and evaluation protocols will be discussed next. Then we will try to give an overview of existing zero-shot learning approaches by grouping them. Finally, the relationship between this thesis and existing works will be discussed.

2.1.1 Problem definition

Let $\mathcal{T} = \{(\mathbf{x}, \mathbf{y})|\mathbf{x}, \mathbf{y} \in \mathcal{S}\}$ be the training set where $\mathbf{x}$ denotes an image instance and $\mathbf{y}$ is its class label belonging to one of seen classes $\mathcal{S}$. We are interested in predicting a disjoint set of classes $\mathcal{U}$ ($\mathcal{S} \cap \mathcal{U} = \emptyset$), called unseen classes, without any observed examples. Clearly, this task can not be solved without any information of unseen classes. So additionally, we assume some auxiliary information, e.g., textual description, about each class i.e. seen and unseen classes, is provided to allow knowledge transfer from the seen classes to unseen classes.

2.1.1.1 *Image embedding*

For a visual recognition task, one of the most important components is to extract features from images. The image feature is in the form of a vector in some arbitrary feature space and should ideally capture discriminative characteristics of an image i.e., shape, color, texture etc. The features are then fed into machine learning algorithms to learn classifiers that distinguish between different objects. In this thesis, we call the image features as image embedding. Formally, we define the image embedding of a given image $\mathbf{x}$ as $\phi(\mathbf{x})$ where $\phi(\bullet)$ is a function that maps an image $\mathbf{x}$ to a $d_{\mathbf{x}}$-dimensional feature space. Before the success of deep learning, image features are often manually designed by computer vision researchers. There have been a lot of studies on how to build robust image features or descriptors manually. Deep learning takes a brave new perspective to learn image representation together with the end task from a big amount of training data. Deep image representation quickly revolutionized the fields and become the standard way to extract image feature. Next, I will briefly review this two groups of image features.

Hand-crafted image representation. Typical hand-crafted image features aggregate some image descriptors extracted from local image regions, which is obtained by interest region detection algorithms e.g., Harris-affine detector(Mikolajczyk and Schmid, 2004). A simple image descriptor is the histogram of pixel intensities. In order to achieve the illumination invariant, (Zabih and Woodfill, 1994) have proposed to use histograms of ordering and reciprocal relations between pixel intensities. A more widely used image descriptor is the scale invariant feature transform (SIFT)(Lowe, 1999), which computes a gradient histogram over local regions obtained by a scale invariant region detector. (Bay *et al.*, 2008) further proposes the speeded up robust features (SURF) which is stronger and faster than the SIFT. A comprehensive review of image descriptors can be found in (Mikolajczyk and Schmid, 2005). A popular way

to aggregate image descriptors extracted from local image regions is Bag-of-visual-words (BOV) which assigns each descriptor to the closest visual vocabulary obtained by k-means clustering (Arandjelovic and Zisserman, 2013). (Sánchez *et al.*, 2013) proposes Fisher Vector that extends BOV to use a Gaussian mixture model. BOV ignores the spatial relationship between image patches, therefore, Spatial Pyramid Matching (Yang *et al.*, 2009) was proposed to address this issue.

Deep image representation In contrast to aforementioned hand-crafted image features that adopt a manually designed extraction pipeline, deep image representation directly learns the image embedding function $\phi(\bullet)$ via a deep convolutional neural network (CNN or ConvNet) (LeCun *et al.*, 2015). A simple example of neural network is the multi-layer perceptron (MLP) which stacks multiple fully connected (FC) layers with a non-linear operation e.g. ReLU, after each layer. The FC layers connect each neuron in current layer to all the neurons in the next layer with different learnable weights. This is obviously prone to overfit because of such huge number of model parameters. Therefore, ConvNet regularizes the neural network by considering only local connection of neurons and sharing weight parameters across different local neighborhood. Such regularizer can be efficiently implemented by the convolution operation. The first convolutional neural network architecture , called LeNet(LeCun *et al.*, 1989), was introduced by Yann Lecun. A 5-Layer LeNet architecture follows *CONV-POOL-CONV-POOL-FC-FC* where *CONV* represents the convolutional layer followed by a non-linear function, *POOL* is the max pooling that subsamples the feature maps, and *FC* is the fully connected layer. AlexNet (Krizhevsky *et al.*, 2012) improves LeNet by stacking more *CONV* layers without pooling and won the ImageNet ILSVRC challenge in 2012. GoogLeNet (Szegedy *et al.*, 2015) introduces the inception module and replaces FC layers with the global average pooling, dramatically reducing the number of parameters compared to AlexNet. VGG (Simonyan and Zisserman, 2014b) shows that depth of the network plays an important role for good performance. Current popular CNN architecture is ResNet which introduces skip-connection and makes the network as deep as 152 layers. There are also a few extensions of ResNet proposed like DenseNet (Huang *et al.*, 2019), ResNeXt, etc. Recently, Neural Architecture Search(Zoph and Le, 2016), which aims to learn the network architecture automatically, has obtained increasing attention. The CNN networks are often learned with the backpropagation algorithm with a task specific loss such as the cross-entropy loss for multi-class image classification. The object function of learning CNN is non-convex because of its highly non-linear structure. But empirically, SGD-based algorithms are sufficient for a good performance. Theoretical studies about the optimization of CNN can be found in (Nguyen *et al.*, 2019).

2.1.1.2 *Class embedding*

Zero-shot image classification is a multi-modal learning problem where image examples of unseen classes are not available and learning of unseen classes relies

on another modality of data. This modality often comes from some high-level semantic information such as human annotated attributes or text descriptions. The semantic information is usually assumed to be in the class level. Therefore, we call it class embedding. One can consider the class embedding as the prototype that represents the abstract of a class. The class embedding plays an important role in the zero-shot learning image classification. Good class embeddings should capture visual similarities between classes. One can refer to (Akata *et al.*, 2015b) for a comprehensive evaluation of different class embeddings in zero-shot learning. In this section, we discuss four different class embeddings that are widely used in zero-shot learning.

Attribute Attributes describe the visual properties of an object, such as "red" , "spotted" or "striped". The appearance of an object class can often be represented by combinations of different colors, shapes, and patterns. Therefore, they are useful cues to recognize objects. Most importantly, attributes are shared among objects such that knowledge learned from seen classes can be transferred to unseen classes. In order to annotate attributes, we have to first define attribute vocabularies that are discriminative enough to distinguish the object classes of our interests. For instance, on the Caltech-UCSD Birds-200-2011 Dataset (CUB), a vocabulary of 312 binary attributes e.g., *eye color yellow*, *beak shape sharp*, was selected based on an online tool for bird species identification[1]. Then each bird image is annotated with those 312 binary attributes i.e., check if this attribute appear in the image or not, with Mechanical Turk. Such annotation provides image-level attributes, while class embedding is defined for each class. Class embeddings are often produced by averaging image-level attributes of the images belonging to each class.

Word embedding Attribute provides accurate visual properties of objects, but it requires expensive manual annotation. An alternative to avoid annotation is the word embedding, which is a technique that maps each word from a vocabulary to a vector of real numbers. This mapping can be learned with a neural network in an unsupervised way on large text corpus e.g., Wikipedia. Popular word embeddings include word2vec(Mikolov *et al.*, 2013a), glove(Pennington *et al.*, 2014), fasttext(Joulin *et al.*, 2016a), etc. Word2vec is a language model parameterized with a neural network. In its continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. By learning the word co-occurrence, the resulting word embedding captures semantic similarities between different words i.e., word embeddings of semantically related words are close in the embedding space. For zero-shot learning, we employ the word embeddings of class names as their class embeddings. Such strategy is inexpensive, but word embeddings often lead to poor zero-shot learning results because they often do not reflect visual similarities between classes. Therefore, there are some works that try to inject visual information into word embeddings. Moreover, one word could represent multiple meanings such that its word embedding is ambiguous. Bert provides a solution for

[1] https://www.whatbird.com/

that by incorporating context to word embeddings.

Class hierarchy Object categories are naturally in a hierarchical structure. For instance, "albatross" and "crow" are subordinate of "bird" which is again subordinate of "animal". Such class hierarchy provides relatedness between object classes as well. WordNet is a database of English words and it defines such a hierarchy where words are linked together by their semantic relationships in a tree structure. Standard neural network cannot be directly applied to the class hierarchy as the tree structure is not Euclidean data. In order to use the class hierarchy for zero-shot learning, we could either derive word embedding for each node or directly apply graph convolution on top of the class hierarchy.

Text description The word embeddings of class names are often insufficient to describe a class category because they are trained on noisy text corpus. As we discussed before, we prefer class embeddings that could capture visual similarities between classes. This motivates us to consider annotating text description for images. More specifically, for each image, we could write several sentences to describe the visual content in the image. The class embedding can then be learned via a language model i.e., LSTM.

2.1.2 Evaluation protocol

In contrast to the supervised image classification where the model is trained and evaluated on the same label space, zero-shot learning methods should be trained and evaluated on different label spaces. Therefore we have to first define disjoint class sets for training and testing respectively. The data split is usually generated within one dataset i.e., classes of a dataset are divided into two disjoint sets i.e., seen classes for training and unseen classes for testing. Next we produce a training set including images of all the seen classes and a test set including hold-out images of the unseen classes. If we are interested in seen classes at the test time, the test set should also include hold-out images of the seen classes. In this section, we will only discuss several existing zero-shot learning evaluation protocol in a high-level. Details of the protocols will be introduced in Section .

Lampert *et al.* (2013) introduce the first evaluation protocol for zero-shot image classification. The authors propose a dataset called AWA consisting of 50 classes in total. Those classes are randomly split into 40 seen and 10 unseen classes. A model is trained on the images of seen class and evaluated on unseen classes with the top-1 classification accuracy. Rohrbach *et al.* (2012) define another zero-shot data split on the ImageNet where they split 1000 classes into 800 seen and 200 unseen classes. Elhoseiny *et al.* introduce zero-shot splits on CUB (Welinder *et al.*, 2010) and Oxford Flowers (Nilsback and Zisserman, 2008) datasets. Classes of CUB are randomly split into 160 seen and 40 unseen classes on CUB, while Oxford flowers are divided into 82 seen and 10 unseen classes. Akata *et al.* (2013) introduce another data split on CUB with 150 seen and 50 unseen classes. Besides, Socher *et al.* (2013) generate a

zero-shot split on CIFAR10. Finally, Lampert *et al.* (2013) extends their work into a journal by extending their evaluation on SUN (Xiao *et al.*, 2010) and aPY (Farhadi *et al.*, 2009).

2.1.3 A literature review of zero-shot approaches

Zero-shot learning has attracted increasing attention since the first paper published by (Lampert *et al.*, 2013). Given such a big number of zero-shot learning publications, it is difficult to discuss all of them. Instead, we summarize popular zero-shot learning approaches published in top conferences or journals by grouping them into five categories i.e., Attribute-based methods, compatibility learning, generative models, direct classifier prediction, transductive zero-shot learning. Chapter of this thesis describes our survey paper about zero-shot learning where we discuss many zero-shot learning works. This section is complementary to that by introducing additional reference and more recent papers.

Attribute-based methods Early works tackle zero-shot learning by first solving the attribute prediction problem. Attribute predictions are then aggregated to make a prediction on unseen classes. To this end, Lampert *et al.* (2013) proposes direct attribute prediction and indirect attribute prediction methods. Jayaraman and Grauman (2014) argue that annotated attributes are not always and adopt a random forest to address this issue. Al-Halah *et al.* (2016) propose to predict the attribute class embedding of unseen classes without manual annotation.

Compatibility learning Instead of learning attribute classifiers, compatibility learning frameworks directly learn a compatibility function that measures the similarity between two modalities i.e., image embedding and class embedding. Because of its efficiency and flexibility, many recent works follow this direction. ALE (Akata *et al.*, 2013) and CONSE (Norouzi *et al.*, 2014) learn linear compatibility function with the ranking loss. Similarly, SJE (Akata *et al.*, 2015b) adopts the multi-class max-margin loss. ESZSL (Romera-Paredes *et al.*, 2015) proposes a loss that has a closed-form solution. Semantic autoencoder (Kodirov *et al.*, 2017) for zero-shot learning regularizes the model by auto-encoder loss. Zhang *et al.* (2017b) argue that semantic embedding space has hubness problem and propose to learn a non-linear embedding function that maps the semantic embedding into the image embedding space. Recently, Ji *et al.* (2018b) propose to learn feature representation with attention conditioned on the semantic embedding. Similarly, Xie *et al.* (2019) propose to learn attention on local regions for more generalized representation.

Generative models The aforementioned methods are discriminative approaches where they directly model the posterior probability distribution of labels given the input i.e., $p(\mathbf{y}|\mathbf{x})$. Generative approaches instead model the joit distribution of input and output i.e., $p(\mathbf{x}, \mathbf{y})$. An advantage of generative model is that arbitrarily many samples can be synthesizing for unseen classes, addressing the issues of lacking data.

Verma and Rai (2017) assume $p(\mathbf{x}|\mathbf{y})$ to be Gaussian distribution. Kumar Verma *et al.* (2018a) learn to synthesize features of unseen classes via a VAE. Similarly, Zhu *et al.* (2018a) proposes a GAN framework to generate features from noisy text descriptions. Both Schonfeld *et al.* (2019) and Mishra *et al.* (2018) learn a VAE to generate features. Felix *et al.* (2018b) use cycle-consistency loss to regularize the GANs.

Direct classifier prediction Instead of synthesizing samples, SYNC (Changpinyo *et al.*, 2016) proposes to directly synthesize the classifier weights of unseen classes. Elhoseiny *et al.* (2013) take a similar approach with textual description as the class embedding. Changpinyo *et al.* (2017) apply kernel methods to synthesize the visual prototype of unseen classes. Lei Ba *et al.* (2015) apply a neural network to predict the classifier weights of unseen classes. Wang *et al.* (2018a) leverage the class hierarchy and learn to regress classifier weights of unseen classes with a graph convolutional neural network. Kampffmeyer *et al.* (2019) extend Wang *et al.* (2018a) by constructing a better graph.

Transductive zero-shot learning Conventional zero-shot learning setting is often inductive i.e., images of unseen classes are not available during training. In the real-world scenario, it is possible that unlabeled images from unseen classes are available and we aim to label them. This motivates us to study the transductive learning setting where labeled images from seen classes and unlabeled images from unseen classes are available. Fu *et al.* (2014) construct a graph with both labeled and unlabeled images and performs label propagation. Kodirov *et al.* (2015) leverage the unlabeled data to reduce the domain gap between seen and unseen classes. In order to address the biased prediction towards seen classes, Song *et al.* (2018) propose to minimize the probability of predicting unseen class images as seen classes. Liu *et al.* (2018) introduce a neural network that calibrates the predicted probabilities with unlabeled images from unseen classes.

2.1.4 Relations to our work

In Chapter 1, we introduce a novel compatibility learning framework for zero-shot learning. In contrast to previous works that learn a linear compatibility function, we propose to learn a non-linear function by learning multiple linear transformations with the selection of which transformation to use being a latent variable.

In Chapter 2, we take a step back and analyze the status quo of the area. We find that there exist inconsistent evaluation protocols for zero-shot learning and some of them are even flawed, leading to incomparable or incorrect results. Therefore, the main purpose of our work is to define an unified evaluation protocol for zero-shot learning and re-evaluate existing approaches under the same protocol to show the true progress of the field. Our benchmark is built on (Lampert *et al.*, 2013), but we extend its evaluation protocol to cover more datasets and the more realistic generalized zero-shot learning setting where the model has to predict both seen and unseen classes. Our work is also inspired by **?**. where they empirically show the

challenges of generalized zero-shot learning. But the main contribution of our work is not only to advocate the generalized zero-shot learning, but also to introduce a unified zero-shot learning benchmark for future research.

In Chapter 3, in order to tackle generalized zero-shot learning, we propose to generate visual features of unseen classes conditioned on class embeddings. There are two concurrent works that share similar ideas with us. Bucher *et al.* (2017) adopt a GMMN (Li *et al.*, 2015) to generate feature and Mishra *et al.* (2018) apply a VAE (Kingma and Welling, 2014). Our paper takes the powerful GANs (e.g. Goodfellow *et al.*, 2014; Arjovsky and Bottou, 2017; Arjovsky *et al.*, 2017) and improve it by including a classification loss that enforces generated features can be better suited for the classification task. In addition, our work shows that our generated feature can be applied to improve many popular zero-shot methods, which is more generalizable. There have been a group of papers which follows our ideas and improve the feature generation process by regularizing the generators, proposing more complicated generative networks, and using different class embeddings.

In Chapter 4, we extend our feature generating networks in Chapter 3 to any-shot and transductive learning settings. We improves our f-CLSWGAN by combining VAE (e.g. Kingma and Welling, 2014) and GANs (e.g. Goodfellow *et al.*, 2014; Arjovsky and Bottou, 2017; Arjovsky *et al.*, 2017), leveraging the strength of adversarial and non-adversarial generative models. In order to learn from unlabeled data, we propose to add an additional discriminator for learning the marginal probability distribution of unseen classes. Previous transductive zero-shot learning (e.g. Fu *et al.*, 2014; Kodirov *et al.*, 2015) is often solved by the label propagation technique. Our approach improves the feature generator by modeling the marginal distribution of unlabeled images. Besides, comparing to other feature generating papers (e.g. Kumar Verma *et al.*, 2018a; Zhu *et al.*, 2018a; Schonfeld *et al.*, 2019; Felix *et al.*, 2018b), our proposed framework is more flexible and can be applied to solve inductive zero-shot learning where there is no image from unseen classes, transductive zero-shot learning where unlabeled images from unseen classes are available, and few-shot learning where there are a few images per unseen classes.

2.2 FEW-SHOT IMAGE CLASSIFICATION

In general, few-shot learning aims to learn a model e.g., deep neural network, with limited labeled data. Learning a deep neural network from scratch with a small amount of data is not possible because of its massive number of model parameters. Therefore, few-shot learning setting assumes the availability of some base classes which have enough labeled data. The task becomes how we learn a model from those base classes such that it generalizes well to novel classes with only few labeled data. This is an important problem to solve because the numbers of labeled data per category follow a long-tail distribution i.e., there are a small number of classes with a lot of data while most of classes have limited training data. In this section, we first formally define the few-shot image classification problem and introduce the existing

evaluation protocols. Then we discuss popular few-shot approaches in Section 2.2.3 and the relations between those and our proposed approaches.

2.2.1 Problem definition

Let $\mathcal{T}_b = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{X}, \mathbf{y} \in C_b\}$ be a labeled training set for base classes where $\mathbf{x}$ denotes an image instance in the RGB image space $\mathcal{X}$ and $\mathbf{y}$ is its class label belonging to one of base classes C_b. Each base class has enough training data (typically larger than 30 images). We are interested in a disjoint set of classes C_n ($C_n \cap C_b = \emptyset$), called novel classes. Similarly, we define its training set as $\mathcal{T}_n = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{X}, \mathbf{y} \in C_n\}$. In contrast to base classes, we assume each novel class consists of only few training data (usually less than 10 images). Therefore, the size of the training set of base classes is much larger than that of the novel classes i.e., $|\mathcal{T}_b| \gg |\mathcal{T}_n|$. Given training sets $\mathcal{T}_b$ and $\mathcal{T}_n$, the task of few-shot learning is to learn a model that generalizes well to the hold-out test set of novel classes C_n.

2.2.2 Evaluation protocols

In order to evaluate few-shot learning approaches, the first step is to produce a data split that consists of a training set $\mathcal{T}_b$ of base classes and a training set $\mathcal{T}_n$ of novel classes. However, there exist multiple different protocols that define how to evaluate few-shot learning approaches on the novel classes. Most of papers focus on the constrained meta-learning setting, while some papers also follows the low-shot setting which is relatively more realistic. Here we will mainly discuss the most popular three protocols i.e., low-shot learning setting, meta-learning setting and improved meta-learning setting.

Low-shot learning setting. In this setting, all the novel classes and base classes are evaluated simultaneously. Qi *et al.* (2018) introduce a data split on CUB where 100 classes are base and the rest 100 classes are novel. For a k-shot learning problem, they randomly draw k samples per novel class to form the training set $\mathcal{T}_n$ where $k \in \{1, 2, 5, 10, 20\}$. The performance is then evaluated on the hold-out test set of the novel classes. To make it more realistic, they also evaluate on all classes including both base and novel classes. In this case, there will be a hold-out test set for base and novel classes respectively. The top-1 image classification accuracy will be reported. CUB is a relatively small-scale and fine-grained dataset with only 10K images. To evaluate few-shot approaches in a large-scale setting, Hariharan and Girshick (2017) propose a low-shot data split on the ImageNet. The 1000 ImgeNet classes are divided into 389 base categories and 611 novel categories. For the purpose of cross-validation, they further construct two disjoint sets of classes by dividing the base categories into two subsets C_b^1 (193 classes) and C_b^2 (196 classes) and the novel categories into C_n^1 (300 classes) and C_n^2 (311 classes). While C_b^1 and C_n^1 are used for tuning hyperparameters, the final results are reported on C_b^2 and C_n^2 for k-shot problems where $k \in \{1, 2, 5, 10, 20\}$. Finally, our f-VAEGAN-D2 extends the

zero-shot splits into few-shot splits by randomly drawing k examples from each unseen class to form the training set $\mathcal{T}_n$.

Meta-learning setting. The meta-learning setting (e.g. Vinyals *et al.*, 2016; Snell *et al.*, 2017; Finn *et al.*, 2017) has gained increasing attention recently. Instead of treating all the novel classes as a big task, this setting generates many small tasks by randomly sampling subsets from the novel classes. More specifically, the evaluation is conducted in the episodic manner where each episode constructs a k-shot, n-way classification task with a training set $\mathcal{T}_n$ and a test set. The final results are obtained by averaging the test accuracy over multiple episodes. Existing papers mainly consider the following four tasks: 1-shot 5-way, 5-shot 5-way, 1-shot 20-way, and 5-shot 20-way. Matching Networks (Vinyals *et al.*, 2016) introduce the meta-learning setting and propose data splits on the Omniglot and the miniImageNet datasets.

Improved meta-learning setting. Triantafillou *et al.* (2019) argues that current meta-learning benchmarks (e.g. Vinyals *et al.*, 2016; Snell *et al.*, 2017; Finn *et al.*, 2017) do not have sufficient complexity to access the few-shot learning process. Therefore, they propose the meta-dataset, a new large-scale, benchmark that is more realistic. Meta-dataset improves current meta-learning setting in three aspects: 1) evaluate the cross-dataset generalization performance with 10 different datasets 2) vary the number of classes and examples per class 3) consider the relationships between classes when forming episodes.

2.2.3 A literature review of few-shot approaches

Few-shot learning is challenging because novel classes have limited labeled data. Directly fine-tuning a deep CNN on the novel classes will inevitably lead to overfitting. On the other hand, due to the domain gap between base and novel classes, directly applying the pretrained model would suffer from domain shift issues. A group of papers investigate ways that efficiently adapt a model pretrained on base classes to novel classes with only a few training examples. In this case, few-shot learning problem is treated as a transfer learning problem. This direction is usually evaluated in the low-shot learning setting. In addition, there are also a significant number of papers that propose novel training strategies that learn fast from few labeled examples. In this senario, the meta-learning setting is adopted to evaluate the performance.

2.2.3.1 *Low-shot learning.*

Low-shot learning approaches mainly focus on how to adapt a pretrained model to novel classes without finetuning the whole deep neural network. Qi *et al.* (2018) propose to normalize the classifier weights and directly produce the weights of novel classes by averaging the image their image embeddings. Qiao *et al.* (2018) learn a MLP that regresses classifier weights from its training samples. Wang *et al.* (2019a) rely on class embedding to generate task-aware feature embedding. Chen *et al.* (2019)

aim to reduce intra-class variations by adopting cosine distance on learned classifier weights. On the other hand, synthesizing data has been a classical way to address the small data problem. In the scenario of few-shot learning, it is natural to investigate how we generate synthetic data for novel classes. Therefore Hariharan and Girshick (2017) propose to generate features from a data point and predefined transformation. Wang *et al.* (2018c) extend this idea by meta-learning the feature generator.

2.2.3.2 *Meta-learning approaches.*

This field is also called learning to learn. The main idea is to learn a "learning algorithm" that can learn from few examples. One can think the "learning algorithm" as a function that takes input as a training set and outputs the classifiers. They (e.g. Vinyals *et al.*, 2016; Snell *et al.*, 2017) argue that it is beneficial to mimic the few-shot learning scenario on base classes. Therefore, the episode learning scheme is applied on the base class training as well. More specifically, in every training episode, a support set of k-shot, n-way classification problem and a query set including test samples of n classes are sampled. Multi-class classifiers are constructed from the support set (by a "learning algorithm") and then evaluated on the query set to compute the loss. Matching networks (Vinyals *et al.*, 2016) meta-learns weighted neareast neighbor classifiers. Prototypical networks (Snell *et al.*, 2017) meta-learns the class prototype and adopt the nearest neighbor classifier as well. Ravi and Larochelle (2016) parameterize the optimization algorithm (SGD) as a LSTM and meta-learns how to optimize the objective function. MAML (Finn *et al.*, 2017) proposes to learn how to initialize the network such that the optimization only takes few steps. Sung *et al.* (2018) meta-learn a siameses network that predict similarities of two images. Triantafillou *et al.* (2017) define a training objective that optimizes over all relative orderings of the batch points simultaneously.

2.2.4 Relations to our work

In Chapter 4, we propose a unified feature generation framework that works both for zero-shot and few-shot learning. Although our method shares similar idea with other feature generation papers (e.g. Hariharan and Girshick, 2017; Wang *et al.*, 2018c), our feature generator is quite different from existing papers. While hallucinate paper (e.g. Hariharan and Girshick, 2017; Wang *et al.*, 2018c) only generate features from image data, our approaches learns a multi-modal feature generator that synthesizes features from semantic embeddings, which allows better knowledge transfer. In addition, our framework can be applied to the transductive learning setting when the unlabeled examples from novel classes are available. Therefore, our method is more versatile.

2.3 ZERO-SHOT AND FEW-SHOT TASKS BEYOND IMAGE CLASSIFICATION

Most of zero-shot and few-shot learning papers focus on the image classification problem. However, the limitation of labeled data arises in almost all the computer vision tasks, e.g. semantic segmention (e.g. Long *et al.*, 2015; Zhang *et al.*, 2018a; Caesar *et al.*, 2016), obeject detection (e.g. Girshick, 2015; He *et al.*, 2017; Redmon *et al.*, 2016), video action recognition (e.g. Karpathy *et al.*, 2014; Feichtenhofer *et al.*, 2016b), 3D vision (e.g. Riegler *et al.*, 2017; Qi *et al.*, 2017), etc.

Although those tasks are as important as the image classification, they are relatively unexplored. While the image classification task is a good starting point to study the zero-shot and few-shot learning problems, it is not always true that few-shot or zero-shot technics for image classification can be directly applied to other vision tasks, for instance, semantic segmentation and video classification. 3D reconstruction is naturally a few-shot problem because it is difficult to acquire 3D data. Wallace and Hariharan (2019) propose a novel method that leverages category-specific priors for few-shot single-image 3D reconstruction problem. For object detection tasks, Bansal *et al.* (2018) introduce an approach that can localises novel categories in an image. Kang *et al.* (2019) proposes a feature reweighting technic to address the few-shot object detection task. This section will mainly discuss the applications of zero-shot and few-shot learning in the context of semantic image segmentation and video action recognition.

2.3.1 Semantic image segmentation

In contrast to the image classification task which predicts a single label for an entire image, the goal of semantic image segmentation is to assign a class label for each pixel in an image. Popular semantic segmentation methods include FCN (Long *et al.*, 2015), deeplab (Chen *et al.*, 2018), and U-Net (Ronneberger *et al.*, 2015). Learning those models often requires pixel-wise annotations which are expensive and hard to obtain. In order to reduce the annotation efforts, weakly supervised learning with bounding box annotation (Khoreva *et al.*, 2017) has been proposed. We are interested in an orthogonal direction that learns from only a few examples, avoiding collecting and annotating data. The main idea behind that is few-shot learning that aims to achieve generalization on novel classes with only a few examples. The extreme case of few-shot learning is zero-shot learning where novel classes have no example at all. In this section, we will introduce some papers that tackle few-shot and zero-shot semantic segmentation problems.

Rakelly *et al.* (2018) proposes a novel conditional FCN (fully convolutional network) learned by the end-to-end optimization. The network takes an annotated support set of images as conditions and performs inference on an unannotated query image. Dong and Xing (2018) propose to learn class prototypes via metric learning. Shaban *et al.* (2017) introduce a two-branched approach to address the one-shot

semantic image segmentation. While the first branch generates parameters from an image, the second branch takes both these parameters and a new image as input and produces a segmentation mask of the image for the new class as output. In the extreme zero-shot learning case, there is no training images for novel classes. Instead, the models rely on semantic class embedding to transfer knowledge from base to novel classes. Zhao *et al.* (2017a) propose to learn a joint embedding function between visual features per pixel and word2vec embedding per class. Bucher *et al.* (2019) extend the feature generation idea to image semantic segmentation.

2.3.2 Video action recognition

Video understanding is another important field in computer vision. It is challenging because the model has to learn the temporal information in addition to the spatial context. Typical video understanding tasks include video action recognition (e.g. Feichtenhofer *et al.*, 2016b, 2017), video captioning (e.g. Gao *et al.*, 2017), self-driving cars (e.g. Geiger *et al.*, 2012), robotics (e.g. Kemp *et al.*, 2007) etc. While the ResNet (He *et al.*, 2016) has been the widely used image representation network, there is no such "ResNet" in video domain. Representation learning for videos is still an open problem. Similarly, few-shot learning in the context of video understanding is unexplored. In this thesis, we mainly focus on the video action recognition which predicts a single label for a trimmed video. Xu *et al.* (2015) propose a zero-shot action recognition approach that constructs a mapping from video feature space to the semantic class embedding space. Zhu and Yang (2018) adopt a memory network that stores multiple prototypes for each class. Cao *et al.* (2019) propose to learn temporal information by solving an video frames alignment problem.

2.3.3 Relations to our work

In Chapter 5, we introduce a semantic projection network (SPNet) that handles both zero-label and few-label semantic segmentation tasks. While Zhao *et al.* (2017a) propose open-vocabulary scene parsing task that segments novel objects by performing hierarchical parsing, we leverage word embeddings to predict the exact unseen classes and address the few-label problem in a unified framework. For few-shot semantic segmentation, previous approaches (e.g. Shaban *et al.*, 2017; Dong and Xing, 2018) follow the meta-learning setup (e.g. Vinyals *et al.*, 2016; Snell *et al.*, 2017), which uses a support set to predict an query image. However, those approaches are restricted to output a binary mask and fail to segment an image with multiple classes. In contrast, our approach is operating in the more realistic (generalized) few-label semantic segmentation setting, i.e. pixel-level labeling of an image where labels come from both base and novel classes.

In Chapter 6, we propose a strong model based on 3D CNNs for few-shot video action recognition and introduce more challenging evaluation settings for future research. Comparing to previous approaches (e.g. Zhu and Yang, 2018; Cao *et al.*,

2019) which extract frame-level features, our model extract clip-level features via 3D CNNs such that temporal information is better captured. In addition, our evaluation is more challenging and realistic than previous ones. We observe that our model saturates previous evaluation settings and therefore introduce more challenging many-way few-shot learning and generalized few-shot learning settings for future research.

3 LATENT EMBEDDING FOR ZERO-SHOT IMAGE CLASSIFICATION

Contents

In this chapter, we present an approach for learning a compatibility function between image and class embedding spaces for image classification when labeled training data is scarce. The proposed method augments the state-of-the-art bilinear compatibility methods (e.g. Akata *et al.*, 2015a,b; Frome *et al.*, 2013) by by incorporating latent variables. Instead of learning a single bilinear map, our novel latent embedding model learns a collection of bilinear maps with the selection of which map to use being a latent variable for the current image-class pair. We empirically demonstrate the strength of our model with respect to six state-of-the-art models (e.g. Akata *et al.*, 2015b; Romera-Paredes *et al.*, 2015; Zhang and Saligrama, 2015; Socher *et al.*, 2013; Zhang and Saligrama, 2016) on three challenging datasets i.e. AWA (Lampert *et al.*, 2013), CUB (Welinder *et al.*, 2010) and Dogs (Khosla *et al.*) using four different class embeddings. In addition to zero-shot learning experiments, we provide an extensive analysis of our method on few-shots and generalized zero-shot learning settings.

This chapter takes the first step towards the few-shot learning and more realistic generalized zero-shot learning setting. In Chapter 4, we evaluate the approaches introduced in this chapter as well as other SOTA approaches under the same evaluation protocol. In Chapter 5, we show that feature generation is an effective way to address generalized zero-shot learning. In Chapter 6, we demonstrate that unlabeled data improves the feature generation, leading to significantly better any-shot learning performance i.e., zero-shot and few-shot learning.

3.1 INTRODUCTION

Humans are highly capable of recognizing novel object categories using some form of external information, without seeing any actual visual example of that category. Enabling computers with this capability has been recently introduced as *zero-shot learning* task in the intersection of computer vision and machine learning. Zero-shot learning (e.g. Bart and Ullman, 2005; Palatucci *et al.*, 2009; Lampert *et al.*, 2013; Larochelle *et al.*, 2008; Yu and Aloimonos, 2010) has been formally posed as follows: labeled images are provided for certain visual classes during training and the task is to learn a model that can make predictions for novel classes at test time. As training and test class sets are disjoint, namely there are no visual examples are provided for some classes during training, the standard supervised image classification frameworks that use class labels cannot be employed. Although object class labels are not available, a list of attributes (e.g. Ferrari and Zisserman, 2007; Farhadi *et al.*, 2009; Lampert *et al.*, 2013), a set of easily recognizable properties of objects such as furry, spotted etc. provide a structured relationships between class labels that facilitates the required induction.

Substantial progress has been made for zero-shot learning task (e.g. Duan *et al.*, 2012; Farhadi *et al.*, 2010; Ferrari and Zisserman, 2007; Kankuekul *et al.*, 2012; Lampert *et al.*, 2013; Parikh and Grauman, 2011; Papadopoulos *et al.*, 2014; Akata *et al.*, 2015c). This progress can be attributed to two recent advances. First, representation learning using deep neural networks (e.g. Krizhevsky *et al.*, 2012; Szegedy *et al.*, 2015) provides image embeddings which perform well across a range of visual classification tasks (e.g. Razavian *et al.*, 2014). Second, multi-modal structured embedding frameworks (e.g. Akata *et al.*, 2015a,c; Frome *et al.*, 2013; Romera-Paredes *et al.*, 2015) provide a means to measure the compatibility between image and class representations. While noting the parallel progress in image representations, i.e. via deep neural networks (He *et al.*, 2016), in this work, we focus on improving the compatibility learning framework.

Compatibility learning frameworks (e.g. Akata *et al.*, 2015a,c; Frome *et al.*, 2013; Hastie *et al.*, 2008; Palatucci *et al.*, 2009; Romera-Paredes *et al.*, 2015; Socher *et al.*, 2013; Xian *et al.*, 2016; Fu and Sigal, 2016; Qiao *et al.*, 2016; Akata *et al.*, 2016; Bucher *et al.*, 2016; Mensink *et al.*, 2014; Fu *et al.*, 2015b; Kodirov *et al.*, 2015) are generally based on the idea of representing both the images *and the classes* in (respective) multi-dimensional vector spaces. Image embeddings are obtained from state-of-the-art image representations e.g. those from deep convolutional neural networks (e.g. Krizhevsky *et al.*, 2012; Szegedy *et al.*, 2015). Class embeddings can be obtained using manually specified side information e.g. attributes (Lampert *et al.*, 2013), extracted automatically from an large but unlabeled large text corpora (e.g. Mikolov *et al.*, 2013b; Pennington *et al.*, 2014) etc. A compatibility function is then learned with a discriminative objective that decreases the distance, in the embedded space, between images from the same class while increasing that between images from different classes. Once learned, such a compatibility function can be used to predict the class (more precisely, the class embedding) of any given image. The predicted

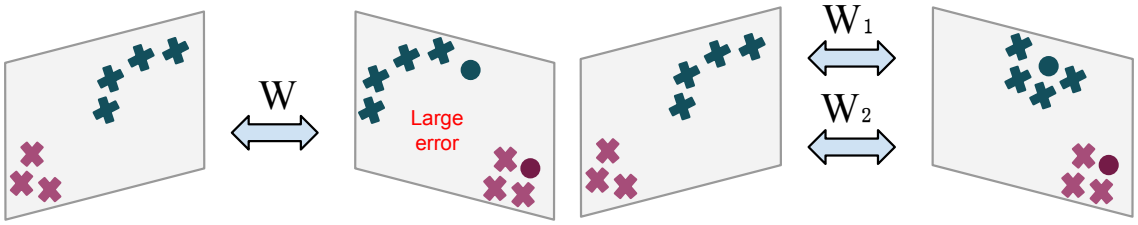

Figure 3.1: Compatibility learning frameworks that use a linear projection, e.g. SJE Akata *et al.* (2015c) (figure on the left) may lead to a large projection error, however learning a piece-wise linear model (figure on the right) leads to more precise projections. Here, crosses represent image embeddings and their projections on the class embedding space, W are the parameters of the compatibility function, solid circles represent the ground truth class embedding.

embedding vector might not correspond to a known class label. Therefore in practice, the nearest embedding corresponding to a class label is taken as the class prediction. Advantageously, this can then be done for images belonging to both seen and unseen classes, hence enabling zero-shot classification.

State-of-the-art compatibility learning frameworks for zero-shot learning (e.g. Akata *et al.*, 2015a,c; Frome *et al.*, 2013; Romera-Paredes *et al.*, 2015) use a linear compatibility function to learn the model. However, learning a linear compatibility function is not sufficient for the challenging fine-grained classification problem. A model that can automatically group objects with similar properties together and then learn different compatibility models, adapted for different groups, is expected to perform better for fine-grained classification. For instance, two different linear functions that separate blue birds with brown wings and from other blue birds with blue wings can be learned separately. With such motivation, we propose a novel model for zero-shot classification which incorporates latent variables to learn a piecewise linear compatibility function between image and class embeddings. The approach is inspired by many recent advances in visual recognition that utilize latent variable models, e.g. object detection (e.g. Felzenszwalb *et al.*, 2010; Hussain and Triggs, 2010), human pose estimation (Yang and Ramanan, 2011) and face detection (Zhu and Ramanan, 2012).

Our contributions are as follows. First, we propose a novel method for zero-shot learning. By incorporating latent variables in the compatibility function our method achieves factorization over such (possibly complex combinations of) variations in pose, appearance and other factors. Instead of learning a single linear function, we propose to learn a collection of linear models while allowing each image-class pair to choose from them. This effectively makes our model non-linear, as in different local regions of the space the decision boundary, while being linear, is different. We use an efficient stochastic gradient descent (SGD) based learning method. Second, we propose a fast and effective method for model selection by successive pruning of an over-complete initialization. We show that such a strategy is competitive compared to standard cross-validation based model selection, while being much

faster to train. Third, we evaluate our novel piece-wise linear model for zero-shot and generalized zero-shot learning setting with various class embeddings (e.g. Mikolov *et al.*, 2013b; Pennington *et al.*, 2014; Miller, 1995) on three challenging datasets, i.e. Caltech-UCSD Birds 200-2011 (CUB) (Welinder *et al.*, 2010), Animals With Attributes (AWA) (Lampert *et al.*, 2013) and Stanford Dogs [2] (Dogs) (Khosla *et al.*). We compare our method on all these configurations with several related existing embedding methods. We show that incorporating latent variables in the compatibility learning framework consistently improves the state-of-the-art for zero-shot learning setting. Fourth, we extensively evaluate our novel piecewise linear model for zero-shot and generalized zero-shot learning settings on various aspects such as stability, interpretability, generalizability to seen and unseen classes. We raise awareness for the challenge of transferring information from zero-shot setting to full multi-class setting and aim to inspire further research in this direction.

In section 4.2, we present an extensive discussion of related work. In section 3.2 we give details of the bilinear compatibility learning framework that our method is based on. In section 4.3 we present our novel Latent Embedding framework which extends the bilinear compatibility learning framework to nonlinearity through learning several piece-wise linear models that each capture a different latent aspect of the data. In section 3.4 we evaluate our Latent Embedding framework with respect to several criteria both on zero-shot and on generalized zero-shot learning settings. In section 3.5 we conclude with a discussion and potential future directions.

3.2 BACKGROUND: BILINEAR JOINT EMBEDDINGS

In this section, we describe the bilinear joint embedding framework (e.g. Akata *et al.*, 2015c,a; Weston *et al.*, 2011), on which we build our Latent Embedding Model that will be detailed in section 4.3.

We work in a supervised setting where we are given an annotated training set

$$\mathcal{T} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{d_x}, \mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^{d_y}\}, \tag{3.1}$$

where $\mathbf{x}$ is the image embedding defined in an image feature space $\mathcal{X}$, e.g. CNN features (Krizhevsky *et al.*, 2012), and $\mathbf{y}$ is the class embedding defined in a label space $\mathcal{Y}$ that models the conceptual relationships between classes, e.g. attributes (e.g. Farhadi *et al.*, 2009; Lampert *et al.*, 2013). The goal is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ to predict the correct class for the query images. In previous work (e.g. Weston *et al.*, 2011; Akata *et al.*, 2015a,c), this is done via learning a function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that measures the compatibility between a given input embedding $\mathbf{x} \in \mathcal{X}$ and an output embedding $\mathbf{y} \in \mathcal{Y}$. The prediction function then chooses the class with the maximum compatibility, i.e.

$$f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}). \tag{3.2}$$

[2] We use the 113 class subset of the Stanford Dogs dataset as in (Akata *et al.*, 2015c)

In general, the class embeddings reflect the common and distinguishing properties of different classes using side-information that is extracted independently of images e.g. attributes of classes. Using these embeddings, the compatibility can be computed even with those unknown classes which have no corresponding images in the training set. Therefore, this framework can be applied to zero-shot learning (e.g. Akata *et al.*, 2015a,c; Palatucci *et al.*, 2009; Romera-Paredes *et al.*, 2015; Socher *et al.*, 2013). In previous work, the compatibility function takes a simple form,

$$F(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top W \mathbf{y} \tag{3.3}$$

with the matrix $W \in \mathbb{R}^{d_x \times d_y}$ being the parameter to be learnt from training data. Due to the bilinearity of F in $\mathbf{x}$ and $\mathbf{y}$, previous workg (e.g. Akata *et al.*, 2015a,c; Weston *et al.*, 2011) refer to this model as a bilinear model, however one can also view it as a linear one since F is linear in the parameter W. In the following, these two terminologies will be used interchangeably depending on the context.

3.3 LATENT EMBEDDINGS MODEL (LATEM)

In general, the linearity of the compatibility function in Equation 3.3 is a limitation as the problem of image classification is usually a complex nonlinear decision problem. Linear decision functions can be extended to nonlinear ones through the use of piecewise linear decision functions. Achieving non-linearity through piece-wise linearity has been used successfully in various models for solving computer vision tasks such as mixture of templates (Hussain and Triggs, 2010) and deformable parts-based model (Felzenszwalb *et al.*, 2010) for object detection, mixture of parts for pose estimation (Yang and Ramanan, 2011) and face detection (Zhu and Ramanan, 2012). The main idea in most of such models, along with modeling parts, is that of incorporating latent variables, e.g. the different templates in the mixture of templates Hussain and Triggs (2010) and the different 'components' in the deformable parts model (Felzenszwalb *et al.*, 2010). Therefore, the model becomes a collection of linear models. The test images then pick one of these linear models, with the selection being latent and image specific. Intuitively, this factorizes the decision function into components which focus on distinctive 'clusters' in the data, e.g. one component may focus on the profile view while another on the frontal view of the object. Incorporating nonlinearity in this way has been shown (e.g. Felzenszwalb *et al.*, 2010; Hussain and Triggs, 2010; Yang and Ramanan, 2011; Zhu and Ramanan, 2012) to improve performance.

In the following subsections, we will detail our novel LatEm model that extends bilinear joint embedding model to nonlinearity through a piece-wise linear formulation. We discuss our optimization algorithm, model selection and finalize with a discussion.

3.3.1 Objective

We propose to construct a nonlinear, albeit piecewise linear, compatibility function. Parallel to the latent SVM formulation, we propose a non-linear compatibility function as follows.

$$F(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq K} \tilde{\mathbf{w}}_i^\top (\mathbf{x} \otimes \mathbf{y}), \tag{3.4}$$

where $i = 1, \ldots, K$, with $K \geq 2$, indexes over the latent choices and $\tilde{\mathbf{w}}_i \in \mathbb{R}^{d_x d_y}$ are the parameters of the individual linear components of the model. This equation can be reformulated as a mixture of bilinear compatibility functions (Equation 3.3),

$$F(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq K} \mathbf{x}^\top W_i \mathbf{y}. \tag{3.5}$$

Our goal here is to learn the set of parameters $\{W_i\}$ of the above compatibility funtion that minimizes the empirical risk given as

$$\frac{1}{N} \sum_{n=1}^{|\mathcal{T}|} L(\mathbf{x}_n, \mathbf{y}_n). \tag{3.6}$$

where $L : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is the loss function defined for a particular example $(\mathbf{x}_n, \mathbf{y}_n)$ as

$$L(\mathbf{x}_n, \mathbf{y}_n) = \sum_{\mathbf{y} \in \mathcal{Y}} \left[\Delta(\mathbf{y}_n, \mathbf{y}) + F(\mathbf{x}_n, \mathbf{y}) - F(\mathbf{x}_n, \mathbf{y}_n) \right]_+, \tag{3.7}$$

with $\Delta(\mathbf{y}_n, \mathbf{y})$ being the zero-one loss defined as,

$$\Delta(\mathbf{y}, \mathbf{y}_n) = \begin{cases} 1 & \text{if } \mathbf{y} \neq \mathbf{y}_n \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

and $[a]_+ = \max(0, a)$ bounds the Equation 3.6 from above. This ranking-based loss function has been previously used in Akata *et al.* (2015a); Frome *et al.* (2013); Weston *et al.* (2011) such that the model is trained to produce a higher compatibility between the matching image and class embedding than the mismatching image and class embedding. Note that by setting $K = 1$, our LatEm framework generalizes to bilinear joint embedding framework as each of the W_i leads to a bilinear compatibility defined in Equation 3.3, while the full compatibility function becomes nonlinear owing to the max operator.

3.3.2 Optimization

Even though F is convex, we first observe that the ranking loss function L from Equation 3.7 is not jointly convex in all the W_i's. Thus, finding a globally optimal solution, which was practical due to convexity in the previous linear models (e.g. Akata *et al.*, 2015a,c), is difficult now. To minimize the empirical risk in Equation 3.6,

Algorithm 1 SGD optimization for LatEm

$\mathcal{T} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathbb{R}^{d_x}, \mathbf{y} \in \mathbb{R}^{d_y}\}$

1: **for all** $t = 1$ to T **do**
2: **for all** $n = 1$ to $|\mathcal{T}|$ **do**
3: Draw $(\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{T}$ and $\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_n\}$
4: **if** $F(\mathbf{x}_n, \mathbf{y}) + 1 > F(\mathbf{x}_n, \mathbf{y}_n)$ **then**
5: $i^* \leftarrow \underset{1 \leq k \leq K}{\operatorname{argmax}} \mathbf{x}_n^\top W_k \mathbf{y}$
6: $j^* \leftarrow \underset{1 \leq k \leq K}{\operatorname{argmax}} \mathbf{x}_n^\top W_k \mathbf{y}_n$
7: **if** $i^* = j^*$ **then**
8: $W_{i^*}^{t+1} \leftarrow W_{i^*}^t - \eta_t \mathbf{x}_n (\mathbf{y} - \mathbf{y}_n)^\top$
9: **end if**
10: **if** $i^* \neq j^*$ **then**
11: $W_{i^*}^{t+1} \leftarrow W_{i^*}^t - \eta_t \mathbf{x}_n \mathbf{y}^\top$
12: $W_{j^*}^{t+1} \leftarrow W_{j^*}^t + \eta_t \mathbf{x}_n \mathbf{y}_n^\top$
13: **end if**
14: **end if**
15: **end for**
16: **end for**

we propose a simple SGD-based method that works in the same fashion as in the convex setting. Our LatEm method, while possibly leading to only local minima, performs well in practice as shown in section 3.4.

The details of the SGD optimization of our LatEm method (Algorithm 1) are as follows. Given a training set $\mathcal{T} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathbb{R}^{d_x}, \mathbf{y} \in \mathbb{R}^{d_y}\}$ of image embeddings, i.e. x and their associated class embeddings, i.e. y, we loop through all our samples for a certain number of epochs T. For each sample $(\mathbf{x}_n, \mathbf{y}_n)$ in the training set, we randomly select a $\mathbf{y}$ that is different from $\mathbf{y}_n$ (step 3 of Algorithm 1). If the randomly selected $\mathbf{y}$ violates the margin condition (step 4 in Algorithm 1), then we update the W_i matrices following the steps $5 - 13$ in Algorithm 1. In particular, we find the W_i that leads to the maximum score for $\mathbf{y}$ (step 5) and the W_j that gives the maximum score for $\mathbf{y}$ (step 6). If the same matrix gives the maximum score, the condition on step 7 in Algorithm 1 has been satisfied so we update that matrix. If two different matrices lead to the maximum score which corresponds to the condition formulated on step 9 in Algorithm 1, we update both matrices, i.e. W_{i^*} and W_{j^*} using the sub-gradient based updates formulated on steps 11 and 12.

3.3.3 Model selection

The number of matrices K in the model is a free parameter. We use two strategies to select the number of matrices. As the first method, we use a standard cross-validation strategy, i.e. we split the dataset randomly into disjoint parts (in a zero-shot setup)

and choose the K with the best cross-validation performance. We denote this strategy as CV in the following sections. While this is a well established strategy which we find to work well in practice, we also propose a pruning based strategy which is competitive while being faster to train. In pruning based strategy, we start with a relatively large number of matrices and prune them as follows. As the training proceeds, each sampled training examples chooses one of the matrices for scoring – we keep track of this information and build a histogram over the number of matrices counting how many times each matrix was chosen by any training example. In particular, this is done by increasing the counter for W_{j^*} by 1 after step 6 of Algorithm 1. With this information, after five passes over the training data, we prune out the matrices which were chosen by less than 5% of the training examples, so far. This is based on the intuition that if a matrix is being chosen only by a very small number of examples, it is probably not critical for performance. With this model pruning approach we have to train only one model which adapts itself, instead of training multiple models for cross-validating K and then training a final model (with full training data) for the chosen K.

3.3.4 Discussion

In the zero-shot learning setting, during training, we have a set of seen classes $\mathcal{Y}_{tr+val} = \{y_1, \ldots, y_{N_1}\}$ and a set of unseen classes $\mathcal{Y}_{ts} = \{y_{N_1+1}, \ldots, y_{N_1+N_2}\}$ with $\mathcal{Y}_{tr+val} \cap \mathcal{Y}_{ts} = \phi$. In addition, all the classes have been assumed to be embedded into a multidimensional real space which connects them via some form of semantics. For example, each class may be written as a binary vector indicating the presence of absence of predefined attributes e.g. furry, has tail, can swim. During training we are provided with annotated training images belonging to the classes in $\mathcal{Y}_{tr+val}$, while at testing we are required to make predictions for images belonging to the classes in $\mathcal{Y}_{ts}$.

Zero-shot learning can be achieved by using any compatibility learning model, such as the bilinear compatibility based model presented in section 3.2, as there is no class specific parameter being learnt (cf. multi-class SVM models) but only a global parameter W which maps the image embeddings to class embeddings (and vice-versa). We build upon the SJE model presented in section 3.2 for the task of zero-shot learning and now discuss the differences between LatEm and SJE to emphasize our technical contributions.

LatEm learns a piecewise linear compatibility function through multiple W_i matrices whereas SJE (Akata *et al.*, 2015c) is linear. With multiple W_i's the compatibility function has the freedom to treat different types of images differently. Let us consider a fixed class $\hat{\mathbf{y}}$ and two substantially visually different types of images $\mathbf{x}_1, \mathbf{x}_2$, e.g. the same bird flying and swimming. In SJE (Akata *et al.*, 2015c) these images will be mapped to the class embedding space with a single mapping $W^\top \mathbf{x}_1, W^\top \mathbf{x}_2$. On the other hand, LatEm will have learned two different matrices for the mapping i.e. $W_1^\top \mathbf{x}_1, W_2^\top \mathbf{x}_2$. In the former case, a single W has to map two visually, and hence numerically, very different vectors (close) to the same point. In the latent case as two

different mappings are factorized separately, therefore the "flying" and "swimming" bird will be mapped to two separate points. Such factorization is also expected to be advantageous when two classes that share partial visual similarity are to be discriminated. For instance, while blue birds could be relative easily distinguished from red birds, to do so for different types of blue birds is harder. In such cases, one of the W_i's could focus on color while another one could focus on the beak shape (in section 3.4 we show that this effect is visible). The task of discrimination against different bird species would then be handled only by the second one. This way of factorizing enables for a more disctiminative classification model.

LatEm uses the ranking based loss (Weston *et al.*, 2011) in Equation 3.7 whereas SJE (Akata *et al.*, 2015c) uses the multiclass loss of Crammer and Singer (Crammer and Singer, 2002) which replaces the $\sum$ in Equation 3.7 with max. The SGD algorithm for multiclass loss of Crammer and Singer (Crammer and Singer, 2002) requires at each iteration a full pass over all the classes to search for the maximum violating class. Therefore it can happen that some matrices will not be updated frequently. On the other hand, the ranking based loss in Equation 3.7 used by our LatEm model ensures that different latent matrices are updated frequently. Thus, the ranking based loss in Equation 3.7 is better suited for our piecewise linear LatEm model.

3.4 EXPERIMENTS

In this section, first we detail our experimental setup in our evaluation procedure and finally report experimental results on zero-shot and generalized zero-shot learning settings.

Datasets. Caltech-UCSD Birds (CUB) (Welinder *et al.*, 2010) and Stanford Dogs (Dogs) (Khosla *et al.*) are fine-grained datasets (e.g. Duan *et al.*, 2012; Deng *et al.*, 2013) and Animals With Attributes (AWA) (Lampert *et al.*, 2013) is a coarse-grained dataset. All the three datasets have been used for zero-shot learning (e.g. Akata *et al.*, 2015c; Rohrbach *et al.*, 2011; Kankuekul *et al.*, 2012; Yu and Aloimonos, 2010) in the literature. As shown on Table 3.1, the set of classes are divided into three disjoint sets of train ($\mathcal{Y}_{tr}$), val ($\mathcal{Y}_v$) and test ($\mathcal{Y}_{ts}$) classes. For a fair comparison with previous works, we follow the same train, val, test set split used by (Akata *et al.*, 2015c).

In zero-shot learning, i.e. $\mathcal{Y}_{tr+v} \cap \mathcal{Y}_{ts} = 0$, to get a more stable estimate of our own results, we make four more splits by randomly sampling the same number of classes as before. Unless indicated otherwise, e.g. in comparison with previous methods, we average results over five splits. We account for the imbalance in the number of images in AWA and Dogs datasets and measure per-class averaged Top-1 accuracy, unless stated otherwise.

In generalized zero-shot learning setting as shown on Table 3.2, the set of images that belong to $\mathcal{Y}_{tr+v}$ and $\mathcal{Y}_{ts}$ is first divided equally into tr+v and ts sets. Namely, following the same seen ($\mathcal{Y}_{tr+v}$) and unseen ($\mathcal{Y}_{ts}$) class split as the zero-shot learning setting, we build tr+v and ts sets of images that belong to seen and unseen classes. This way we can evaluate our model on images that belong to only ts or both tr+v

	Total		train+val			test	
	img	$\mathcal{Y}$	img	$\mathcal{Y}_{tr}$	$\mathcal{Y}_v$	img	$\mathcal{Y}_{ts}$
CUB	11788	200	8855	100	50	2931	50
AWA	30475	50	24293	30	10	6180	10
Dogs	19499	113	14681	57	28	4818	28

Table 3.1: The statistics of CUB, AWA and Dogs datasets in zero-shot setting. CUB and Dogs are fine-grained datasets whereas AWA is a more general concept dataset. $\mathcal{Y}_{tr+v}$ and $\mathcal{Y}_{ts}$ are seen and unseen class embeddings respectively.

	img		cls	img		cls
	tr+v	ts	$\mathcal{Y}_{tr+v}$	tr+v	ts	$\mathcal{Y}_{ts}$
CUB	4495	4360	150	1499	1434	50
AWA	12176	12119	40	3062	3118	10
Dogs	7317	7364	85	2433	2385	28

Table 3.2: The statistics of CUB, AWA and Dogs datasets in the generalized zero-shot learning setting.

and ts classes.

Image and class embeddings. For direct comparison with the state-of-the-art, we use embeddings provided by (Akata *et al.*, 2015c). Briefly, as image embeddings we use the 1024 dimensional top-layer pooling units of the pre-trained GoogleNet (Szegedy *et al.*, 2015) extracted from the whole image. We do not do any task specific pre-processing on images such as cropping foreground objects. As class embeddings we evaluate four different alternatives, i.e. attributes (att) (Lampert *et al.*, 2013), word2vec (w2v) (Mikolov *et al.*, 2013b), glove (glo) (Pennington *et al.*, 2014) and hierarchies (hie) (Miller, 1995). Note that, CUB contains 312 and AWA contains 85 attributes. Our att embedding for a class is a vector measuring the strength of each attribute for that class, based on human judgment. On the other hand, w2v and glo are 400 dimensional whereas hie is ≈ 200 dimensional.

Implementation details. Our image features are z-score normalized such that each dimension has zero mean and unit variance. All the class embeddings are ℓ_2 normalized. The matrices W_i are initialized at random with zero mean and standard deviation $\frac{1}{\sqrt{d_x}}$ (Akata *et al.*, 2015a). The number of epochs is fixed to be 150. The learning rates for the CUB, AWA and Dog datasets are chosen as $\eta_t = 0.1, 0.001, 0.01$, respectively, and kept constant over iterations. For each dataset, these parameters are tuned on the validation set of the default dataset split and kept constant for all other dataset splits and for all class embeddings. We use two strategies for selecting the number of latent matrices K, i.e either cross-validation or pruning. For cross-validation, K is varied in $\{2, 4, 6, 8, 10\}$ and the optimal K is chosen based the

	CUB				AWA				Dogs		
	att	w2v	glo	hie	att	w2v	glo	hie	w2v	glo	hie
ESZSL	30.5	23.7	7.1	2.1	65.3	29.3	38.4	52.2	10.0	6.5	21.3
ESZSL*	47.1	**33.7**	**33.3**	23.2	68.8	57.4	61.7	55.1	21.6	20.0	22.1
CMT	29.4	24.8	25.8	17.9	54.9	46.6	47.6	40.1	13.7	16.7	14.8
SSE	42.1	28.4	24.9	21.4	64.8	60.4	**65.8**	55.8	20.5	18.9	**29.9**
JLSE	37.6	28.4	29.9	20.3	67.5	49.7	56.4	39.3	**26.2**	16.4	23.7
SJE	**50.1**	28.4	24.2	20.6	66.7	51.2	58.8	51.2	19.6	17.8	24.3
LatEm (Ours)	45.5	31.8	32.5	**24.2**	**71.9**	**61.1**	62.9	**57.5**	22.6	**20.9**	25.2

Table 3.3: Average per-class top-1 accuracy in zero-shot setting on AWA, CUB and Dogs datasets. We compare ESZSL (Romera-Paredes *et al.*, 2015), ESZSL* (Romera-Paredes *et al.*, 2015), CMT (Socher *et al.*, 2013), SSE (Zhang and Saligrama, 2015), JLSE (Zhang and Saligrama, 2016), SJE (Akata *et al.*, 2015c) and Latent Embedding model (K is cross-validated) using the same splits, image and class embeddings as in (Akata *et al.*, 2015c).

accuracy on a validation set. For pruning, unless stated otherwise, K is initially set to be 16 and then at every fifth epoch during training, we prune all matrices that support less than 5% of the data points.

3.4.1 Zero-shot Learning Experiments

In this section, we provide results on zero-shot learning setting where $\mathcal{Y}_{tr} \cap \mathcal{Y}_v \cap \mathcal{Y}_{ts} = 0$. In this setting, at training time, LatEm has access to labeled images of $\mathcal{Y}_{tr+v}$ and the search space at test time is $\mathcal{Y}_{ts}$. We either use the splits provided by (Akata *et al.*, 2015c) or report the average performance of five splits to show stability. We specify the splits we used for each experiment in their respective sections.

Comparison with State-of-the-Art. We start our experimental evaluation with an analysis of (Lampert *et al.*, 2013) and quantitative comparisons with ESZSL (Romera-Paredes *et al.*, 2015), CMT (Socher *et al.*, 2013), SSE (Zhang and Saligrama, 2015), JLSE (Zhang and Saligrama, 2016), and SJE (Akata *et al.*, 2015c) which are among the most relevant related work to ours. Note that we fairly re-evaluate all seven state-of-the-art methods using the same four class embeddings, the same image embeddings and the same evaluation criteria on three challenging zero-shot learning datasets. Therefore, ours is one of the most comprehensive re-evaluation of zero-shot state-of-the-art.

Among competing state-of-the-art methods, (Lampert *et al.*, 2013) proposes a two-step method that follows a different principle than ours: (1) Learning attribute classifiers and (2) Combining the scores of these attribute classifiers to make a class prediction. Typically, the positive/negative samples used to train the attribute classifiers are obtained by binarizing the class-attribute matrix wrt. a threshold,

	CUB		**AWA**		**Dogs**	
	PR	CV	PR	CV	PR	CV
att	3	4	7	2	n/a	
w2v	8	10	8	4	6	8
glo	6	10	7	6	9	4
hie	8	2	7	2	11	10

Table 3.4: Number of matrices selected using pruning (PR) and using cross-validation (CV). PR is obtained by $K_0 = 16$.

that leads to loss of information. As it is not clear how to extend this idea to unsupervised class embeddings, we compare (Lampert *et al.*, 2013) and LatEm using attributes on AWA where (Lampert *et al.*, 2013) obtains 56.2% whereas LatEm obtains 71.9% accuracy which is mostly due to binary attributes. On the other hand, we emphasize that we focus on unsupervised class embeddings that do not require human supervision. Additionally, we re-implemented (Romera-Paredes *et al.*, 2015) following the paper because their method is embarrassingly simple. (Romera-Paredes *et al.*, 2015) define a binary matrix Y of size $m \times z$ to denote the ground-truth labels of m training instances belonging to any of the z classes. The scale of this matrix has been given as $Y \in \{-1, 1\}^{m \times z}$ in (Romera-Paredes *et al.*, 2015) which is a parameter to tune. Therefore, we also validate our results with $Y \in \{0, 1\}^{m \times z}$. We denote the experiment that uses $Y \in \{0, 1\}^{m \times z}$ as (Romera-Paredes *et al.*, 2015)*. For our experiments, we got the code from the authors of (Socher *et al.*, 2013), (Zhang and Saligrama, 2015), and (Zhang and Saligrama, 2016) and we use the publicly available implementation of SJE (Akata *et al.*, 2015c). We ran the experiments using our image and class embeddings by carefully validating all the parameters of all the methods on the validation set.

We present results in Table 3.3. Our LatEm consistently outperforms (Socher *et al.*, 2013) and (Romera-Paredes *et al.*, 2015) on all three datasets for all four class embeddings. We observe a significant increase in accuracy from ESZSL (Romera-Paredes *et al.*, 2015) to ESZSL* (Romera-Paredes *et al.*, 2015) in all cases. However, even with $Y \in \{0, 1\}^{m \times z}$, our LatEm still outperforms ESZSL*(Romera-Paredes *et al.*, 2015) in 8 out of 11 cases. On the other hand, our LatEm outperforms (Zhang and Saligrama, 2015) in 9 out of 11 cases and (Zhang and Saligrama, 2016) in 10 out of 11 cases. For (Zhang and Saligrama, 2015) λ_1, λ_2, γ are the three regularization parameters, also the number of iterations and number of sample pairs are hyperparameters to tune whereas (Zhang and Saligrama, 2016) requires the regularization λ_s, dictionary size, number of sample pairs and number of iterations to be tuned. Note that, apart from doing an extensive parameter validation, we used exactly the same SVM solver and quadratic programming solver with (Zhang and Saligrama, 2015) and (Zhang and Saligrama, 2016) to obtain the results in Table 3.3. Being a competitive state-of-the-art and the closest work related to ours, we now

					CUB		AWA		Dog	
	att	w2v	glo	hie	SJE	LatEm	SJE	LatEm	SJE	LatEm
cnc	✓	✓		✓	45.1	42.0	71.3	64.5	n/a	n/a
cmb					**51.0**	46.2	73.5	**73.6**	n/a	n/a
cnc	✓		✓	✓	42.2	39.7	73.3	70.7	n/a	n/a
cmb					**51.7**	46.6	73.9	**75.7**	n/a	n/a
cnc		✓		✓	28.2	30.7	53.9	59.7	23.5	30.0
cmb					29.4	**33.2**	55.5	**62.2**	26.6	**33.8**
cnc			✓	✓	28.5	31.3	60.1	**71.1**	23.5	25.9
cmb					29.9	**32.6**	59.5	64.8	26.7	**26.8**

Table 3.5: Class embeddings combined as in (Akata *et al.*, 2015c) (cnc: early fusion of class embeddings, cmb: late fusion of scores).

provide a detailed comparison with SJE (Akata *et al.*, 2015c) and our LatEm.

Using att, LatEm improves over SJE on AWA (71.9% vs. 66.7%) significantly. However, as our aim is to reduce the accuracy gap between supervised and unsupervised class embeddings, therefore we focus on w2v, glo and hie embeddings. Here, on all datasets, LatEm improves the SJE (Akata *et al.*, 2015c) (section 3.2) significantly. With w2v, LatEm achieves 31.8% (vs. 28.4%) on CUB, 61.1% (vs. 51.2%) on AWA and finally 22.6% (vs 19.6%) accuracy on Dogs. Similarly, using glo, LatEm achieves 32.5% (vs 24.2%) on CUB, 62.9% (vs. 58.8%) on AWA and 20.9% (vs. 17.8%) accuracy on Dogs. Finally, while LatEm with hie on Dogs improves the result to 25.2% from 24.3%, the improvement is more significant on CUB (24.2% from 20.6%) and on AWA (57.5% from 51.2%). These results place our LatEm in the context with most recent and relevant methods as well as establish it as another competitive state-of-the-art method for zero-shot learning on three datasets. The results are encouraging, as they quantitatively show that learning piece-wise linear latent embeddings indeed capture latent semantics on the class embedding space.

Here, we emphasize two disadvantages of attributes. First, since fine-grained object classes share many common properties we need a large number of attributes which is costly to obtain. Second, attribute annotations need to be done on a dataset basis, i.e. the attributes collected for birds do not work with dogs. Therefore, we stress the importance of the unsupervised class embeddings i.e. w2v, glo, hie.

Pruning versus cross-validation for model selection. Our aim is to determine if our LatEm selects different number of models through pruning and through cross-validation. Pruning (PR) selects matrices based on the data itself, on the other hand, cross-validation (CR) validates the number of matrices necessary to obtain the highest accuracy on the validation set. Table 3.4 presents the results of this experiment on splits provided by (Akata *et al.*, 2015c).

We set the initial number of embeddings K_0 to 16 and pruning threshold to $1/K_0$ which assumes that samples are equally distributed to each embedding. In terms

of the model size, cross validation seems to have a slight advantage. It selects a smaller model in 7 cases out of 11 which is more space and time efficient. The trend is consistent for all the class embeddings for the AwA dataset but is mixed for CUB and Dogs. The advantage of pruning over cross-validation is that it is much faster to train. While cross validation requires training and testing with multiple models (once each for every possible choice of K), pruning just requires training once. We measure the sensitivity of K_0 and corresponding pruning thresholds by setting $K_0 = [10, 12, 14, 16, 18, 20, 22]$ and $th = 1/10, 1/12, 1/14, \ldots, 1/22$. Mean accuracy with standard deviation with att, w2v, glo, hie on CUB are 44.9% (0.6), 32.4% (0.7), 31.6% (1.3), 22.8% (0.9) which shows that the results we reported with $K_0 = 16$ is stable.

Combination of class embeddings. Here, we provide results with direct comparison with (Akata *et al.*, 2015c) where class embeddings are combined using two strategies: (1) through early fusion (cnc), i.e. concatenating class embeddings and (2) through late fusion (cmb) of compatibility scores, i.e. averaging the scores obtained with different class embeddings. We use the same combination of class embeddings, image features and zero-shot splits as (Akata *et al.*, 2015c) for a fair comparison. The results are presented in Table 3.5.

First, we combine att with w2v, glo and hie for AWA and CUB. LatEm improves the results over SJE significantly on AWA (75.7% vs 73.9%). On the other hand, LatEm does not improve over the state-of-the-art (46.6% vs 51.7%) on CUB. This observation is in line with the results reported in Table 3.3 where LatEm does not provide a significant advantage over SJE on CUB with human-annotated attributes.

Second, we combine unsupervised class embeddings w2v, glo and hie. LatEm consistently improves over SJE in this setting. On CUB combining w2v, glo and hie achieves 34.9% (vs. 29.9%), on AWA it achieves 66.2% (vs. 60.1%) and on Dogs it obtains 36.3% (vs. 35.1%). These experiments show that unsupervised class embeddings contain complimentary information and, therefore, the results tend to improve by combining them. Another observation is late fusion of classification scores, i.e. cmb, leads to higher accuracy compared to early fusion of class embeddings, i.e. cnc. In cnc, a single W_i, learned with all the class embeddings concatenated together, fails to address the confusion that is introduced by each class embedding. On the other hand, in cmb, each W_i prefers to assign a different class label to an image based on the score, i.e. $F(\mathbf{x}, \mathbf{y})$. This way, different W_is that are learned with different but complimentary class embeddings get weighted accordingly and, hence, class labels are more accurate.

Finally, on CUB and Dogs by combining w2v and hie we obtain better results than by combining glo and hie. This is due to the fact that glo uses only class-relevant articles while w2v uses the entire wikipedia. As a conclusion, wikipedia articles that are not directly related to our classes, i.e. the context, lead to more descriptive class embeddings individually (see w2v results in Table 3.3) and in combination as well (see results in Table 3.5 that include w2v).

Stability of zero-shot learning results. As during training time in zero-shot learning

	CUB		AWA		Dogs	
	SJE	LatEm	SJE	LatEm	SJE	LatEm
att	**49.5**	45.6	70.7	**72.5**	n/a	
w2v	27.7	**33.1**	49.3	**52.3**	23.0	**24.5**
glo	24.8	**30.7**	50.1	**50.7**	14.8	**20.2**
hie	21.4	**23.7**	43.4	**46.2**	24.6	**25.6**

Table 3.6: Average per-class top-1 accuracy on unseen classes (the results are averaged on five folds). SJE: (Akata *et al.*, 2015c), LatEm: Latent embedding model (K is cross-validated).

	CUB		AWA		Dogs	
	PR	CV	PR	CV	PR	CV
att	43.8	**45.6**	63.2	**72.5**	n/a	
w2v	**33.9**	33.1	48.9	**52.3**	**25.0**	24.5
glo	**31.5**	30.7	**51.6**	50.7	18.8	**20.2**
hie	**23.8**	23.7	45.5	**46.2**	25.2	**25.6**

Table 3.7: Average per-class top-1 accuracy on unseen classes (averaged over five zero-shot splits that we used in the stability experiments). PR: proposed model learnt with pruning using $K_0 = 16$, CV: with cross validation.

neither images nor class relationships of test classes are seen, methods suffer from the difficulty in parameter selection. The standard way is to use disjoint train, val and test classes. In addition to the standard splits, we experimented on four more independently and randomly chosen data splits to get stable estimates of our predictions. Both with our LatEm and the publicly available implementation of SJE (Akata *et al.*, 2015c) we repeat these experiments five times and report the average.

For all datasets Table 3.6 shows that all the result comparisons between SJE and LatEm hold and therefore conclusions are the same. Although SJE outperforms LatEm with supervised attributes on CUB, LatEm outperforms the SJE results with supervised attributes on AWA and consistently outperforms all the SJE results obtained with unsupervised class embeddings. Using attributes, on AWA LatEm obtains an impressive 72.5% (vs. 70.5%) and using unsupervised class embeddings the highest accuracy is observed with w2v with 52.3% (vs. 49.3%). On CUB, LatEm with w2v obtains the highest accuracy with 33.1% (vs. 27.7%) On Dogs, LatEm with hie obtains the highest accuracy, i.e. 25.6% (vs 24.6%). These results insure that our accuracy improvements reported in Table 3.3 were not due to a bias in the dataset split. By augmenting the datasets with four more splits, our LatEm obtains a

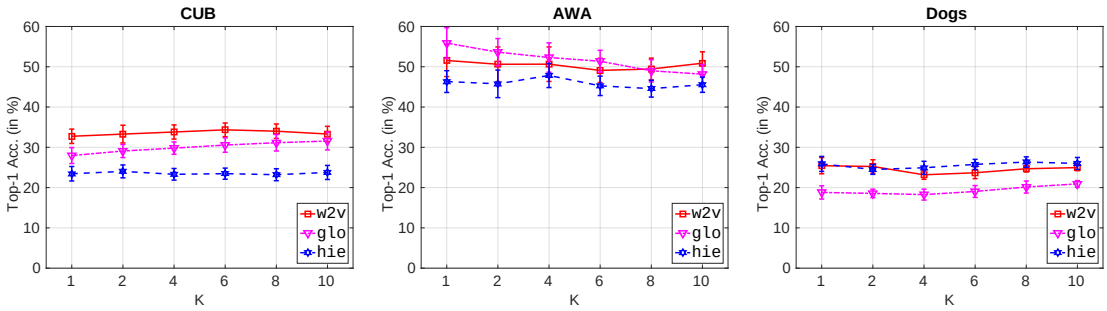

Figure 3.2: Effect of latent variable K on CUB, AWA and Dogs datasets. We measure Top-1 Accuracy (in %) with the increasing number of latent models, i.e. K, learned with unsupervised class embeddings, i.e. w2v, glo, hie.

consistent improvement on all the class embeddings on all datasets over the state-of-the-art. On the other hand, these results helped us notice one crucial difference of doing zero-shot learning on fine-grained and on coarse-grained datasets. The results reported on the original split of AWA (Lampert *et al.*, 2013) that is being widely used in the literature has been constructed in a way that seen and unseen class splits have visually similar classes e.g. while *gorilla* is in the seen classes, *chimpanzee* is in the unseen classes. This insures that by using *gorilla* images, the methods will generalize to images of the the visually similar *chimpanzee* class whose images were not seen on training. When we build another split that places both *gorilla* and *chimpanzee* classes in the unseen/test set, there is no means of distinguishing these objects, as there is no visually closely similar class left in the seen/train set. We observe a significant drop in accuracy for the weaker unsupervised class embeddings on AWA when we randomly select the class splits, as given in Table 3.6, in addition to the original split (Lampert *et al.*, 2013). However, this drop effects our LatEm as well as the state-of-the-art SJE method. Our conclusion from this observation is that the zero-shot learning setting may be better suited for fine-grained classification task.

We also evaluate the accuracy of LatEm when the number of matrices in the model is obtained with pruning versus when it is obtained with cross-validation. Table 3.7 presents the performance of LatEm when the model selection is done by pruning (PR) or by cross-validation (CR) on the three datasets. In terms of performance, both methods are equally competitive. Pruning outperforms cross validation on five cases and is outperformed on the remaining six cases. The performance gaps are usually within 1-2% absolute, with the exception of AWA dataset with att and w2v with 72.5% vs. 70.7% and 52.3% vs. 49.3%, for CV and for PR respectively. Hence, neither of the methods has a clear advantage in terms of performance, however cross validation in general performs slightly better and is faster.

Effect of K. In this section, we investigate the experiments performed using five-folds on the CUB, AWA and Dogs datasets and provide further analysis for a varying number of K. For completeness of the analysis, we also evaluate the single latent embedding case, namely $K \in \{1, 2, 4, 6, 8, 10\}$ using unsupervised embeddings, i.e. w2v, glo and hie for consistency.

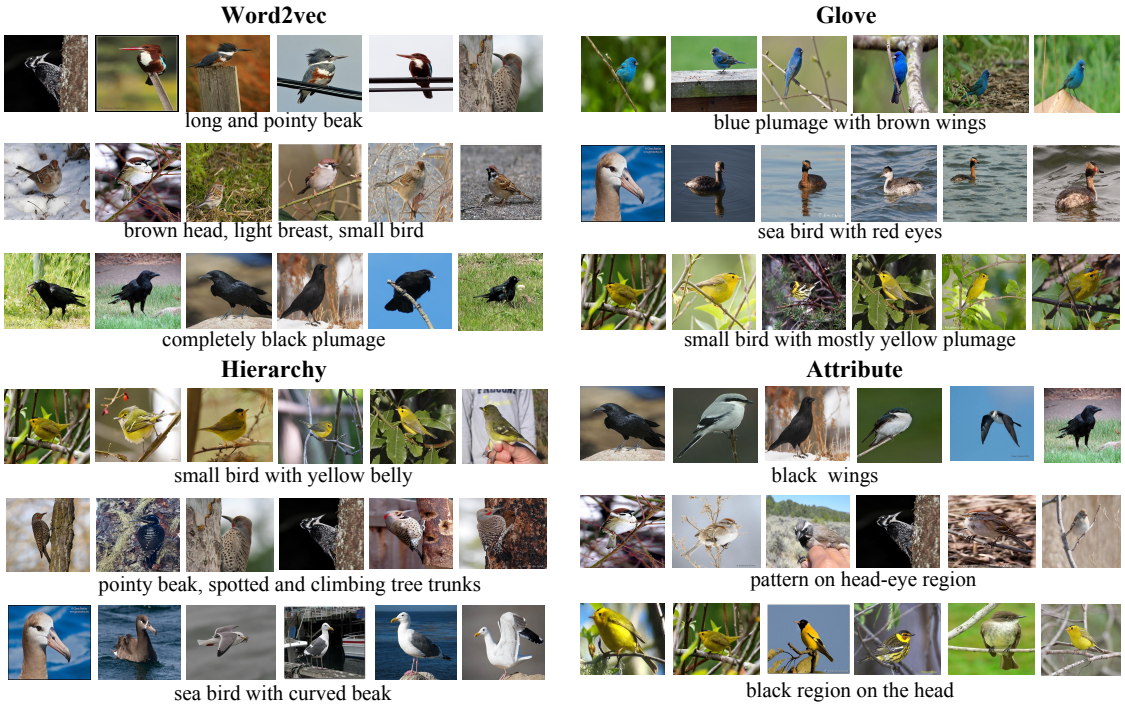

Figure 3.3: Top images ranked by the matrices using word2vec, glove, hierarchy and attribute class embeddings on CUB dataset, each row corresponds to different matrix in the model. Qualitative examples support our intuition – each latent variable captures certain visual aspects of the bird. Note that, while the images may not belong to the same fine-grained class, they share common visual properties.

In Figure 3.2 we present the performance of the model with a different number of matrices on CUB, AWA and Dogs datasets. For CUB, we observe that the performance generally increases with increasing K, initially, and then the patterns differ with different embeddings. With w2v the performance keeps increasing until $K = 6$ and then starts decreasing, probably due to model overfitting. With glo the performance increases until $K = 10$ where the final accuracy is $\approx 5\%$ higher than with $K = 1$. With the hie embedding the standard errors do not increase significantly in any of the cases, are similar for all values of K and there is no clear trend in the performance. For AWA, although glo results decrease with the increasing number of K, for w2v and hie the results do not vary significantly but they pick the values 10 and 4 respectively. For Dogs, this time w2v results decrease slightly with the increasing number of K for $K > 2$. In this dataset, $K = 2, 8, 10$ seems to be the best options for w2v, hie and glo respectively.

Interpretability of latent embeddings. As we demonstrated previously, our novel LatEm model improves the state-of-the-art SJE model for zero-shot classification on two fine-grained datasets, i.e. CUB and Dogs, and one coarse-grained dataset, i.e. AWA. In this section, we take a closer look at the results on the challenging CUB dataset and investigate if individual W_i's learn visually consistent and interpretable latent relationships between images and classes. Figure 7.7 shows the top scoring

	CUB			AWA			Dogs		
	T1	T5	T10	T1	T5	T10	T1	T5	T10
att	12.4	46.8	67.4	4.8	65.6	90.6		n/a	
w2v	0.7	29.2	46.3	0.0	31.2	63.5	0.0	6.6	20.5
glo	0.5	26.0	40.5	0.0	36.1	66.2	0.0	6.1	21.3
hie	0.0	19.7	36.3	0.0	40.0	62.1	1.3	15.0	31.8

Table 3.8: Average per-class top-1, 5 and 10 accuracy, i.e. T1, T5 and T10 respectively, in generalized zero-shot learning setting when we have no samples from $\mathcal{Y}_{ts}$ during training, however the search space during testing includes all the available labels, i.e. namely $\mathcal{Y} = \mathcal{Y}_{tr} \cup \mathcal{Y}_v \cup \mathcal{Y}_{ts}$.

images retrieved by three different W_i for w2v, glo, hie and att.

For w2v, the images in the first row are of birds which have long and pointy beaks. Note that they belong to different classes; having a long and pointy beak is one of the shared aspect of those different bird species. Similarly, for the second row images are of small birds with brown head and light-colored breast and the last row contains large birds with completely black plumage. These results are interesting because they show that, our LatEm is able to (i) infer hidden common properties of classes and (ii) support them with visual evidence, leading to a clustering which is optimized for classification, and also performs well in retrieval.

For glo, similar to the results with w2v, the top-scoring images of the same W_i consistently show distinguishing visual properties of classes. The first row shows that blue birds from different species are clustered together which indicates that this matrix captures the "blue"ness of the birds. The second row has exclusively aquatic birds, i.e. surrounded by water. Finally, the third row shows yellow birds only. Similar to w2v, for glo our LatEm is able to bring out the latent information that reflect object attributes and support this with its visual counterpart.

For completion, we also include qualitative results with hie and att class embeddings. The first row with hie shows small yellow birds with yellow belly, the second row shows different species of birds with a pointy beak climbing on tree trunks and the third row shows sea birds with curved beaks. Similarly, the first row with att shows different birds with a common property of having "black wings", the second row shows a distinctive pattern on the head region and the third row shows birds with different amount of blackness on their heads. These results clearly demonstrate that our model factorizes the space with visually interpretable relations between classes, also with hie and att.

3.4.2 Generalized Zero-shot Learning Setting

Most existing works on zero-shot learning assume that all the images are from unseen classes during the test phase, which simplifies the problem as the classifiers

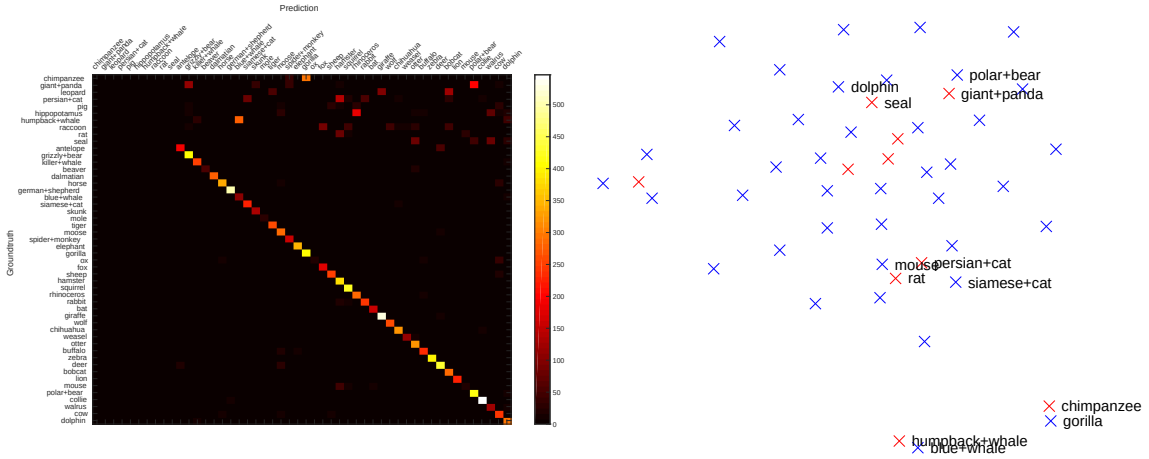

Figure 3.4: Left: Confusion matrix of all the classes on AWA dataset based on the latent factors learned using LatEm in the general setting (we use glo as class embedding). 10 unseen classes are shown at the top of the confusion matrix. Right: t-SNE visualization of the confusion matrix with seen and unseen classes marked with blue and red respectively. Visually similar classes such as chimpanzee and gorilla are embedded close to each other, hence being confused by the classifier.

only need to distinguish between unseen classes. In this section, we evaluate our LatEm in a more challenging yet realistic setting, here the prediction function is:

$$f(\mathbf{x}) = \underset{\mathbf{y} \in \{\mathcal{Y}_u \cup \mathcal{Y}_s\}}{\operatorname{argmax}} \; F(\mathbf{x}, \mathbf{y}). \tag{3.9}$$

As shown in Equation 3.9, in the generalized zero-shot learning setting (e.g. **?**Socher *et al.*, 2013) the search space includes all the class embeddings both at training time and at test time. Similar to the zero-shot learning setting, the extreme case of generalized zero-shot learning setting assumes no availability of visual samples from test classes during training. As we do not have access to any images of $\mathcal{Y}_{ts}$ during training, class embeddings of $\mathcal{Y}_{ts}$ do not get coupled with any visual information, hence act only as distractors. In the following sections, we first evaluate the extreme case of generalized zero-shot learning setting, i.e. when we have no visual samples from test classes during training, and then we gradually increase the number of images from $\mathcal{Y}_{ts}$ during training.

No samples from $\mathcal{Y}_{ts}$ during training. In this setting, during training although we do not have access to any visual samples from test classes, our scoring function takes a max over all the available class embeddings. As the class embeddings of test classes never get any supervision signal, they act as distractors. We present results obtained in this setting on CUB, AWA and Dogs using all four class embeddings on Table 3.8.

Our observation from Table 3.8 is that with Top-1 accuracy LatEm gives poor results even with expert annotated attributes. Note that, a similar behavior was observed in (Rohrbach *et al.*, 2011, 2013; Socher *et al.*, 2013). These results show that

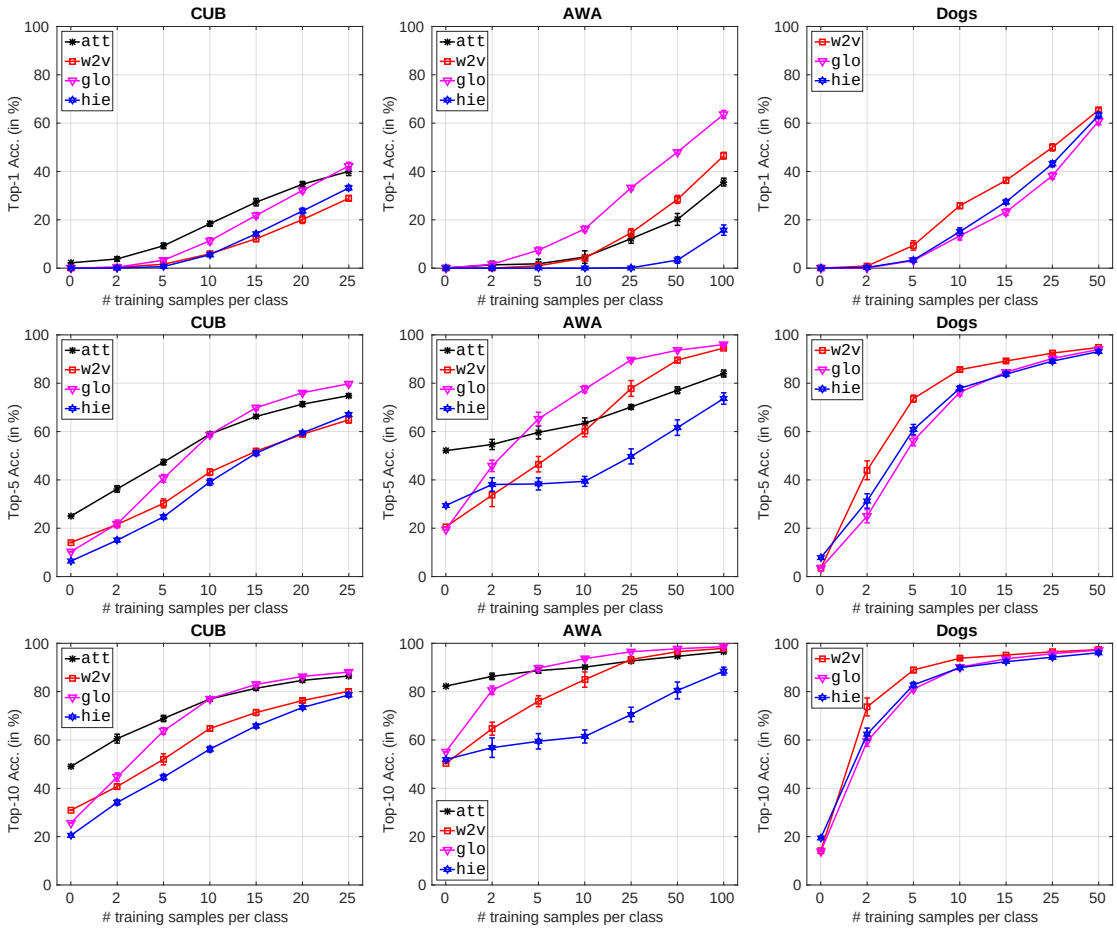

Figure 3.5: Generalized zero- and few-shots learning settings evaluated on all for CUB, AWA and Dogs using att (where available), w2v, glo and hie embeddings. We show the Top-1, Top-5 and top-10 Accuracy (in%) with the increasing number of images per unseen class used during training.

evaluating the model on both seen and unseen classes is a harder problem and it requires more attention. Although solving this problem is out of the scope of this chapter, we provide further analysis on understanding the problem itself.

Our hypothesis is that the classes that are similar in context, i.e. *chimpanzee* and *gorilla*, are separated into different sets in terms of seen and unseen classes. To evaluate this hypothesis, after learning the LatEm model on AWA using glo embedding, we build a confusion matrix of the test images that belong to both seen and unseen classes. Figure 3.4 plots the confusion matrix and t-SNE (van der Maaten and Hinton, 2008) visualization of the confusion matrix. We observe that the classifier is indeed able to embed images of chimpanzees close to the *chimpanzee* and *gorilla*. However, without having seen sufficient examples of the unseen class *chimpanzee*, it is not able to distinguish between a *chimpanzee* and a *gorilla*. Same phenomenon is observed for other visually similar class pairs, e.g. *blue-whale* and *humpback-whale*, *polar-bear* and *giant-panda*, *mouse* and *rat*, which are visually similar

animals belonging to seen and unseen classes respectively.

Following this analysis, we argue that in the presence of seen and unseen classes for testing, evaluating Top-5 or Top-10 accuracy may be a more suitable way to measure performance. Indeed, Top-5 accuracy has been the evaluation criteria of image classification challenge (Berg *et al.*) of ImageNet (Deng *et al.*, 2009). We present results with Top-5 accuracy on Table 3.8. Our immediate observation is that for all datasets the results improve by 6 to 40% compared to the results with Top-1 accuracy. This shows that $6-40\%$ of the time, the images of unseen classes are incorrectly assigned to the second closest class among the seen classes, e.g. *chimpanzee* versus *gorilla*, or vice versa. This outcome follows our intuition that LatEm confuses two similar classes especially when they belong to disjoint sets of seen and unseen classes. Finally, our results with Top-10 accuracy shows a similar tendency to the difference between Top-5 and Top-1 accuracy. We observe another accuracy increase of 15 to 30% compared to the Top-5 accuracy depending on the dataset and class embedding. Moreover, as expected the Top-10 accuracy results are higher than Top-1 and Top-5 accuracy while the relative difference between different class embeddings remain similar in all cases. We also observe from these results that supervised attributes remain important with the lack of training data in the extreme case.

In CUB and AWA, top-10 accuracy obtained with unsupervised class embeddings extracted from wikipedia, i.e. w2v and glo perform similarly to the top-5 accuracy obtained with attribute class embeddings. On the other hand, the human supervision signal that comes from attributes leads to an accuracy boost of almost 30% when we measure top-5 or top-10 accuracy.

Finally in Dogs, hie class embeddings perform higher than w2v and glo that are extracted from wikipedia. It is interesting to note that this observation is unique to this dataset and it is in line with our observations in the classic zero-shot learning setting. This shows that finding the most suitable class embedding is an important aspect of tackling the zero-shot learning task.

Generalized zero-shot to generalized few-shots setting. As shown in the previous section, the presence of all class embeddings, i.e. generalized zero-shot setting, in its extreme case, i.e. no visual samples from test classes during training, result in a significant loss in accuracy compared to the classic zero-shot learning setting. This is expected since during training the test class embedding act as distractors since they are not coupled with any visual examples. In this section, we investigate the generalized zero-shot and generalized few-shot learning settings, namely the settings with the presence of either no or a few examples from test classes for training, respectively. We present the stability of our LatEm in this setting by running it on five dataset folds with the error bars in Figure 3.5. We report per-class averaged Top-1, Top5 and Top-10 accuracy results with all four class embeddings, i.e. att (on CUB and AWA), w2v, glo and hie. We show the importance of visual data by increasing the number of images from 0 to $25, 100$ and 50 on CUB, AWA and Dogs respectively.

On CUB, although att class embedding obtains the highest top-1, top-5 and top-10 accuracy on both the generalized zero-shot and generalized $2-5$-shots settings, it

is interesting to observe that glo embedding reaches the same accuracy after the presence of 10 samples on Top-5 and Top-10 accuracies and obtains the highest accuracy in all cases, i.e. Top-1,5 and 10, when all 25 images are used for training. Another observation from CUB results is that the results are stable in all five folds of the data.

On AWA, a striking observation is how well glo class embedding performs for Top-1, Top-5 and Top-10 accuracy on generalized few-shots learning setting. With the presence of 100 images per class, Top-1 accuracy between att and glo embeddings is 20% for both Top-1 and Top-5 accuracy. Also, on AWA, the accuracy difference between different class embeddings is quite high. This may be because AWA is a coarse-grained dataset as the similar observation does not hold for CUB and Dogs.

On Dogs, unlike the classical and generalized zero-shot learning results, hie embedding is not the best performing class embedding in generalized few-shot learning setting. In this dataset, w2v is the best performing embedding on all evaluation metrics. i.e. Top-1, Top-5 and Top-10 accuracy. Another observation from Dogs results is that with the presence of 50 images per-class during training, all class embeddings converge to the same value, i.e. class embeddings lose their importance.

As a conclusion, with the increasing the number of additional training samples from unseen classes the results improve significantly in all cases until the accuracy improvements flatten out gradually. These results show that with the availability of a large number of images from both seen and unseen classes, the importance of the contribution of class embeddings has been reduced. (Akata *et al.*, 2015a) has shown that using hand-crafted image features, the one-vs-rest SVM strategy becomes more favourable compared to embedding-based methods only with the availability of a large number of annotated images. Here, we show that leveraging deep image features with even a few additional samples, i.e. 2, 5, 10, we improve over human annotated attributes and increase zero-shot accuracy by approximately 20%, demonstrated by the results obtained with AWA.

3.5 CONCLUSIONS

We presented a novel latent variable model, Latent Embeddings (LatEm), for learning a nonlinear (piecewise linear) compatibility function for the task of zero-shot classification. LatEm is a multi-modal method, it uses images and class-level side-information either obtained through human annotation or in an unsupervised way from a large text corpus. LatEm incorporates multiple linear compatibility units and allows each image to choose one of them – such choices being the latent variables. We proposed a ranking based objective to learn the model using an efficient and scalable SGD based solver.

We empirically validated our model on three challenging benchmark datasets for zero-shot classification of Birds, Dogs and Animals. We improved the state-of-the-art for zero-shot learning using unsupervised class embeddings on AWA up to 71.1% (vs. 60.1%) and on two fine-grained datasets, achieving 33.2% (vs.

29.9%) on CUB as well as achieving 33.8% (vs. 26.7%) on Dogs. On AWA, we also improve the accuracy obtained with supervised class embeddings, obtaining 75.7% (vs. 73.9%). This demonstrates quantitatively that our method learns a latent structure in the embedding space through multiple compatibility units. We also presented a qualitative analysis of our results and showed that the latent embeddings learned with our method leads to visual consistencies. Our stability analysis on five dataset folds for all three benchmark datasets showed that our method can generalize well and does not overfit to the current dataset splits. We proposed a new method for selecting the number of latent variables automatically from the data by pruning. Such pruning based method speeds up the training and leads to models with competitive space-time complexities compared to the cross-validation based method.

We further extended our application domain to generalized zero-shot and generalized few-shot learning setting where at training time we assume the availability of either no or a few labeled samples from unseen classes. On the other hand, both at training and test time the search space includes all the class embeddings from seen and unseen classes. As expected, our evaluation on generalized zero-shot learning setting showed a significant loss of accuracy compared to the standard zero-shot learning setting which we analyzed through visualizations and quantitative results. Through these experiments we raised awareness that even state-of-the-art methods confuse two visually similar classes if one of them is an unseen class, i.e. the method has seen no samples from that class. Our evaluation on generalized few-shots setting showed that with as few as two to ten samples from unseen classes, unsupervised class embeddings can outperform the supervised attributes. Therefore, with increasing number of additional training samples, the difference between different class embeddings are reduced. As a future work, we plan to investigate the challenging however realistic generalized zero-shot and generalized few-shots settings further.

4 ZERO-SHOT LEARNING: THE GOOD, THE BAD AND THE UGLY

Contents

In the previous chapter, we propose a non-linear embedding function for better zero-shot learning performance. However, we realize that evaluation settings of previous works are inconsistent, leading to incomparable results. Therefore, in this chapter, we introduce a better zero-shot image classification benchmark and evaluate SOTA approaches under the same evaluation protocols. Our new evaluation protocol includes the convention zero-shot learning that predicts only novel classes and the realistic generalized zero-shot learning where both base and novel classes should be evaluated. We also propose correct class splits where novel classes are not present in the pretraining dataset e.g. ImageNet.

In Chapter 5, we adopt the evaluation setting introduced in this chapter and propose an efficient feature generation approach for the challenging generalized zero-shot learning task. In Chapter 6, we follow the same evaluation protocol, introduce a stronger feature generator by combining VAE and GANs, and show unlabeled data significantly improves quality of generated features. Chapter 7 and

Chapter 8 demonstrate our efforts in advancing zero-shot and few-shot learning for the semantic segmentation and video classification tasks.

4.1 INTRODUCTION

Zero-shot learning aims to recognize objects whose instances may not have been seen during training (e.g. Lampert *et al.*, 2013; Larochelle *et al.*, 2008; Rohrbach *et al.*, 2011; Yu and Aloimonos, 2010; Xu *et al.*, 2017; Ding *et al.*, 2017). The number of new zero-shot learning methods proposed every year has been increasing rapidly, i.e. the good aspects as our title suggests. Although each new method has been shown to make progress over the previous one, it is difficult to quantify this progress without an established evaluation protocol, i.e. the bad aspects. In fact, the quest for improving numbers has lead to even flawed evaluation protocols, i.e. the ugly aspects. Therefore, in this work, we propose to extensively evaluate a significant number of recent zero-shot learning methods in depth on several small to large-scale datasets using the same evaluation protocol both in zero-shot, i.e. training and test classes are disjoint, and the more realistic generalized zero-shot learning settings, i.e. training classes are present at test time. Figure 8.1 presents an illustration of zero-shot and generalized zero-shot learning tasks.

We benchmark and systematically evaluate zero-shot learning w.r.t. three aspects, i.e. methods, datasets and evaluation protocol. The crux of the matter for all zero-shot learning methods is to associate observed and non observed classes through some form of auxiliary information which encodes visually distinguishing properties of objects. Different flavors of zero-shot learning methods that we evaluate in this work are linear (e.g. Frome *et al.*, 2013; Akata *et al.*, 2013, 2015c; Romera-Paredes *et al.*, 2015) and nonlinear (e.g. Xian *et al.*, 2016; Socher *et al.*, 2013) compatibility learning frameworks which have dominated the zero-shot learning literature in the past few years whereas an orthogonal direction is learning independent attribute (Lampert *et al.*, 2013) classifiers and finally others (e.g. Zhang and Saligrama, 2015; Changpinyo *et al.*, 2016; Norouzi *et al.*, 2014) propose a hybrid model between independent classifier learning and compatibility learning frameworks which have demonstrated improved results over the compatibility learning frameworks both for zero-shot and generalized zero-shot learning settings.

We thoroughly evaluate the second aspect of zero-shot learning, by using multiple splits of several small, medium and large-scale datasets (e.g. Patterson and Hays, 2012; Welinder *et al.*, 2010; Lampert *et al.*, 2013; Farhadi *et al.*, 2009; Deng *et al.*, 2009). Among these, the Animals with Attributes (AWA1) dataset (Lampert *et al.*, 2013) introduced as a zero-shot learning dataset with per-class attribute annotations, has been one of the most widely used datasets for zero-shot learning. However, as AWA1 images does not have the public copyright license, only some image features, i.e. SIFT (Lowe, 2004), DECAF (Donahue *et al.*, 2014), VGG19 (Simonyan and Zisserman, 2014b) of AWA1 dataset is publicly available, rather than the raw images. On the other hand, improving image features is a significant part of the progress both

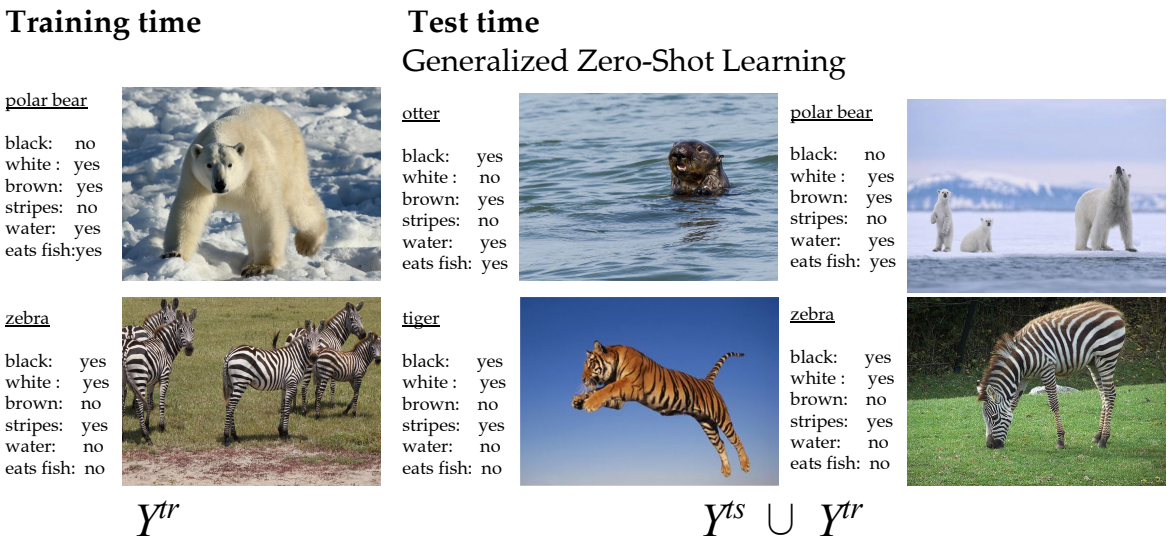

Figure 4.1: Zero-shot learning (ZSL) vs generalized zero-shot learning (GZSL): At training time, for both cases the images and attributes of the seen classes ($\mathcal{Y}^{tr}$) are available. At test time, in the ZSL setting, the learned model is evaluated only on unseen classes ($\mathcal{Y}^{ts}$) whereas in GZSL setting, the search space contains both training and test classes ($\mathcal{Y}^{tr} \cup \mathcal{Y}^{ts}$). To facilitate classification without labels, both tasks use some form of side information, e.g. attributes. The attributes are annotated per class, therefore the labeling cost is significantly reduced.

for supervised learning and for zero-shot learning. In fact, with the fast pace of deep learning, everyday new deep neural network models improve the ImageNet classification performance are being proposed. Without access to images, those new DNN models can not be evaluated on AWA1 dataset. Therefore, with this work, we introduce the Animals with Attributes 2 (AWA2) dataset that has roughly the same number of images all with public licenses, exactly the same number of classes and attributes as the AWA1 dataset. We will make both ResNet (He *et al.*, 2016) features of AWA2 images and the images themselves publicly available.

We propose a unified evaluation protocol to address the third aspect of zero-shot learning which is one of the most important ones. We emphasize the necessity of tuning hyperparameters of the methods on a validation class split that is disjoint from training classes as improving zero-shot learning performance via tuning parameters on test classes violates the zero-shot assumption. We argue that per-class averaged top-1 accuracy is an important evaluation metric when the dataset is not well balanced with respect to the number of images per class. We point out that extracting image features via a pre-trained deep neural network (DNN) on a large dataset that contains zero-shot test classes also violates the zero-shot learning idea as image feature extraction is a part of the training procedure. Moreover, we argue that demonstrating zero-shot performance on small-scale and coarse grained datasets, i.e. aPY (Farhadi *et al.*, 2009) is not conclusive. On the other hand, with this work

we emphasize that it is hard to obtain labeled training data for fine-grained classes of rare objects recognizing which requires expert opinion. Therefore, we argue that zero-shot learning methods should be also evaluated on least populated or rare classes. We recommend to abstract away from the restricted nature of zero-shot evaluation and make the task more practical by including training classes in the search space, i.e. generalized zero-shot learning setting. Therefore, we argue that our work plays an important role in advancing the zero-shot learning field by analyzing the good and bad aspects of the zero-shot learning task as well as proposing ways to eliminate the ugly ones.

4.2 RELATED WORK

A more comprehensive literature review can be found in Chapter 2. Here we only discuss the relation of our benchmark to existing zero-shot learning evaluation protocols.

Zero-shot learning has been criticized for being a restrictive set up as it comes with a strong assumption of the image used at prediction time can only come from unseen classes. Therefore, generalized zero-shot learning setting (Scheirer *et al.*, 2013) has been proposed to generalize the zero-shot learning task to the case where both seen and unseen classes are used at test time. (Jain *et al.*, 2014) argues that although ImageNet classification challenge performance has reached beyond human performance, we do not observe similar behavior of the methods that compete at the detection challenge which involves rejecting unknown objects while detecting the position and label of a known object. (Frome *et al.*, 2013) uses label embeddings to operate on the generalized zero-shot learning setting whereas (Zhang *et al.*, 2016a) proposes to learn latent representations for images and classes through coupled linear regression of factorized joint embeddings. On the other hand, (Bendale and Boult, 2016) introduces a new model layer to the deep net which estimates the probability of an input being from an unknown class and (Socher *et al.*, 2013) proposes a novelty detection mechanism.

Although zero-shot vs generalized zero-shot learning evaluation works exist (Rohrbach *et al.*, 2011; Chao *et al.*, 2016) in the literature, our work stands out in multiple aspects. For instance, (Rohrbach *et al.*, 2011) operates on the ImageNet 1K by using 800 classes for training and 200 for test. One of the most comprehensive works, (Chao *et al.*, 2016) provides a comparison between five methods evaluated on three datasets including ImageNet with three standard splits and proposes a metric to evaluate generalized zero-shot learning performance. On the other hand, we evaluate ten zero-shot learning methods on five datasets with several splits both for zero-shot and generalized zero-shot learning settings, provide statistical significance and robustness tests, and present other valuable insights that emerge from our benchmark. In this sense, ours is the most extensive evaluation of zero-shot and generalized zero-shot learning tasks in the literature.

4.3 EVALUATED METHODS

We start by formalizing the zero-shot learning task and then we describe the zero-shot learning methods that we evaluate in this work. Given a training set $\mathcal{S} = \{(x_n, y_n), n = 1...N\}$, with $y_n \in \mathcal{Y}^{tr}$ belonging to training classes, the task is to learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ by minimizing the regularized empirical risk:

$$\frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n; W)) + \Omega(W) \tag{4.1}$$

where $L(.)$ is the loss function and $\Omega(.)$ is the regularization term. Here, the mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ from input to output embeddings is defined as:

$$f(x; W) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, F(x, y; W) \tag{4.2}$$

At test time, in zero-shot learning setting, the aim is to assign a test image to an unseen class label, i.e. $\mathcal{Y}^{ts} \subset \mathcal{Y}$ and in generalized zero-shot learning setting, the test image can be assigned either to seen or unseen classes, i.e. $\mathcal{Y}^{tr+ts} \subset \mathcal{Y}$ with the highest compatibility score.

4.3.1 Learning Linear Compatibility

Attribute Label Embedding (ALE) (Akata *et al.*, 2015a), Deep Visual Semantic Embedding (DEVISE) (Frome *et al.*, 2013) and Structured Joint Embedding (SJE) (Akata *et al.*, 2015c) use bi-linear compatibility function to associate visual and auxiliary information:

$$F(x, y; W) = \theta(x)^T W \phi(y) \tag{4.3}$$

where $\theta(x)$ and $\phi(y)$, i.e. image and class embeddings, both of which are given. $F(.)$ is parameterized by the mapping W, that is to be learned. Given an image, compatibility learning frameworks predict the class which attains the maximum compatibility score with the image.

Among the methods that are detailed below, ALE (Akata *et al.*, 2015a), DEVISE (Frome *et al.*, 2013) and SJE (Akata *et al.*, 2015c) do early stopping to implicitly regularize Stochastic Gradient Descent (SGD) while ESZSL (Romera-Paredes *et al.*, 2015) and SAE (Kodirov *et al.*, 2017) explicitly regularize the embedding model as detailed below. In the following, we provide a unified formulation of these five zero-shot learning methods.

DEVISE (Frome *et al.*, 2013) uses pairwise ranking objective that is inspired from unregularized ranking SVM (Joachims, 2002):

$$\sum_{y \in \mathcal{Y}^{tr}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+ \tag{4.4}$$

where $\Delta(y_n, y)$ is equal to 1 if $y_n = y$, otherwise 0. The objective function is convex and is optimized by Stochastic Gradient Descent.

ALE (Akata *et al.*, 2015a) uses the weighted approximate ranking objective (Usunier *et al.*, 2009) for zero-shot learning in the following way:

$$\sum_{y \in \mathcal{Y}^{tr}} \frac{l_{r_{\Delta(x_n, y_n)}}}{r_{\Delta(x_n, y_n)}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+ \tag{4.5}$$

where $l_k = \sum_{i=1}^{k} \alpha_i$ and $r_{\Delta(x_n, y_n)}$ is defined as:

$$\sum_{y \in \mathcal{Y}^{tr}} \mathbb{1}(F(x_n, y; W) + \Delta(y_n, y) \geq F(x_n, y_n; W)) \tag{4.6}$$

Following the heuristic in (Weston *et al.*, 2011), (Akata *et al.*, 2015a) selects $\alpha_i = 1/i$ which puts a high emphasis on the top of the rank list.

SJE (Akata *et al.*, 2015c) gives the full weight to the top of the ranked list and is inspired from the structured SVM (Tsochantaridis *et al.*, 2005):

$$[\max_{y \in \mathcal{Y}^{tr}} (\Delta(y_n, y) + F(x_n, y; W)) - F(x_n, y_n; W)]_+ \tag{4.7}$$

The prediction can only be made after computing the score against all the classifiers, i.e. so as to find the maximum violating class, which makes SJE less efficient than DEVISE and ALE.

ESZSL (Romera-Paredes *et al.*, 2015) applies a square loss to the ranking formulation and adds the following implicit regularization term to the unregularized risk minimization formulation:

$$\gamma \|W\phi(y)\|^2 + \lambda \|\theta(x)^T W\|^2 + \beta \|W\|^2 \tag{4.8}$$

where γ, λ, β are regularization parameters. The first two terms bound the Euclidean norm of projected attributes in the feature space and projected image feature in the attribute space respectively. The advantage of this approach is that the objective function is convex and has a closed form solution.

SAE (Kodirov *et al.*, 2017) also learns the linear projection from image embedding space to class embedding space, but it further constrains that the projection must be able to reconstruct the original image embedding. Similar to the linear auto-encoder, SAE optimizes the following objective:

$$\min_W ||\theta(x) - W^T \phi(y)||^2 + \lambda ||W\theta(x) - \phi(y)||^2, \tag{4.9}$$

where λ is a hyperparameter to be tuned. The optimization problem can be transformed such that Bartels-Stewart algorithm (Bartels and Stewart, 1972) is able to solve it efficiently.

4.3.2 Learning Nonlinear Compatibility

Latent Embeddings (LATEM) (Xian *et al.*, 2016) and Cross Modal Transfer (CMT) (Socher *et al.*, 2013) encode an additional non-linearity component to linear compatibility learning framework.

LATEM (Xian *et al.*, 2016) constructs a piece-wise linear compatibility:

$$F(x, y; W_i) = \max_{1 \leq i \leq K} \theta(x)^T W_i \phi(y) \tag{4.10}$$

where every W_i models a different visual characteristic of the data and the selection of which matrix to use to do the mapping is a latent variable and K is a hyperparameter to be tuned. LATEM uses the ranking loss formulated in Equation 4.4 and Stochastic Gradient Descent as the optimizer.

CMT (Socher *et al.*, 2013) first maps images into a semantic space of words, i.e. class names, where a neural network with tanh nonlinearity learns the mapping:

$$\sum_{y \in \mathcal{Y}^{tr}} \sum_{x \in \mathcal{X}_y} \|\phi(y) - W_1 \tanh(W_2.\theta(x)\|^2 \tag{4.11}$$

where (W_1, W_2) are weights of the two layer neural network. This is followed by a novelty detection mechanism that assigns images to unseen or seen classes. The novelty is detected either via thresholds learned using the embedded images of the seen classes or the outlier probabilities are obtained in an unsupervised way. As zero-shot learning assumes that test images are only from unseen classes, in our experiments when we refer to CMT, that means we do not use the novelty detection component. On the other hand, we name the CMT with novelty detection as CMT* when we apply it to the generalized zero-shot learning setting.

4.3.3 Learning Intermediate Attribute Classifiers

Although Direct Attribute Prediction (DAP) (Lampert *et al.*, 2013) and Indirect Attribute Prediction (IAP) (Lampert *et al.*, 2013) have been shown to perform poorly compared to compatibility learning frameworks (Akata *et al.*, 2015a), we include them to our evaluation for being historically the most widely used methods in the literature.

DAP (Lampert *et al.*, 2013) learns probabilistic attribute classifiers and makes a class prediction by combining scores of the learned attribute classifiers. A novel image is assigned to one of the unknown classes using:

$$f(x) = \operatorname*{argmax}_c \prod_{m=1}^{M} \frac{p(a_m^c|x)}{p(a_m^c)}. \tag{4.12}$$

with M being the total number of attributes, a_m^c is the m-th attribute of class c, $p(a_m^c|x)$ is the attribute probability given image x which is obtained from the attribute

classifiers whereas $p(a_m^c)$ is the attribute prior estimated by the empirical mean of attributes over training classes. We train binary classifiers with logistic regression that gives probability scores of attributes with respect to training classes.

IAP (Lampert *et al.*, 2013) indirectly estimates attributes probabilities of an image by first predicting the probabilities of each training class, then multiplying the class attribute matrix. Once the attributes probabilities are obtained by the following equation:

$$p(a_m|x) = \sum_{k=1}^{K} p(a_m|y_k)p(y_k|x), \tag{4.13}$$

where K is the number of training classes, $p(a_m|y_k)$ is the predefined class attribute and $p(y_k|x)$ is training class posterior from multi-class classifier, the Equation 4.12 is used to predict the class label for which we train a multi-class classifier on training classes with logistic regression.

4.3.4 Hybrid Models

Semantic Similarity Embedding (SSE) (Zhang and Saligrama, 2015), Convex Combination of Semantic Embeddings (CONSE) (Norouzi *et al.*, 2014) and Synthesized Classifiers (SYNC) (Changpinyo *et al.*, 2016) express images and semantic class embeddings as a mixture of seen class proportions, hence we group them as hybrid models.

SSE (Zhang and Saligrama, 2015) leverages similar class relationships both in image and semantic embedding space. An image is labeled with:

$$\underset{u\in\mathcal{U}}{\operatorname{argmax}}\, \pi(\theta(x))^T \psi(\phi(y_u)) \tag{4.14}$$

where π, ψ are mappings of class and image embeddings into a common space defined by the mixture of seen classes proportions. Specifically, ψ is learned by sparse coding and π is by class dependent transformation.

CONSE (Norouzi *et al.*, 2014) learns the probability of a training image belonging to a training class:

$$f(x,t) = \underset{y\in\mathcal{Y}^{tr}}{\operatorname{argmax}}\, p_{tr}(y|x) \tag{4.15}$$

where y denotes the most likely training label (t=1) for image x. Combination of semantic embeddings (s) is used to assign an unknown image to an unseen class:

$$\frac{1}{Z}\sum_{i=1}^{T} p_{tr}(f(x,t)|x).s(f(x,t)) \tag{4.16}$$

where $Z = \sum_{i=1}^{T} p_{tr}(f(x,t)|x)$, $f(x,t)$ denotes the tth most likely label for image x and T controls the maximum number of semantic embedding vectors.

SYNC (Changpinyo *et al.*, 2016) learns a mapping between the semantic class embedding space and a model space. In the model space, training classes and a set of phantom classes form a weighted bipartite graph. The objective is to minimize distortion error:

$$\min_{w_c} \|w_c - \sum_{r=1}^{R} s_{cr} v_r\|_2^2. \tag{4.17}$$

Semantic and model spaces are aligned by embedding classifiers of real classes (w_c) and classifiers of phantom classes (v_r) in the weighted graph (s_{cr}). The classifiers for novel classes are constructed by linearly combining classifiers of phantom classes.

GFZSL (Verm and Rai, 2017) proposes a generative framework for zero-shot learning by modeling each class-conditional distribution as a multi-variate Gaussian with mean vector μ and diagonal covariance matrix σ. While the parameters of seen classes can be estimated by MLE, that of unseen classes are computed by learning the following two regression functions:

$$\mu_y = f_\mu(\phi(y)) \text{ and } \sigma_y = f_\sigma(\phi(y)) \tag{4.18}$$

with an image x, its class is predicted by searching the class with the maximum probability, i.e. $\text{argmax}_y\, p(x|\sigma_y, \mu_y)$.

4.3.5 Transductive Zero-Shot Learning Setting

In zero-shot learning, transductive setting (Chapelle *et al.*, 2009; Zhou *et al.*, 2004) implies that unlabeled images from unseen classes are available during training. Using unlabeled images are expected to improve performance as they possibly contain useful latent information of unseen classes. Here, we mainly focus on two state-of-the-art transductive approaches(Verm and Rai, 2017; Ye and Guo, 2017) and show how to extend ALE (Akata *et al.*, 2015a) into the transductive learning setting.

GFZSL-tran (Verm and Rai, 2017) uses an Expectation-Maximization (EM) based procedure that alternates between inferring the labels of unlabeled examples of unseen classes and using the inferred labels to update the parameter estimates of unseen class distributions. Since the class-conditional distribution is assumed to be Gaussian, this procedure is equivalent to repeatedly estimating a Gaussian Mixture Model (GMM) with the unlabeled data from unseen classes and use the inferred class labels to re-estimate the GMM.

DSRL (Ye and Guo, 2017) proposes to simultaneously learn image features with non-negative matrix factorization and align them with their corresponding class attributes. This step gives us an initial prediction score matrix S_0 in which each row is one instance and indicates the prediction scores for all unseen classes. To improve the prediction score matrix by transductive learning, a graph-based label propagation algorithm is applied. Specifically, a KNN graph is constructed with the

projected instances of unseen classes in the class embedding space,

$$M_{ij} = \begin{cases} exp(-\frac{d(x_i,x_j)}{2\sigma^2}) & \text{if } i \in \text{KNN}(j) \text{ or } j \in \text{KNN}(i) \\ 0 & \text{otherwise} \end{cases} \tag{4.19}$$

where KNN(i) denotes the k-nearest neighbor of i-th instance and $d(x_i, x_j)$ measures the Euclidean distance between x_i and x_j. Given the affinity matrix M, a normalized Laplacian matrix L can be computed as $L = Q^{-1/2}MQ^{-1/2}$ where Q is a diagonal matrix with $Q_{ii} = \sum_j M_{ij}$. Finally, the standard label propagation (**?**) gives the closed-form solution:

$$S = (I - \alpha L)^{-1} \times S_0 \tag{4.20}$$

where $\alpha \in [0, 1]$ is a regularization trade-off parameter and S is the score matrix. The class label of an instance is predicted by searching the class with the highest score, i.e. $\text{argmax}_y S_{iy}$.

ALE-tran Any compatibility learning method that explicitly learns cross-modal mapping from image feature space to class embedding space can be extended to transductive setting following the label propagation procedure of DSRL (Ye and Guo, 2017). Taking the ALE (Akata *et al.*, 2015a) as an example, after learning the linear mapping W, instances of unseen classes can be projected into the class embedding space and a score matrix S_0 can be computed similarly.

4.4 DATASETS

Among the most widely used datasets for zero-shot learning, we select two coarse-grained, one small (aPY (Farhadi *et al.*, 2009)) and one medium-scale (AWA1 (Lampert *et al.*, 2013)), and two fine-grained, both medium-scale, datasets (SUN (Patterson and Hays, 2012), CUB (Welinder *et al.*, 2010)) with attributes and one large-scale dataset (ImageNet (Deng *et al.*, 2009)) without. Here, we consider between $10K$ and $1M$ images, and, between 100 and $1K$ classes as medium-scale. Details of dataset statistics in terms of the number of images, classes, attributes for the attribute datasets are in Table 5.1. Furthermore, we introduce our Animals With Attributes 2 (AWA2) dataset and position it with respect to existing datasets.

4.4.1 Attribute Datasets

Attribute Pascal and Yahoo (aPY) (Farhadi *et al.*, 2009) is a small-scale coarse-grained dataset with 64 attributes. Among the total number of 32 classes, 20 Pascal classes are used for training (we randomly select 5 for validation) and 12 Yahoo classes are used for testing. The original Animals with Attributes (AWA1) (Lampert *et al.*, 2013) is a coarse-grained dataset that is medium-scale in terms of the number of images, i.e. 30,475 and small-scale in terms of number of classes, i.e. 50 classes. (Lampert *et al.*, 2013) introduces a standard zero-shot split with 40 classes for training (we randomly

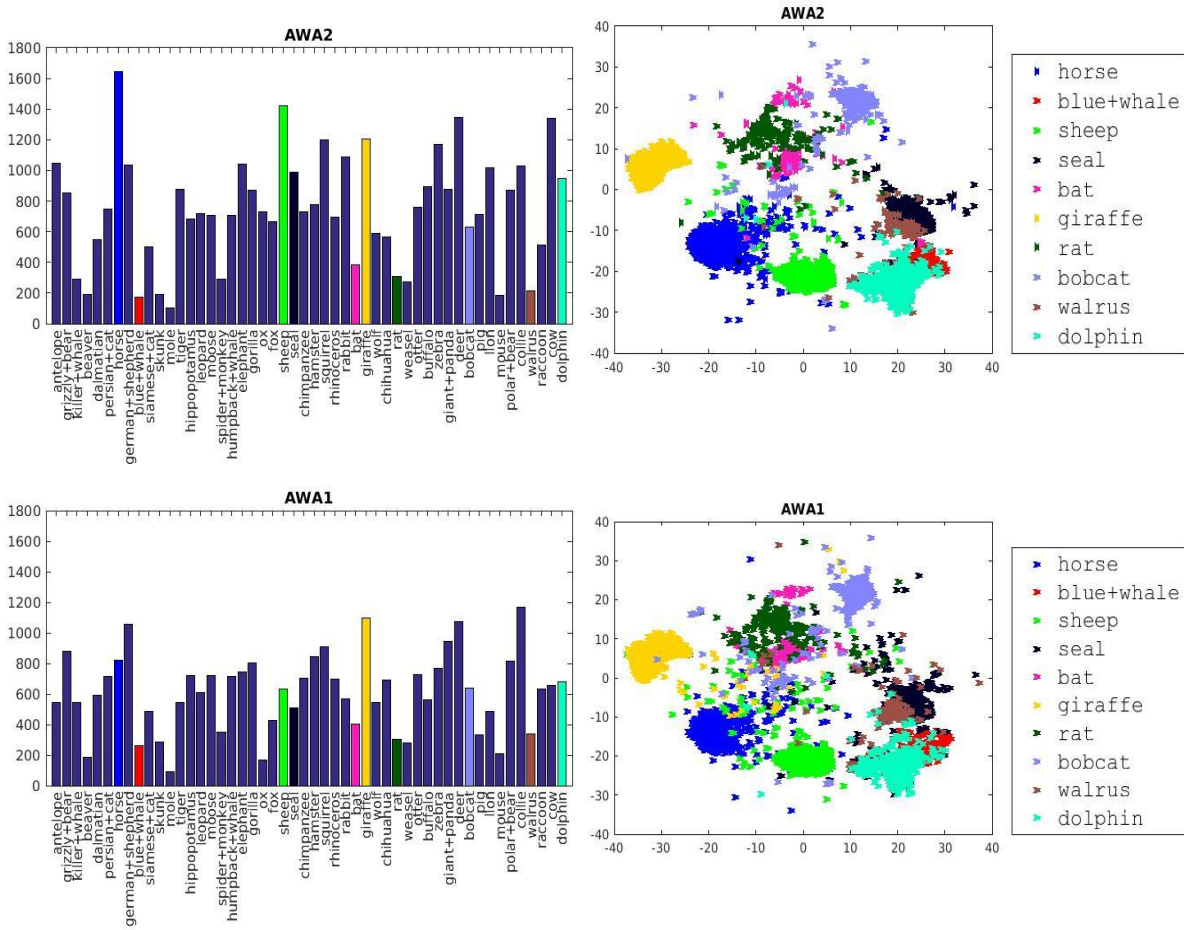

Figure 4.2: Comparing AWA1 (Lampert *et al.*, 2013) and our AWA2 in terms of number of images (Left) and t-SNE embedding of the image features (the embedding is learned on AWA1 and AWA2 simultaneously, therefore the figures are comparable). AWA2 follows a similar distribution as AWA1 and it contains more examples.

select 13 classes for validation) and 10 classes for testing. AWA1 has 85 attributes. Caltech-UCSD-Birds 200-2011 (CUB) (Welinder *et al.*, 2010) is a fine-grained and medium scale dataset with respect to both number of images and number of classes, i.e. 11,788 images from 200 different types of birds annotated with 312 attributes. (Akata *et al.*, 2015a) introduces the first zero-shot split of CUB with 150 training (50 validation classes) and 50 test classes. SUN (Patterson and Hays, 2012) is a fine-grained and medium-scale dataset with respect to both number of images and number of classes, i.e. SUN contains 14340 images coming from 717 types of scenes annotated with 102 attributes. Following (Lampert *et al.*, 2013) we use 645 classes of SUN for training (we randomly select 65 classes for validation) and 72 classes for testing.

Animals with Attributes2 (AWA2) Dataset. One disadvantage of AWA1 dataset

is that the images are not publicly available. As having highly descriptive image features is an important component for zero-shot learning, in order to enable vision research on the objects of the AWA1 dataset, we introduce the Animals with Attributes2 (AWA2) dataset. Following (Lampert *et al.*, 2013), we collect 37,322 images for the 50 classes of AWA1 dataset from public web sources, i.e. Flickr, Wikipedia, etc., making sure that all images of AWA2 have free-use and redistribution licenses and they do not overlap with images of the original Animal with Attributes dataset. The AWA2 dataset uses the same 50 animal classes as AWA1 dataset, similarly the 85 binary and continuous class attributes are common. In total, AWA2 has 37,322 images compared to 30,475 images of AWA1. On average, each class includes 746 images where the least populated class, i.e. mole, has 100 and the most populated class, i.e. horse has 1645 examples. Some example images from *polar bear, zebra, otter* and *tiger* classes along with sample attributes from our AWA2 dataset are shown in Figure 8.1.

In Figure 4.2, we provide some statistics on the AWA2 dataset in comparison with the AWA1 dataset in terms of the number of images and also the distribution of the image features. Compared to AWA1, our proposed AWA2 dataset contains more images, e.g. *horse* and *dolphin* among the test classes, *antelope* and *cow* among the training classes. Moreover, the t-SNE embedding of these test classes with more training data, e.g. *horse*, *dolphin*, *seal* etc. shows that AWA2 leads to slightly more visible clusters of ResNet features. The images, their labels and ResNet features of our AWA2 are publicly available in http://cvml.ist.ac.at/AwA2.

4.4.2 Large-Scale ImageNet

We also evaluate the performance of methods on the large scale ImageNet (Deng *et al.*, 2009) which contains a total of 14 million images from 21K classes, each one labeled with one label, and the classes are hierarchically related as ImageNet follows the WordNet (Miller, 1995).

ImageNet is a natural fit for zero-shot and generalized zero-shot learning as there is a large class imbalance problem. Moreover, ImageNet is diverse in terms of granularity, i.e. it contains a collection of fine-grained datasets, e.g. different vehicle types, as well as coarse-grained datasets. The highest populated class contains 3,047 images whereas there are many classes that contains only a single image. A balanced subset of ImageNet with 1K classes containing about 1000 images each is used to train CNNs.

Previous works (Rohrbach *et al.*, 2011) proposed to split the balanced subset of 1K classes into 800 training and 200 test classes. In this work, from the total of 21K classes, we use 1K classes for training (among which we use 200 classes for validation) and the test split is either all the remaining 20K classes or a subset of it, e.g. we determine these subsets based on the hierarchical distance between classes and the population of classes. The details of these splits are provided in the following section.

		Number of Classes			Number of Images								
						At Training Time				At Evaluation Time			
						SS		PS		SS		PS	
Dataset	**Att**	$\mathcal{Y}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	**Total**	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$
SUN	102	717	580 + 65	72	14340	12900	0	10320	0	0	1440	2580	1440
CUB	312	200	100 + 50	50	11788	8855	0	7057	0	0	2933	1764	2967
AWA1	85	50	27 + 13	10	30475	24295	0	19832	0	0	6180	4958	5685
AWA2	85	50	27 + 13	10	37322	30337	0	23527	0	0	6985	5882	7913
aPY	64	32	15 + 5	12	15339	12695	0	5932	0	0	2644	1483	7924

Table 4.1: Statistics for SUN (Patterson and Hays, 2012), CUB (Welinder *et al.*, 2010), AWA1 (Lampert *et al.*, 2013), proposed AWA2, aPY (Farhadi *et al.*, 2009) in terms of size, granularity, number of attributes, number of classes in $\mathcal{Y}^{tr}$ and $\mathcal{Y}^{ts}$, number of images at training and test time for standard split (SS) and our proposed splits (PS).

4.5 EVALUATION PROTOCOL

In this section, we provide several components of previously used and our proposed ZSL and GZSL evaluation protocols, e.g. image and class encodings, dataset splits and the evaluation criteria[3].

4.5.1 Image and Class Embedding

We extract image features, namely image embeddings, from the entire image for SUN, CUB, AWA1, our AWA2 and ImageNet, with no image pre-processing. For aPY, following the original publication in (Farhadi *et al.*, 2009), we crop the images from bounding boxes. Our image embeddings are 2048-dim top-layer pooling units of the 101-layered ResNet (He *et al.*, 2016) as we found that it performs better than 1,024-dim top-layer pooling units of GoogleNet (Szegedy *et al.*, 2015). We use the original ResNet-101 that is pre-trained on ImageNet with 1K classes, i.e. the balanced subset, and we do not fine-tune it for any of the mentioned datasets. In addition to the ResNet features, we re-evaluate all methods with their published image features.

In zero-shot learning, class embeddings are as important as image features. As class embeddings, for aPY, AWA1, AWA2, CUB and SUN, we use the per-class attributes between values 0 and 1 that are provided with the datasets as binary attributes have been shown (Akata *et al.*, 2015a) to be weaker than continuous attributes. For ImageNet as attributes of 21K classes are not available, we use Word2Vec (Mikolov *et al.*, 2013b) trained on Wikipedia provided by (Changpinyo *et al.*, 2016). Note that an evaluation of class embeddings is out of the scope of this chapter. We refer the reader to (Akata *et al.*, 2015c) for more details on the topic.

[3]Our benchmark is in: http://www.mpi-inf.mpg.de/zsl-benchmark

4.5.2 Dataset Splits

Zero-shot learning assumes disjoint training and test classes. Hence, as deep neural network (DNN) training for image feature extraction is actually a part of model training, the dataset used to train DNNs, e.g. ImageNet, should not include any of the test classes. However, we notice from the standard splits (SS) of aPY and AWA1 datasets that 7 aPY test classes out of 12 (monkey, wolf, zebra, mug, building, bag, carriage), 6 AWA1 test classes out of 10 (chimpanzee, giant panda, leopard, persian cat, pig, hippopotamus), are among the 1K classes of ImageNet, i.e. are used to pre-train ResNet. On the other hand, the mostly widely used splits, i.e. we term them as standard splits (SS), for SUN from (Lampert *et al.*, 2013) and CUB from (Akata *et al.*, 2013) shows us that 1 CUB test class out of 50 (Indigo Bunting), and 6 SUN test classes out of 72 (restaurant, supermarket, planetarium, tent, market, bridge), are also among the 1K classes of ImageNet.

We noticed that the accuracy for all methods on those overlapping test classes are higher than others. Therefore, we propose new dataset splits, i.e. proposed splits (PS), insuring that none of the test classes appear in ImageNet 1K, i.e. used to train the ResNet model. We present the differences between the standard splits (SS) and the proposed splits (PS) in Table 5.1. While in SS and PS no image from test classes is present at training time, at test time our PS includes images from training classes. We designed the PS this way as evaluating accuracy on both training and test classes is crucial to show the generalization of the methods.

For SUN, CUB, AWA1, aPY, and our proposed AWA2 dataset, for measuring the significance of the results, we propose 3 different splits of 580, 100, 27, 15 and 27 training classes respectively while keeping 72, 50, 10, 12 and 10 test classes the same. It is important to perform hyperparameter search on a disjoint set of validation set of 65, 50, 13, 5 and 13 classes respectively. We keep the number of classes the same for SS and PS, however we choose different classes while making sure that the test classes do not overlap with the 1K training classes of ImageNet.

ImageNet provides possibilities of constructing several zero-shot evaluation splits. Following (Changpinyo *et al.*, 2016), our first two standard splits consider all the classes that are 2-hops and 3-hops away from the original 1K classes according to the ImageNet label hierarchy, corresponding to 1509 and 7678 classes. This split measures the generalization ability of the models with respect to the hierarchical and semantic similarity between classes. As discussed in the previous section, another characteristic of ImageNet is the imbalanced sample size. Therefore, our proposed split considers 500, 1K and 5K most populated classes among the remaining 21K classes of ImageNet with approximately 1756, 1624 and 1335 images per class on average. Similarly, we consider 500, 1K and 5K least-populated classes in ImageNet which correspond to most fine-grained subsets of ImageNet with approximately 1, 3 and 51 images per class on average. We measure the generalization of methods to the entire ImageNet data distribution by considering a final split of all the remaining approximately 20K classes of ImageNet with at least 1 image per-class, i.e. approximately 631 images per class on average.

4.5.3 Evaluation Criteria

Single label image classification accuracy has been measured with Top-1 accuracy, i.e. the prediction is accurate when the predicted class is the correct one. If the accuracy is averaged for all images, high performance on densely populated classes is encouraged. However, we are interested in having high performance also on sparsely populated classes. Therefore, we average the correct predictions independently for each class before dividing their cumulative sum w.r.t the number of classes, i.e. we measure average per-class top-1 accuracy in the following way:

$$acc_{\mathcal{Y}} = \frac{1}{\|\mathcal{Y}\|} \sum_{c=1}^{\|\mathcal{Y}\|} \frac{\text{\# correct predictions in c}}{\text{\# samples in c}} \tag{4.21}$$

In the generalized zero-shot learning setting, the search space at evaluation time is not restricted to only test classes ($\mathcal{Y}^{ts}$), but includes also the training classes ($\mathcal{Y}^{tr}$), hence this setting is more practical. As with our proposed split at test time we have access to some images from training classes, after having computed the average per-class top-1 accuracy on training and test classes, we compute the harmonic mean of training and test accuracies:

$$H = \frac{2 * acc_{\mathcal{Y}^{tr}} * acc_{\mathcal{Y}^{ts}}}{acc_{\mathcal{Y}^{tr}} + acc_{\mathcal{Y}^{ts}}} \tag{4.22}$$

where $acc_{\mathcal{Y}^{tr}}$ and $acc_{\mathcal{Y}^{ts}}$ represent the accuracy of images from seen ($\mathcal{Y}^{tr}$), and images from unseen ($\mathcal{Y}^{ts}$) classes respectively. We choose harmonic mean as our evaluation criteria and not arithmetic mean because in arithmetic mean if the seen class accuracy is much higher, it effects the overall results significantly. Instead, our aim is high accuracy on both seen and unseen classes.

	SUN		CUB		AWA1		aPY	
Model	**R**	**O**	**R**	**O**	**R**	**O**	**R**	**O**
DAP	22.1	22.2	–	–	41.4	41.4	19.1	19.1
SSE	83.0	82.5	44.2	30.4	64.9	76.3	45.7	46.2
LATEM	–	–	45.1	45.5	71.2	71.9	–	–
SJE	–	–	50.1	50.1	67.2	66.7	–	–
ESZSL	64.3	65.8	–	–	48.0	49.3	14.3	15.1
SYNC	62.8	62.8	53.4	53.4	69.7	69.7	–	–
SAE	–	–	–	–	84.7	84.7	–	–
GFZSL	86.5	86.5	56.6	56.5	80.4	80.8	–	–
GFZSL-tran	87.0	87.0	63.8	63.7	94.9	94.3	–	–
DSRL	86.0	85.4	57.6	57.1	87.7	87.2	47.8	51.3

Table 4.2: Reproducing zero-shot results with methods that have a public implementation: O = Original results, R = Reproduced using provided image features and code. We measure top-1 accuracy in %. –: image features are not provided in the original paper for this dataset. Top: ZSL, Bottom: transductive ZSL.

4.6 EXPERIMENTS

We first provide ZSL results on the attribute datasets SUN, CUB, AWA1, AWA2 and aPY and then on the large-scale ImageNet dataset. Finally, we present results for the GZSL setting.

4.6.1 Zero-Shot Learning Experiments

On attribute datasets, i.e. SUN, CUB, AWA1, AWA2, and aPY, we first reproduce the results of each method using their evaluation protocol, then provide a unified evaluation protocol using the same train/val/test class splits, followed by our proposed train/val/test class splits on SUN, CUB, AWA1, aPY and AWA2. We also evaluate the robustness of the methods to parameter tuning and visualize the ranking of different methods. Finally, we evaluate the methods on the large-scale ImageNet dataset.

Comparing State-of-The-Art Models. For sanity-check, we re-evaluate methods (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016) and (Kodirov *et al.*, 2017) using publicly available features and code from the original publication on SUN, CUB, AWA1 and aPY (CMT (Socher *et al.*, 2013) evaluates on CIFAR dataset.). We observe from the results in Table 4.2 that our reproduced results of DAP(Lampert *et al.*, 2013), SYNC (Changpinyo *et al.*, 2016), GFZSL (Verm and Rai, 2017), GFZSL-tran (Verm and Rai, 2017), DSRL (Ye and Guo, 2017) and SAE (Kodirov *et al.*, 2017) are nearly identical

to the reported number in their original publications. For LATEM (Xian *et al.*, 2016), we obtain slightly different results which can be explained by the non-convexity and thus the sensibility to initialization. Similarly for SJE (Akata *et al.*, 2015c) random sampling in SGD might lead to slightly different results. ESZSL (Romera-Paredes *et al.*, 2015) has some variance because its algorithm randomly picks a validation set during each run, which leads to different hyperparameters. Notable observations on SSE (Zhang and Saligrama, 2015) results are as follows. The published code has hard-coded hyperparameters operational on aPY, i.e. number of iterations, number of data points to train SVM, and one regularizer parameter γ which lead to inferior results than the ones reported here, therefore we set these parameters on validation sets. On SUN, SSE uses 10 classes (instead of 72) and our results with validated parameters got an improvement of 0.5% that may be due to random sampling of training images. On AWA1, our reproduced result being 64.9% is significantly lower than the reported result (76.3%). However, we could not reach the reported result even by tuning parameters on the test set (73.8%).

In addition to (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Kodirov *et al.*, 2017), we re-implement (Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a) based on the original publications. We use train, validation, test splits as provided in Table 5.1 and report results in Table 4.3 with deep ResNet features. DAP (Lampert *et al.*, 2013) uses hand-crafted image features and thus reproduced results with those features are significantly lower than the results with deep features (22.1% vs 38.9%). When we investigate the results in detail, we noticed two irregularities with reported results on SUN. First, SSE (Zhang and Saligrama, 2015) and ESZSL (Romera-Paredes *et al.*, 2015) report results on a test split with 10 classes whereas the standard split of SUN contains 72 test classes (74.5% vs 54.5% with SSE (Zhang and Saligrama, 2015) and 64.3% vs 57.3% with ESZSL (Romera-Paredes *et al.*, 2015)). Second, after careful examination and correspondence with the authors of SYNC (Changpinyo *et al.*, 2016), we detected that SUN features were extracted with a MIT Places (Zhou *et al.*, 2014) pre-trained model. As the MIT Places dataset intersects with both training and test classes of SUN, it is expected to lead to significantly better results than ImageNet pre-trained models (62.8% vs 59.1%). In addition, while SAE (Kodirov *et al.*, 2017) reported 84.7% on AWA1, we obtain only 80.7% on the standard split. This could be explained by two differences. First, we measure per-class accuracy but SAE (Kodirov *et al.*, 2017) reports per-image accuracy which is typically higher when the dataset is class-imbalanced, e.g. AWA1. Indeed, their reported accuracy decreases to 82.0% if per-class accuracy is applied. Second, we confirmed with the authors of SAE (Kodirov *et al.*, 2017) that they improved GoogleNet (Szegedy *et al.*, 2015) by adding Batch Normalization and averaging 5 randomly cropped images to obtain better image features. Therefore, as expected, improving visual features lead to improved results in zero-shot learning.

Promoting Our Proposed Splits (PS). We propose new dataset splits (see details in section 4.4) ensuring that test classes of any of the datasets do not overlap with the ImageNet1K used to pre-train ResNet. As training ResNet is a part of the training

	SUN		CUB		AWA1		AWA2		aPY	
Method	**SS**	**PS**	**SS**	**PS**	**SS**	**PS**	**SS**	**PS**	**SS**	**PS**
DAP	38.9	39.9	37.5	40.0	57.1	44.1	58.7	46.1	35.2	33.8
IAP	17.4	19.4	27.1	24.0	48.1	35.9	46.9	35.9	22.4	36.6
CONSE	44.2	38.8	36.7	34.3	63.6	45.6	67.9	44.5	25.9	26.9
CMT	41.9	39.9	37.3	34.6	58.9	39.5	66.3	37.9	26.9	28.0
SSE	54.5	51.5	43.7	43.9	68.8	60.1	67.5	61.0	31.1	34.0
LATEM	56.9	55.3	49.4	49.3	74.8	55.1	68.7	55.8	34.5	35.2
ALE	59.1	58.1	53.2	54.9	78.6	59.9	80.3	62.5	30.9	39.7
DEVISE	57.5	56.5	53.2	52.0	72.9	54.2	68.6	59.7	35.4	**39.8**
SJE	57.1	53.7	**55.3**	53.9	76.7	65.6	69.5	61.9	32.0	32.9
ESZSL	57.3	54.5	55.1	53.9	74.7	58.2	75.6	58.6	34.4	38.3
SYNC	59.1	56.3	54.1	**55.6**	72.2	54.0	71.2	46.6	39.7	23.9
SAE	42.4	40.3	33.4	33.3	**80.6**	53.0	**80.7**	54.1	8.3	8.3
GFZSL	**62.9**	**60.6**	53.0	49.3	80.5	**68.3**	79.3	**63.8**	**51.3**	38.4

Table 4.3: Zero-shot learning results on SUN, CUB, AWA1, AWA2 and aPY using SS = Standard Split, PS = Proposed Split with ResNet features. The results report top-1 accuracy in %.

procedure, including test classes in the dataset used for pre-training ResNet would violate the zero-shot learning conditions. We compare the results obtained with our proposed split (PS) with previously published standard split (SS) results in Table 4.3.

Our first observation is that the results on the PS are significantly lower than the SS for AWA1 and AWA2. This is expected as most of the test classes of AWA1 and AWA2 in SS overlaps with ImageNet 1K. On the other hand, for fine-grained datasets CUB and SUN, the results are not significantly effected as the overlap in that case was not as significant. Our second observation regarding the method ranking is as follows. On SS, SYNC (Changpinyo *et al.*, 2016) is the best performing method on SUN (59.1%) and aPY (39.7%) datasets whereas SJE (Akata *et al.*, 2015c) performs the best on CUB (55.3%) and SAE (Kodirov *et al.*, 2017) performs the best on AWA1 (80.6%) and AWA2 (80.7%) dataset. On PS, ALE (Akata *et al.*, 2015a) performs the best on SUN (58.1%) and AWA2 (62.5%), SYNC (Changpinyo *et al.*, 2016) on CUB (55.6%), SJE (Akata *et al.*, 2015c) on AWA1 (65.6%) and DEVISE (Frome *et al.*, 2013) on aPY (39.8%). ALE, SJE and DEVISE all use max-margin bi-linear compatibility learning framework which seem to perform better than others. It is also worth to note that SYNC and SAE perform well on SS, i.e. SYNC is the best performing model for SUN and aPY whereas SAE is for AWA1 and AWA2 on SS, while they perform significantly lower in PS which indicates that they do not generalize well in zero-shot learning task.

Evaluating Robustness. We evaluate robustness of 13 methods, i.e. (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-

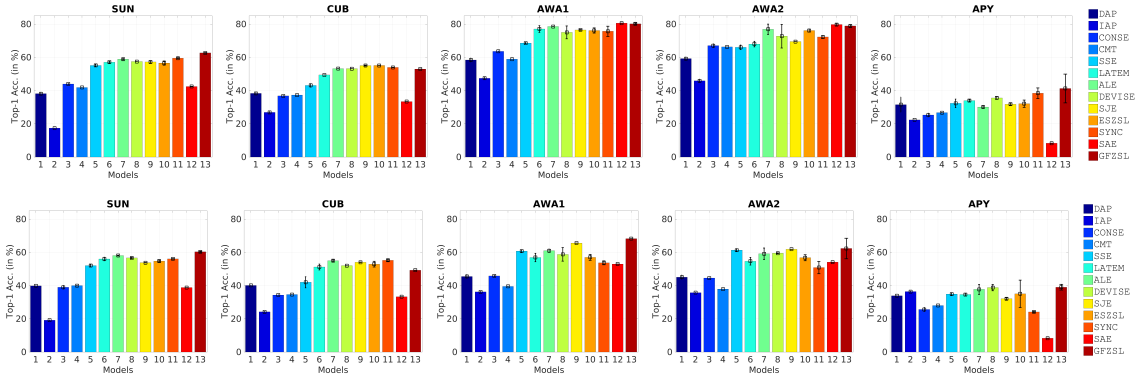

Figure 4.3: Robustness of 10 methods evaluated on SUN, CUB, AWA1, aPY using 3 validation set splits (results are on the same test split). Top: original split, Bottom: proposed split (Image embeddings = ResNet). We measure top-1 accuracy in %.

Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a; Kodirov *et al.*, 2017; Verm and Rai, 2017), to hyperparameters by setting them on 3 different validation splits while keeping the test split intact. We report results on SS (Figure 4.3, top) and PS (Figure 4.3, bottom) for SUN, CUB, AWA1, AWA2 and aPY datasets. On SUN and CUB, the results are stable across methods and across dataset splits. This is expected as these datasets both have a balanced number of images across classes and they are fine-grained datasets. Therefore, the validation splits are similar. On the other hand, aPY being a small and coarse-grained dataset has several issues. First, many of the test classes of aPY are included in ImageNet1K. Second, it is not well balanced, i.e. different validation class splits contain significantly different number of images. Third, the class embeddings are far from each other, i.e. objects are semantically different, therefore different validation splits learn a different mapping between images and classes. On AWA1 and AWA2, on SS, the DEVISE method seems to show the largest variance. This might be due to the fact that AWA1 and AWA2 datasets are also coarse-grained and test classes overlap with ImageNet training classes. Indeed, AWA2 being slightly more balanced than AWA1, in the proposed split it does not lead to such a high variance for DEVISE.

Visualizing Method Ranking. We first evaluate the 13 methods using three different validation splits as in the previous experiment. We then rank them based on their per-class top-1 accuracy using the non-parametric Friedman test (Garcia and Herrera, 2008), which does not assume a distribution on performance but rather uses algorithm ranking. Each entry of the rank matrix on Figure 4.4 indicates the number of times the method is ranked at the first to thirteenth rank. We then compute the mean rank of each method and order them based on the mean rank across datasets.

Our general observation is that the highest ranked method on both splits is GFZSL, the second highest ranked method on the standard split (SS) is SYNC while it drops to the seventh rank on the proposed split (PS). On the other hand, ALE ranks the second on the SS and the first on the PS. We reinforce our initial observation

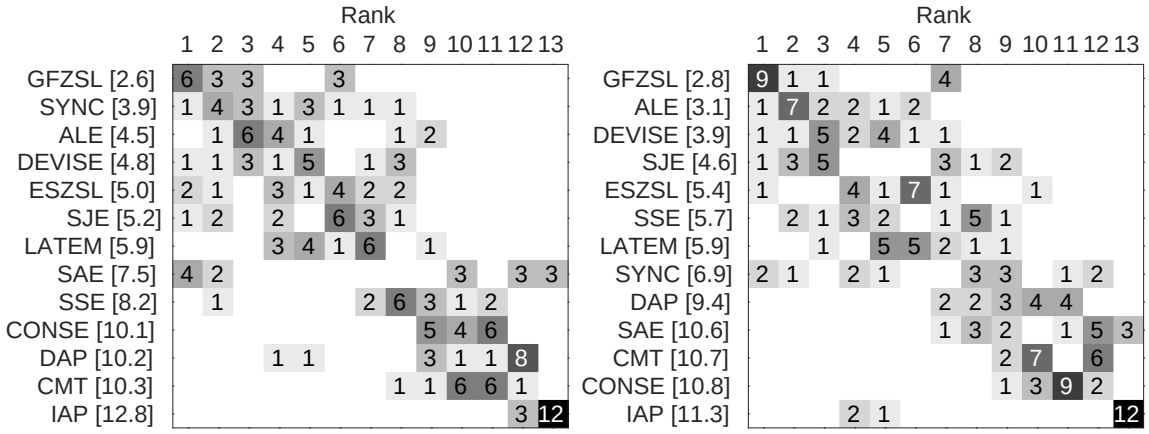

Figure 4.4: Ranking 12 models by setting parameters on three validation splits on the standard (SS, left) and proposed (PS, right) setting. Element (i, j) indicates number of times model i ranks at jth over all 4×3 observations. Models are ordered by their mean rank (displayed in brackets).

from numerical results and conclude that GFZSL and ALE seems to be the method that is the most robust in zero-shot learning setting for attribute datasets. These results also indicate the importance of choosing zero-shot splits carefully. On the PS, the two of three highest ranked methods are compatibility learning methods, i.e. ALE and DEVISE whereas the three lowest ranked methods are attribute classifier learning or hybrid methods, i.e. IAP, CMT and CONSE. Therefore, max-margin compatibility learning methods lead to consistently better results in the zero-shot learning task compared to learning independent classifiers. Finally, visualizing the method ranking in this way provides a visually interpretable way of how models compare across datasets.

Results on Our Proposed AWA2. We introduce AWA2 which has the same classes and attributes as AWA1, but contains different images each coming with a public copyright license. In order to show that AWA1 and AWA2 images are not the same but similar in nature, we compare the zero-shot learning results on AWA1 and AWA2 in Table. 4.3. Under the Standard Splits (SS), SAE (Kodirov *et al.*, 2017) is the best performing method on both AWA1 (80.6%) and AWA2 (80.7%). Similarly, for most of the methods, the results on AWA1 are close to those on AWA2, for instance, DAP obtains 57.1% on AWA1 and 58.7% on AWA2, SSE obtains 68.8% on AWA1 and 67.5% AWA2, etc. The results under the Proposed Splits (PS) are also consistent across AWA1 and AWA2. For 8 out of 12 methods, the performance difference between AWA1 and AWA2 is within 2%. On the other hand, the same consistency is not observed for DEVISE (Frome *et al.*, 2013), SJE (Akata *et al.*, 2015c) and SYNC (Changpinyo *et al.*, 2016). For instance, SJE (Akata *et al.*, 2015c) obtains 65.6% on AWA1 and 61.9% on AWA2. After careful examination, we noticed that SJE (Akata *et al.*, 2015c) selects different hyperparameters for AWA1 and AWA2, which results in different performance on those two datasets. In our opinion, this does not indicate a possible dataset artifact, however shows that zero-shot learning

	Training Set : Test Set			
Method	**AWA1:AWA1**	**AWA1:AWA2**	**AWA2:AWA2**	**AWA2:AWA1**
DAP	44.1	44.2	46.1	46.2
IAP	35.9	36.1	35.9	35.3
CONSE	45.6	46.5	44.5	43.7
CMT	39.5	40.7	37.9	37.7
SSE	60.1	61.6	61.0	59.8
LATEM	55.1	55.4	55.8	53.5
ALE	59.9	59.9	62.5	60.9
DEVISE	54.2	55.2	59.7	57.7
SJE	65.6	65.5	61.9	62.0
ESZSL	58.2	58.5	58.6	59.9
SYNC	54.0	53.7	46.6	46.9
SAE	53.0	52.4	54.1	53.1

Table 4.4: Cross-dataset evaluation over AWA1 and AWA2 in zero-shot learning setting on the Proposed Splits: Left of the colon indicates the training set and right of the colon indicates the test set, e.g. AWA1:AWA2 means that the model is trained on the train set of AWA1 and evaluated on the test set of AWA2. We measure top-1 accuracy in %.

is sensitive to parameter setting.

Commonly, a model is trained and evaluated on the same dataset. Across dataset experiments are not easy as different datasets do not share the same attributes. However, AWA1 and AWA2 share both classes and attributes. In order to verify that AWA2 is a good replacement for AWA1, we conduct across-dataset evaluation for 12 methods, i.e. (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a; Kodirov *et al.*, 2017). In particular, with our Proposed Splits (PS), we train one model on the training set of AWA1 and evaluate it on the test set of AWA2 in the zero-shot learning setting, and vice versa. From Table. 4.4, we observe that all the models trained on AWA1 generalize well to AWA2 and vice versa.

In addition, we notice that the cross-dataset result is dependent on the training set. For instance, for all the methods, if we fix training set to be from AWA1, the results on the test set of AWA1 and AWA2 are close. To verify this hypothesis, we performed a paired t-test which determines if the mean difference between paired results is significantly higher than zero. To that end, we take the 24 pairs of results whose test sets are the same, i.e. the results obtained with 12 methods on AWA1:AWA2 and AWA2:AWA2 (2nd and 3rd column) as well as the results obtained with 12 methods on AWA1:AWA1 and AWA2:AWA1 (1st and 4th column). The paired t-test rejects the null hypothesis with p-value$= 0.007$, indicating that the results are significantly different if the test set is the same but the training set is different. As a conclusion, the training set is an important indicator of the final result and the two datasets, i.e.

	Hierarchy		Most Populated			Least Populated			All
Method	**2 H**	**3 H**	**500**	**1K**	**5K**	**500**	**1K**	**5K**	**20K**
CONSE	7.63	2.18	12.33	8.31	3.22	3.53	2.69	1.05	0.95
CMT	2.88	0.67	5.10	3.04	1.04	1.87	1.08	0.33	0.29
LATEM	5.45	1.32	10.81	6.63	1.90	4.53	2.74	0.76	0.50
ALE	5.38	1.32	10.40	6.77	2.00	4.27	2.85	0.79	0.50
DEVISE	5.25	1.29	10.36	6.68	1.94	4.23	2.86	0.78	0.49
SJE	5.31	1.33	9.88	6.53	1.99	4.93	2.93	0.78	0.52
ESZSL	6.35	1.51	11.91	7.69	2.34	4.50	3.23	0.94	0.62
SYNC	**9.26**	**2.29**	**15.83**	**10.75**	**3.42**	**5.83**	**3.52**	**1.26**	**0.96**
SAE	4.89	1.26	9.96	6.57	2.09	2.50	2.17	0.72	0.56
GFZSL	1.45	−−	2.01	1.35	−−	1.40	1.11	0.13	−−

Table 4.5: ImageNet with different splits: 2/3 H = classes with 2/3 hops away from the $\mathcal{Y}^{tr}$ of ImageNet1K, 500/1K/5K most populated classes, 500/1K/5K least populated classes, All = The remaining 20K categories of ImageNet ($\mathcal{Y}^{ts}$). We measure top-1 accuracy in %.

AWA1 and AWA2 are sufficiently similar. Therefore, our cross-dataset experimental results indicate that AWA2 is a good replacement for AWA1.

Zero-Shot Learning Results on ImageNet. ImageNet scales the methods to a truly large-scale setting, thus these experiments provide further insights on how to tackle the zero-shot learning problem from the practical point of view. Here, we evaluate 10 methods, i.e. (Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a; Kodirov *et al.*, 2017; Verm and Rai, 2017). We exclude DAP and IAP as attributes are not available for all ImageNet classes as well as SSE (Zhang and Saligrama, 2015) due to scalability issues of the public implementation of the method. Table 4.5 shows that the best performing method is SYNC (Changpinyo *et al.*, 2016) which may either indicate that it performs well in large-scale setting or it can learn under uncertainty due to usage of Word2Vec instead of attributes. Another possibility is Word2Vec may be tuned for SYNC as it is provided by the same authors. However, we refrain to make a strong claim as this would requires a full evaluation on class embeddings which is out of the scope of this chapter. On the other hand, GFZSL (Verm and Rai, 2017) which is the best performing model for attribute datasets perform poorly on ImageNet which may indicate that generative models require a strong class embedding space such as attributes to perform well on ZSL task. Note that due to the computational issues, we were not able to obtain results for GFZSL for 3H, M5K, L5K and All 20K classes.

More detailed observations are as follows. The second highest performing method is ESZSL (Romera-Paredes *et al.*, 2015) which is one of the linear embedding models that have an implicit regularization mechanism, which seems to be more effective than early stopping as an explicit regularizer. A general observation from the results of all the methods is that in the most populated classes, the results are

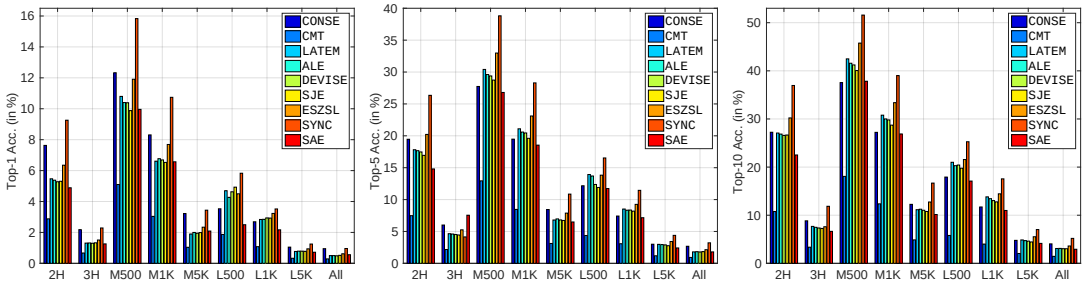

Figure 4.5: Zero-Shot Learning experiments on Imagenet, measuring Top-1, Top-5 and Top-10 accuracy. 2/3 H = classes with 2/3 hops away from ImageNet1K training classes ($\mathcal{Y}^{tr}$), M500/M1K/M5K denote 500, 1K and 5K most populated classes, L500/L1K/L5K denote 500, 1K and 5K least populated classes, All = The remaining 20K categories of ImageNet.

higher than the least populated classes which indicates that zero-shot learning on fine-grained ImageNet subsets is a more difficult task. Moreover, we conclude that the nature of the test set, e.g. type of the classes being tested, is more important than the number of classes. Therefore, the selection of the test set is an important aspect of zero-shot learning on large-scale datasets. Furthermore, for all methods we consistently observe a large drop in accuracy between 1K and 5K most populated classes which is expected as 5K contains $\approx$ 6.6M images, making the problem much more difficult than 1K ($\approx$ 1624 images). It is worth to note that, measuring per-image accuracy in this case would lead to higher results if the labels of the highly populated class samples are predicted correctly. Finally, the largest test set, i.e. All 20K, the results are poor for all methods which indicates the difficulty of this problem where there is a large room for improvement.

Several models in the literature evaluate Top-5 and Top-10 as well as Top-1 accuracy on ImageNet. Top-5 and Top-10 accuracy in this case is reasonable as an image usually contains multiple objects however by construction it is associated with a single label in ImageNet. Hence, we provide a comparison of the same 9 models according to all these three criteria in Figure 4.5. We observe that SYNC (Changpinyo *et al.*, 2016) performs significantly better than other methods when the number of images is higher, e.g. 2H, M500, M1K, whereas the gap reduces when the number of images and the number of classes increase, e.g. 3H, L5K and All. In fact, when for All, all the methods perform similarly and poorly which indicates that there is a large room for improvement in this task. In fact, this observation carries on for all three accuracy measures. For Top-5 (middle) and Top-10 (right) accuracy although the numbers are as expected in general higher, the winning model remains as SYNC, significantly for 2H, M500 and M1K whereas the difference is smaller with 3H, L5H, L1K. On the other hand, all methods perform similarly when all 20K classes are tested.

	SUN			CUB			AWA1			AWA2			aPY		
Method	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
DAP	4.2	25.1	7.2	1.7	67.9	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.8	78.3	9.0
IAP	1.0	37.8	1.8	0.2	72.8	0.4	2.1	78.2	4.1	0.9	87.6	1.8	5.7	65.6	10.4
CONSE	6.8	**39.9**	11.6	1.6	**72.2**	3.1	0.4	88.6	0.8	0.5	**90.6**	1.0	0.0	**91.2**	0.0
CMT	8.1	21.8	11.8	7.2	49.8	12.6	0.9	87.6	1.8	0.5	90.0	1.0	1.4	85.2	2.8
CMT*	8.7	28.0	13.3	4.7	60.1	8.7	8.4	86.9	15.3	8.7	89.0	15.9	**10.9**	74.2	**19.0**
SSE	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	8.1	82.5	14.8	0.2	78.9	0.4
LATEM	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	0.1	73.0	0.2
ALE	**21.8**	33.1	**26.3**	23.7	62.8	**34.4**	**16.8**	76.1	**27.5**	14.0	81.8	23.9	4.6	73.7	8.7
DEVISE	16.9	27.4	20.9	**23.8**	53.0	32.8	13.4	68.7	22.4	**17.1**	74.7	**27.8**	4.9	76.9	9.2
SJE	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	3.7	55.7	6.9
ESZSL	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	2.4	70.1	4.6
SYNC	7.9	43.3	13.4	11.5	70.9	19.8	8.9	87.3	16.2	10.0	90.5	18.0	7.4	66.3	13.3
SAE	8.8	18.0	11.8	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9
GFZSL	0.0	39.6	0.0	0.0	45.7	0.0	1.8	80.3	3.5	2.5	80.1	4.8	0.0	83.3	0.0

Table 4.6: Generalized Zero-Shot Learning on Proposed Split (PS) measuring ts = Top-1 accuracy on $\mathcal{Y}^{ts}$, tr=Top-1 accuracy on $\mathcal{Y}^{tr}$, H = harmonic mean (CMT*: CMT with novelty detection). We measure top-1 accuracy in %.

4.6.2 Generalized Zero-Shot Learning Results

In real world applications, image classification systems do not have access to whether a novel image belongs to a seen or unseen class in advance. Hence, generalized zero-shot learning is interesting from a practical point of view. Here, we use same models trained on ZSL setting on our proposed splits (PS). We evaluate performance on both $\mathcal{Y}^{tr}$ and $\mathcal{Y}^{ts}$ (using held-out images).

As shown in Table 4.6, generalized zero-shot learning results are significantly lower than zero-shot learning results. This is due to the fact that training classes are included in the search space which act as distractors for the images that come from test classes, e.g. most of the images that are being evaluated. An interesting observation is that compatibility learning frameworks, e.g. ALE, DEVISE, SJE, perform well on test classes. However, methods that learn independent attribute or object classifiers, e.g. DAP and CONSE, perform well on training classes. Due to this discrepancy, we evaluate the harmonic mean which takes a weighted average of training and test class accuracy as shown in Equation 4.17. The harmonic mean measure ranks ALE as the best performing method on SUN, CUB and AWA1 datasets whereas on our AWA2 dataset DEVISE performs the best and on aPY dataset CMT* performs the best. Note that CMT* has an integrated novelty detection phase for which the method receives another supervision signal determining if the image belongs to a training or a test class. Similar to the ImageNet results, GFZSL (Verm and Rai, 2017) performs poorly on GZSL setting.

As for the generalized zero-shot learning setting on ImageNet, we report results measured on unseen classes as no images are reserved from seen classes on Figure 4.6. Our first observation is that there is no winner model in all cases, the results diverge for different splits and different accuracy measures. For instance, when the

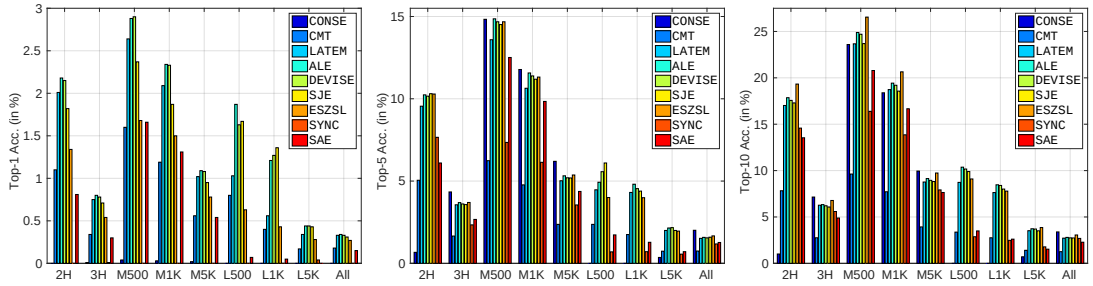

Figure 4.6: GZSL on Imagenet, measuring Top-1, Top-5 and Top-10 accuracy. 2/3H: classes with 2/3 hops away from ImageNet1K $\mathcal{Y}^{tr}$, M500/M1K/M5K: 500/1K/5K most populated classes, L500/L1K/L5K: 500/1K/5K least populated classes, All: Remaining 20K classes.

performance is measured with Top-1 accuracy, in general the best performing model seems to be DEVISE, ALE and SJE which are all linear compatibility learning models. On the other hand, for Top-5 accuracy different models take the lead in different splits, e.g. CONSE works the best for 3H and M5K indicating that it performs better when the number of images that come from unseen classes is larger. Whereas SJE and ESZSL works better for 2H, M500, L5H settings. Finally, for Top-10 accuracy, the best performing model overall is ESZSL which is the model that learns a linear compatibility with an explicit regularization scheme. Finally, for Top-1, Top-5 and Top-10 results we observe the same trend for when all the unseen classes are included in the test set, i.e. the models perform similarly however CONSE slightly stands out for Top-5 and Top-10 accuracy plots.

In summary, generalized zero-shot learning setting provides one more level of detail on the performance of zero-shot learning methods. Our take-home message is that the accuracy of training classes is as important as the accuracy of test classes in real world scenarios. Therefore, methods should be designed in a way that they are able to predict labels well both in train and test classes.

Visualizing Method Ranking. Similar to the analysis in the previous section that was conducted for zero-shot learning setting, we rank the 13 methods, i.e. (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a; Kodirov *et al.*, 2017; Verm and Rai, 2017), based on their results obtained on SUN, CUB, AWA1, AWA2 and aPY. The performance is measured on seen classes, unseen classes and the Harmonic mean of the two.

The rank matrix of test classes, i.e. Figure 4.7 top left, shows that highest ranked methods,i.e. ALE, DEVISE, SJE, although overall the absolute accuracy numbers are lower (Table 4.6). Note that in Figure 4.4 GFZSL ranked highest which shows that GFZSL is not as strong for GZSL task. The rank matrix of the harmonic mean shows the same trend. However, the rank matrix of training classes, i.e. Figure 4.7 top right, shows that models that learn intermediate attribute classifiers perform well for the images that come from training classes. However, these models typically do not lead to a high accuracy for the images that belong to unseen classes as shown

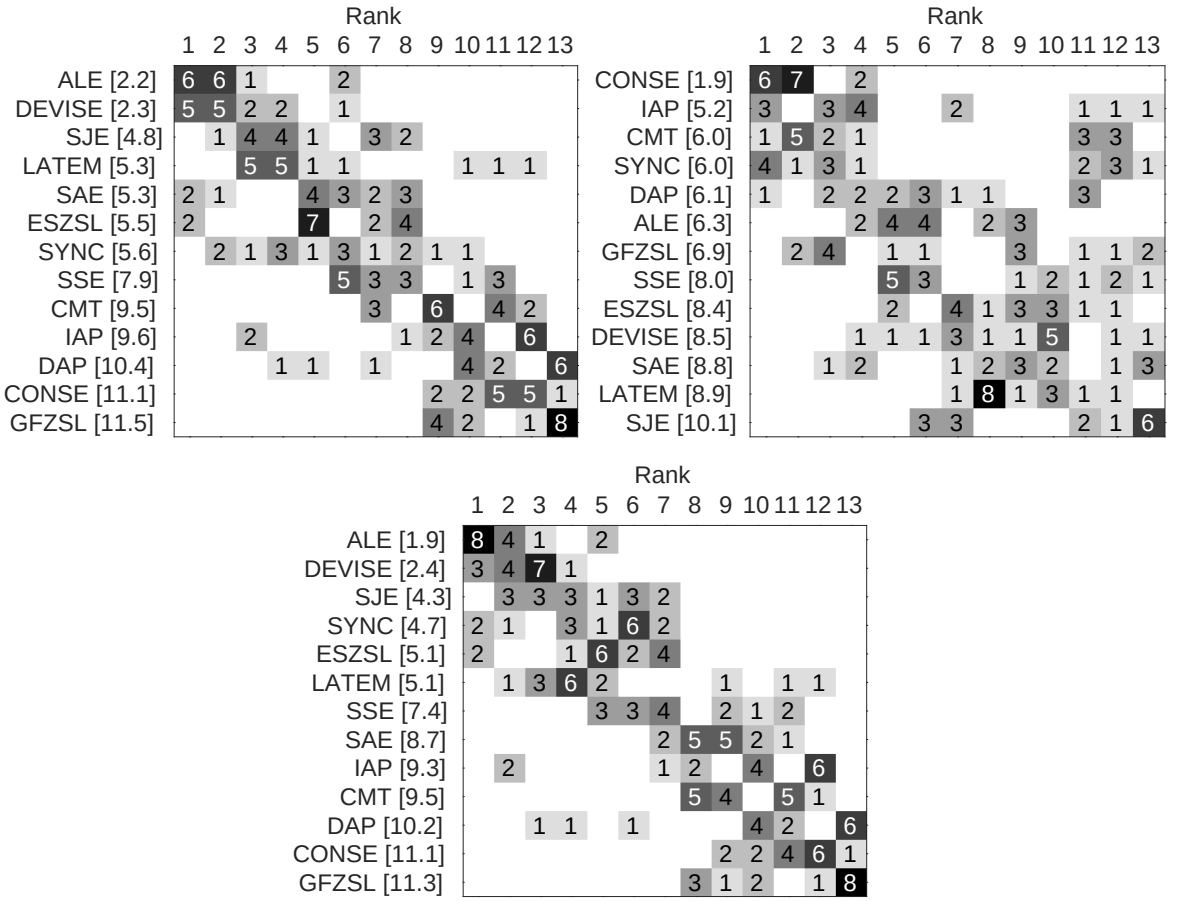

Figure 4.7: Ranking 13 models on the proposed split (PS) in generalized zero-shot learning setting. Top-Left: Top-1 accuracy (T1) is measured on unseen classes (ts), Top-Right: T1 is measured on seen classes (tr), Bottom: T1 is measured on Harmonic mean (H).

in Table 4.6. This eventually makes the harmonic mean, i.e. the overall accuracy on both training and test classes, lower. These results clearly suggest that one should not only optimize for test class accuracy but also for training class accuracy while evaluating generalized zero-shot learning.

Our final observation from Figure 4.7 is that CMT* is better than CMT in all cases which supports the argument that a simple novelty detection scheme helps to improve results. However, it is important to note that the proposed novelty detection mechanism uses more supervision than classic zero-shot learning models. Although the label of test classes is not used, whether the sample comes from a seen or unseen class is an additional supervision.

4.6.3 Transductive (Generalized) Zero-Shot Learning

In contrast to previous zero-shot learning approaches that learn only with data from training classes, transductive approaches use unlabaled images from test classes.

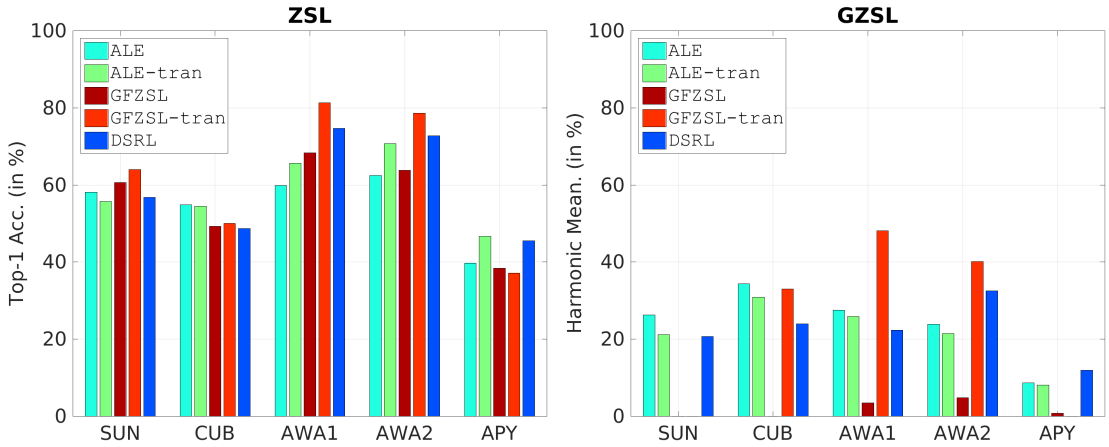

Figure 4.8: Zero-shot (left) and generalized zero-shot learning (right) results in the transductive learning setting on our Proposesd Split.

In this section, we evaluate three state-of-the-art transductive ZSL approaches, i.e. DSRL (Ye and Guo, 2017), GFZSL-tran (Verm and Rai, 2017), and ALE-tran (Akata *et al.*, 2015a). Similar to the previous section, we evaluate those approaches on our proposed splits in both zero-shot learning where test time search space is composed of only unseen classes and generalized zero-shot learning where it contains both seen and unseen classes. The performance is per-class averaged top-1 accuracy.

Our transductive learning results are presented in Figure 4.8. We observe that in ZSL setting, transductive learning leads to accuracy improvement, e.g. ALE-tran and GFZSL-tran outperforms ALE and GFZSL respectively in almost all cases. In particular, on AWA2, GFZSL-tran achieves 78.6%, significantly improving GFZSL (63.8%). On APY, ALE-tran obtains 45.5% and significantly improves ALE (37.1%) as well. Moreover, GFZSL-tran outperforms ALE-tran and DSRL on SUN, AWA1 and AWA2. However, ALE-tran performs the best on CUB and APY. In GZSL setting we observe a different trend, i.e. transductive learning does not improve results for ALE in any of the datasets. Although, on AWA1 and AWA2 GFZSL results improve significantly for the transductive learning setting, on other datasets GFZSL model performs poorly both in inductive and in transductive settings.

4.7 CONCLUSION

In this work, we evaluated a significant number of state-of-the-art zero-shot learning methods, i.e. (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a; Kodirov *et al.*, 2017; Verm and Rai, 2017; Ye and Guo, 2017), on several datasets, i.e. SUN, CUB, AWA1, AWA2, aPY and ImageNet, within a unified evaluation protocol both in zero-shot and generalized zero-shot settings.

Our evaluation showed that generative models and compatibility learning frame-

works have an edge over learning independent object or attribute classifiers and also over other hybrid models for the classic zero-shot learning setting. We observed that unlabeled data of unseen classes can further improve the zero-shot learning results, thus it is not fair to compare transductive learning approaches with inductive ones. We discovered that some standard zero-shot dataset splits may treat feature learning disjoint from the training stage as several test classes are included in the ImageNet1K dataset that is used to train the deep neural networks that act as feature extractor. Therefore, we proposed new dataset splits making sure that none of the test classes in none of the datasets belong to ImageNet1K. Moreover, disjoint training and validation class split is a necessary component of parameter tuning in zero-shot learning setting.

In addition, we introduced a new Animal with Attributes (AWA2) dataset. AWA2 inherits the same 50 classes and attributes annotations from the original Animal with Attributes (AWA1) dataset, but consists of different 37,322 images with publicly available redistribution license. Our experimental results showed that the 12 methods that we evaluated perform similarly on AWA2 and AWA1. Moreover, our statistical consistency test indicated that AWA1 and AWA2 are compatible with each other.

Finally, including training classes in the search space while evaluating the methods, i.e. generalized zero-shot learning, provides an interesting playground for future research. Although the generalized zero-shot learning accuracy obtained with 13 models compared to their zero-shot learning accuracy is significantly lower, the relative performance comparison of different models remain the same. Having noticed that some models perform well when the test set is composed only of seen classes, while some others perform well when the test set is composed of only of unseen classes, we proposed the Harmonic mean of seen and unseen class accuracy as a unified measure for performance in GZSL setting. The Harmonic mean encourages the models to perform well on both seen and unseen class samples, which is closer to a real world setting. In summary, our work extensively evaluated the good and bad aspects of zero-shot learning while sanitizing the ugly ones.

5 FEATURE GENERATING NETWORKS FOR ZERO-SHOT IMAGE CLASSIFICATION

Contents

In Chapter 4, we observe that almost all zero-shot learning approaches fail to predict novel classes in the realistic generalized zero-shot learning setting. In this chapter, our goal is to develop methods to tackle generalized zero-shot learning under the benchmark proposed in Chapter 4. In a high-level point of view, we propose to learn a feature generator that synthesizes visual features for novel classes. The generated features alleviate the imbalanced issues and consistently improve the zero-shot and generalized zero-shot learning results.

In Chapter 6, we extend the approach introduced this chapter by improving the generative model and incorporating unlabeled data. We also show the effectiveness of our approach on few-shot learning tasks. Chapter 7 defines and addresses the zero-shot and few-shot learning problems in the scenario of semantic segmentation. Chapter 8 tackles few-shot learning challenges arised in video action classification tasks.

5.1 INTRODUCTION

Deep learning has allowed to push performance considerably across a wide range of computer vision and machine learning tasks. However, almost always, deep learning requires large amounts of training data which we are lacking in many practical scenarios, e.g. it is impractical to annotate all the concepts that surround us, and have enough of those annotated samples to train a deep network. Therefore, training data generation has become a hot research topic (e.g. Chawla *et al.*, 2002;

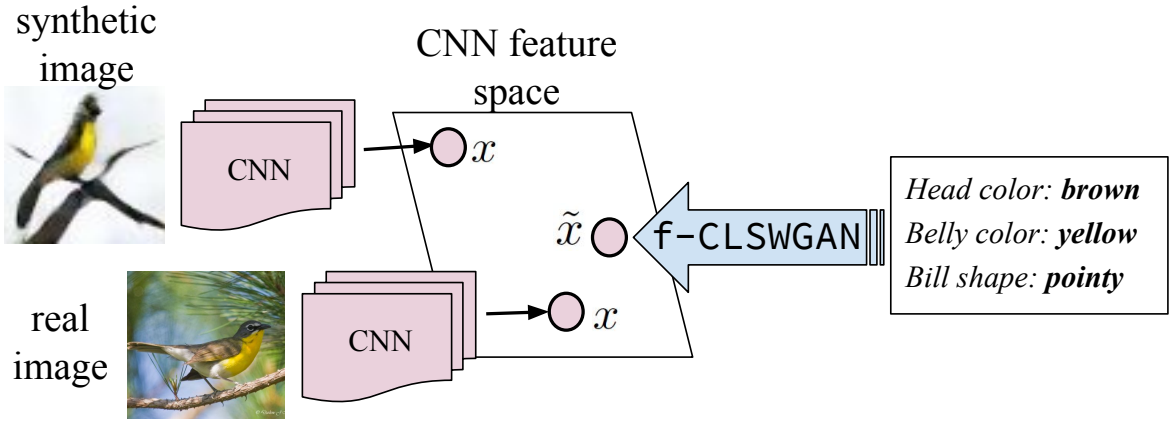

Figure 5.1: CNN features can be extracted from: 1) real images, however in zero-shot learning we do not have access to any real images of unseen classes, 2) synthetic images, however they are not accurate enough to improve image classification performance. We tackle both of these problems and propose a novel attribute conditional feature generating adversarial network formulation, i.e. f-CLSWGAN, to generate CNN features of unseen classes.

Goodfellow *et al.*, 2014; Chen and Koltun, 2017; Reed *et al.*, 2016c; Zhang *et al.*, 2017a; Salimans *et al.*, 2016). Generative Adversarial Networks (Goodfellow *et al.*, 2014) are particularly appealing as they allow generating realistic and sharp images conditioned, for instance, on object categories (e.g. Reed *et al.*, 2016c; Zhang *et al.*, 2017a). However, they do not yet generate images of sufficient quality to train deep learning architectures as demonstrated by our experimental results.

In this work, we are focusing on arguably the most extreme case of lacking data, namely zero-shot learning (e.g. Lampert *et al.*, 2013; Xian *et al.*, 2017; Chao *et al.*, 2016), where the task is to learn to classify when *no* labeled examples of certain classes are available during training. We argue that this scenario is a great testbed for evaluating the robustness and generalization of generative models. In particular, if the generator learns discriminative visual data with enough variation, the generated data should be useful for supervised learning. Hence, one contribution of this chapter is a comparison of various existing GAN-models and another competing generative model, i.e. GMMN, for visual feature generation. In particular, we look into both zero-shot learning (ZSL) where the test time search space is restricted to unseen class labels and generalized zero-shot learning (GZSL) for being a more realistic scenario as at test time the classifier has to decide between both seen and unseen class labels. In this context, we propose a novel GAN-method – namely f-CLSWGAN that generates features instead of images and is trained with a novel loss improving over alternative GAN-models.

We summarize our contributions as follows. (1) We propose a novel conditional generative model f-CLSWGAN that synthesizes CNN features of unseen classes by optimizing the Wasserstein distance regularized by a classification loss. (2) Across

five datasets with varying granularity and sizes, we consistently improve upon the state of the art in both the ZSL and GZSL settings. We demonstrate a practical application for adversarial training and propose GZSL as a proxy task to evaluate the performance of generative models. (3) Our model is generalizable to different deep CNN features, e.g. extracted from GoogleNet or ResNet, and may use different class-level auxiliary information, e.g. sentence, attribute, and word2vec embeddings.

5.2 RELATED WORK

In this section we review some recent relevant literature on Generative Adversarial Networks, Zero-Shot Learning (ZSL) and Generalized Zero-Shot (GZSL) Learning.

Generative Adversarial Network. GAN (Goodfellow *et al.*, 2014) was originally proposed as a means of learning a generative model which captures an arbitrary data distribution, such as images, from a particular domain. The input to a generator network is a "noise" vector z drawn from a latent distribution, such as a multivariate Gaussian. DCGAN (Radford *et al.*, 2016) extends GAN by leveraging deep convolution neural networks and providing best practices for GAN training. (Wang and Gupta, 2016) improves DCGAN by factorizing the image generation process into style and structure networks, InfoGAN (Chen *et al.*, 2016) extends GAN by additionally maximizing the mutual information between interpretable latent variables and the generator distribution. GAN has also been extended to a conditional GAN by feeding the class label (Mirza and Osindero, 2014), sentence descriptions (Reed *et al.*, 2016b,c; Zhang *et al.*, 2017a), into both the generator and discriminator. The theory of GAN is recently investigated in (Arjovsky and Bottou, 2017; Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017), where they show that the Jenson-shannon divergence optimized by the original GAN leads to instability issues. To cure the unstable training issues of GANs, (Arjovsky *et al.*, 2017) proposes Wasserstein-GAN (WGAN), which optimizes an efficient approximation of the Earth Mover, i.e. Wasserstein-1, distance. While WGAN attains better theoretical properties than the original GAN, it still suffers from vanishing and exploding gradient problems due to weight clipping to enforce the 1-Lipschitz constraint on the discriminator. Hence, (Gulrajani *et al.*, 2017) proposes an improved version of WGAN enforcing the Lipschitz constraint through gradient penalty. Although those papers have demonstrated realistic looking images, they have not applied this idea to image feature generation.

In this chapter, we empirically show that images generated by the state-of-the-art GAN (Gulrajani *et al.*, 2017) are not ready to be used as training data for learning a classifier. Hence, we propose a novel GAN architecture to directly generate CNN features that can be used to train a discriminative classifier for zero-shot learning. Combining the powerful WGAN (Gulrajani *et al.*, 2017) loss and a classification loss which enforces the generated features to be discriminative, our proposed GAN architecture improves the original GAN (Goodfellow *et al.*, 2014) by a large margin and has an edge over WGAN (Gulrajani *et al.*, 2017) thanks to our regularizer.

For zero-shot and generalized zero-shot learning literature, readers can refer to

Chapter 2.

In this chapter, we propose to tackle generalized zero-shot learning by generating CNN features for unseen classes via a novel GAN model. Our work is different from (Hariharan and Girshick, 2017) because they generate additional examples for data-starved classes from feature vectors alone, which is unimodal and do not generalize to unseen classes. Our work is closer to (Bucher *et al.*, 2017) in which they generate features via GMMN (Li *et al.*, 2015). Hence, we directly compare with them on the latest zero-shot learning benchmark (Xian *et al.*, 2017) and show that WGAN (Arjovsky *et al.*, 2017) coupled with our proposed classification loss can further improve GMMN in feature generation on most datasets for both ZSL and GZSL tasks.

5.3 FEATURE GENERATION & CLASSIFICATION IN ZSL

Existing ZSL models only see labeled data from seen classes during training biasing the predictions to seen classes. The main insight of our proposed model is that by feeding additional synthetic CNN features of unseen classes, the learned classifier will also explore the embedding space of unseen classes. Hence, the key to our approach is the ability to generate semantically rich CNN feature distributions conditioned on a class specific semantic vector e.g. attributes, without access to any images of that class. This alleviates the imbalance between seen and unseen classes, as there is no limit to the number of synthetic CNN features that our model can generate. It also allows to directly train a discriminative classifier, i.e. Softmax classifier, even for unseen classes.

We begin by defining the problem of our interest. Let $\mathcal{S} = \{(x, y, c(y))|x \in \mathcal{X}, y \in \mathcal{Y}^s, c(y) \in \mathcal{C}\}$ where $\mathcal{S}$ stands for the training data of seen classes, $x \in \mathbb{R}^{d_x}$ is the CNN features, y denotes the class label in $\mathcal{Y}^s = \{y_1, \ldots, y_K\}$ consisting of K discrete seen classes, and $c(y) \in \mathbb{R}^{d_c}$ is the class embedding, e.g. attributes, of class y that models the semantic relationship between classes. In addition, we have a disjoint class label set $\mathcal{Y}^u = \{u_1, \ldots, u_L\}$ of unseen classes, whose class embedding set $\mathcal{U} = \{(u, c(u))|u \in \mathcal{Y}^u, c(u) \in \mathcal{C}\}$ is available but images and image features are missing. Given $\mathcal{S}$ and $\mathcal{U}$, the task of ZSL is to learn a classifier $f_{zsl} : \mathcal{X} \rightarrow \mathcal{Y}^u$ and in GZSL we learn a classifier $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$.

5.3.1 Feature Generation

In this section, we begin our discussion with Generative Adversarial Networks (GAN) (Goodfellow *et al.*, 2014) for it being the basis of our model. GAN consists of a generative network G and a discriminative network D that compete in a two player minimax game. In the context of generating image pixels, D tries to accurately distinguish real images from generated images, while G tries to fool the discriminator by generating images that are mistakable for real. Following (Mirza and Osindero, 2014), we extend GAN to conditional GAN by including a conditional variable to

both G and D. In the following we give the details of the conditional GAN variants that we develop. Our novelty lies in that we develop three conditional GAN variants, i.e. f-GAN, f-WGAN and f-CLSWGAN, to generate image features rather than image pixels. It is worth noting that our models are only trained with seen class data $\mathcal{S}$ but can also generate image features of unseen classes.

f-GAN. Given the train data $\mathcal{S}$ of seen classes, we aim to learn a conditional generator $G : \mathcal{Z} \times \mathcal{C} \rightarrow \mathcal{X}$, which takes random Gaussian noise $z \in \mathcal{Z} \subset \mathbb{R}^{d_z}$ and class embedding $c(y) \in \mathcal{C}$ as its inputs, and outputs a CNN image feature $\tilde{x} \in \mathcal{X}$ of class y. Once the generator G learns to generate CNN features of real images, i.e. x, conditioned on the seen class embedding $c(y) \in \mathcal{Y}^s$, it can also generate $\tilde{x}$ of any unseen class u via its class embedding $c(u)$. Our feature generator f-GAN is learned by optimizing the following objective,

$$\begin{aligned} \min_G \max_D \mathcal{L}_{GAN} = & E[\log D(x, c(y))] + \\ & E[\log (1 - D(\tilde{x}, c(y)))], \end{aligned} \tag{5.1}$$

with $\tilde{x} = G(z, c(y))$. The discriminator $D : \mathcal{X} \times \mathcal{C} \rightarrow [0, 1]$ is a multi-layer perceptron with a sigmoid function as the last layer. While D tries to maximize the loss, G tries to minimizes it. Although GAN has been shown to capture complex data distributions, e.g. pixel images, they are notoriously difficult to train (Arjovsky and Bottou, 2017).

f-WGAN. We extend the improved WGAN (Gulrajani *et al.*, 2017) to a conditional WGAN by integrating the class embedding $c(y)$ to both the generator and the discriminator. The loss is,

$$\begin{aligned} \mathcal{L}_{WGAN} = & E[D(x, c(y))] - E[D(\tilde{x}, c(y))] - \\ & \lambda E[(||\nabla_{\hat{x}} D(\hat{x}, c(y))||_2 - 1)^2], \end{aligned} \tag{5.2}$$

where $\tilde{x} = G(z, c(y))$, $\hat{x} = \alpha x + (1 - \alpha)\tilde{x}$ with $\alpha \sim U(0, 1)$, and λ is the penalty coefficient. In contrast to the GAN, the discriminative network here is defined as $D : \mathcal{X} \times \mathcal{C} \rightarrow \mathbb{R}$, which eliminates the sigmoid layer and outputs a real value. The log in Equation 5.1 is also removed since we are not optimizing the log likelihood. Instead, the first two terms in Equation 6.1 approximate the Wasserstein distance, and the third term is the gradient penalty which enforces the gradient of D to have unit norm along the straight line between pairs of real and generated points. Again, we solve a minmax optimization problem,

$$\min_G \max_D \mathcal{L}_{WGAN} \tag{5.3}$$

f-CLSWGAN. f-WGAN does not guarantee that the generated CNN features are well suited for training a discriminative classifier, which is our goal. We conjecture that this issue could be alleviated by encouraging the generator to construct features that can be correctly classified by a discriminative classifier trained on the input

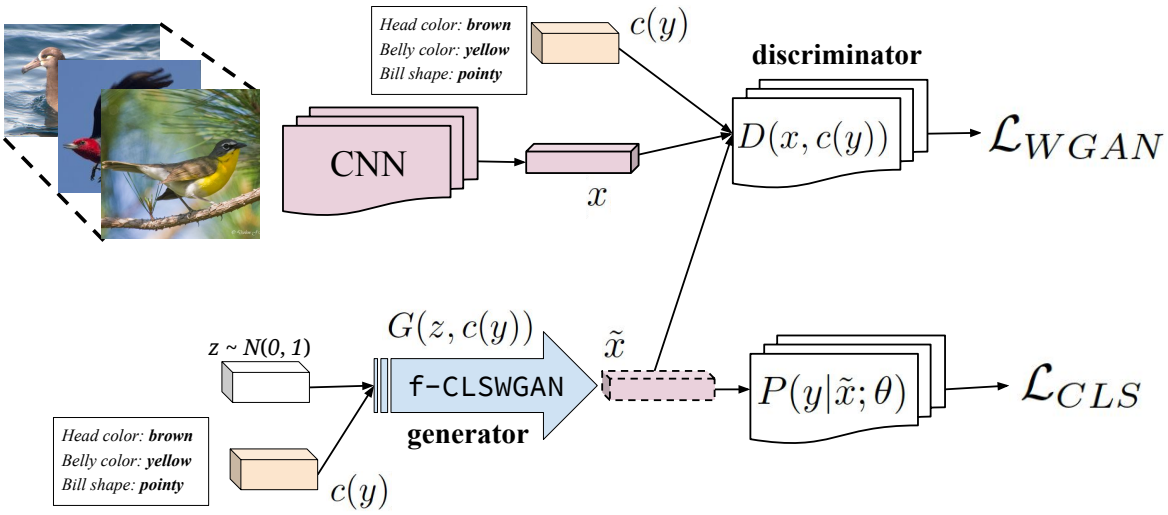

Figure 5.2: Our f-CLSWGAN: we propose to minimize the classification loss over the generated features and the Wasserstein distance with gradient penalty.

data. To this end, we propose to minimize the classification loss over the generated features in our novel f-CLSWGAN formulation. We use the negative log likelihood,

$$\mathcal{L}_{CLS} = -E_{\tilde{x}\sim p_{\tilde{x}}}[\log P(y|\tilde{x};\theta)], \tag{5.4}$$

where $\tilde{x} = G(z, c(y))$, y is the class label of $\tilde{x}$, $P(y|\tilde{x};\theta)$ denotes the probability of $\tilde{x}$ being predicted with its true class label y. The conditional probability is computed by a linear softmax classifier parameterized by θ, which is pretrained on the real features of seen classes. The classification loss can be thought of as a regularizer enforcing the generator to construct discriminative features. Our full objective then becomes,

$$\min_G \max_D \mathcal{L}_{WGAN} + \beta\mathcal{L}_{CLS} \tag{5.5}$$

where β is a hyperparameter weighting the classifier.

5.3.2 Classification

Given $c(u)$ of any unseen class $u \in \mathcal{Y}^u$, by resampling the noise z and then recomputing $\tilde{x} = G(z, c(u))$, arbitrarily many visual CNN features $\tilde{x}$ can be synthesized. After repeating this feature generation process for every unseen class, we obtain a synthetic training set $\tilde{\mathcal{U}} = \{(\tilde{x}, u, c(u))\}$. We then learn a classifier by training either a multimodal embedding model or a softmax classifier. Our generated features allow to train those methods on the combinations of real seen class data $\mathcal{S}$ and generated unseen class data $\tilde{\mathcal{U}}$.

Multimodal Embedding. Many efficient zero-shot learning approaches, e.g. (Akata *et al.*, 2015a), DEVISE (Frome *et al.*, 2013), SJE (Akata *et al.*, 2015c), ESZSL (**?**) and

LATEM (Xian *et al.*, 2016), learn a multimodal embedding between the image feature space $\mathcal{X}$ and the class embedding space $\mathcal{C}$ using seen classes data $\mathcal{S}$. With our generated features, those methods can be trained with seen classes data $\mathcal{S}$ together with unseen classes data $\tilde{\mathcal{U}}$ to learn a more robust classifier. The embedding model $F(x, c(y); W)$, parameterized by W, measures the compatibility score between any image feature x and class embedding $c(y)$ pair. Given a query image feature x, the classifier searches for the class embedding with the highest compatibility via:

$$f(x) = \underset{y}{\operatorname{argmax}} F(x, c(y); W), \tag{5.6}$$

where in ZSL, $y \in \mathcal{Y}^u$ and in GZSL, $y \in \mathcal{Y}^s \cup \mathcal{Y}^u$.

Softmax. The standard softmax classifier minimizes the negative log likelihood loss,

$$\min_{\theta} -\frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \log P(y|x;\theta), \tag{5.7}$$

where $\theta \in \mathbb{R}^{d_x \times N}$ is the weight matrix of a fully connected layer which maps the image feature x to N unnormalized probabilities with N being the number of classes, and $P(y|x;\theta) = \frac{\exp(\theta_y^T x)}{\sum_i^N \exp(\theta_i^T x)}$. Depending on the task, $\mathcal{T} = \tilde{\mathcal{U}}$ if it is ZSL and $\mathcal{T} = \mathcal{S} \cup \tilde{\mathcal{U}}$ if it is GZSL. The prediction function is:

$$f(x) = \arg\max_{y} P(y|x;\theta), \tag{5.8}$$

where in ZSL, $y \in \mathcal{Y}^u$ and in GZSL, $y \in \mathcal{Y}^s \cup \mathcal{Y}^u$.

5.4 EXPERIMENTS

First we detail our experimental protocol, then we present (1) our results comparing our framework with the state of the art for GZSL and ZSL tasks on four challenging datasets, (2) our analysis of f-xGAN [4] under different conditions, (3) our large-scale experiments on ImageNet and (4) our comparison of image and image feature generation.

Datasets. Caltech-UCSD-Birds 200-2011 (CUB) (Welinder *et al.*, 2010), Oxford Flowers (FLO) (Nilsback and Zisserman, 2008) and SUN Attribute (SUN) (Patterson and Hays, 2012) are all fine-grained datasets. CUB contains 11,788 images from 200 different types of birds annotated with 312 attributes. FLO dataset 8189 images from 102 different types of flowers without attribute annotations. However, for both CUB and FLO we use the fine-grained visual descriptions collected by (Reed *et al.*, 2016a). SUN contains 14,340 images from 717 scenes annotated with 102 attributes. Finally, Animals with Attributes (AWA) (Lampert *et al.*, 2013) is a coarse-grained

[4] We denote our f-GAN, f-WGAN, f-CLSWGAN as f-xGAN

Dataset	att	stc	$\lvert\mathcal{Y}^s\rvert + \lvert\mathcal{Y}^u\rvert$	$\lvert\mathcal{Y}^s\rvert$	$\lvert\mathcal{Y}^u\rvert$
CUB (Welinder *et al.*, 2010)	312	Y	200	100 + 50	50
FLO (Nilsback and Zisserman, 2008)	–	Y	102	62 + 20	20
SUN (Patterson and Hays, 2012)	102	N	717	580 + 65	72
AWA (Lampert *et al.*, 2013)	85	N	50	27 + 13	10

Table 5.1: CUB, SUN, FLO, AWA datasets, in terms of number of attributes per class (att), sentences (stc), number of classes in training + validation ($\mathcal{Y}^s$) and test classes ($\mathcal{Y}^u$).

dataset with 30,475 images, 50 classes and 85 attributes. Statistics of the datasets are presented in Table 5.1. We use the zero-shot splits proposed by (Xian *et al.*, 2017) for AWA, CUB and SUN insuring that none of the training classes are present in ImageNet (Deng *et al.*, 2009)[5]. For FLO, we use the standard split provided by (Reed *et al.*, 2016a).

Features. As real CNN features, we extract 2048-dim top-layer pooling units of the 101-layered ResNet (He *et al.*, 2016) from the entire image. We do not do any image pre-processing such as cropping, background subtraction etc, or use any other data augmentation techniques. ResNet is pre-trained on ImageNet 1K and not fine-tuned. As synthetic CNN features, we generate 2048-dim CNN features using our f-xGAN model. As the class embedding, unless it is stated otherwise, we use per-class attributes for AWA (85-dim), CUB (312-dim) and SUN (102-dim). Furthermore, for CUB and Flowers, we extract 1024-dim character-based CNN-RNN (Reed *et al.*, 2016a) features from fine-grained visual descriptions (10 sentences per image). None of the $\mathcal{Y}^u$ sentences are seen during training the CNN-RNN. We build per-class sentences by averaging the CNN-RNN features that belong to the same class.

Evaluation Protocol. At test time, in the ZSL setting, the aim is to assign an unseen class label, i.e. $\mathcal{Y}^u$ to the test image and in GZSL setting, the search space includes both seen or unseen classes, i.e. $\mathcal{Y}^s \cup \mathcal{Y}^u$. We use the unified evaluation protocol proposed in (Xian *et al.*, 2017). In the ZSL setting, the average accuracy is computed independently for each class before dividing their cumulative sum by the number of classes; i.e., we measure average per-class top-1 accuracy (T1). In the GZSL setting, we compute average per-class top-1 accuracy on seen classes ($\mathcal{Y}^s$) denoted as $\mathbf{s}$, average per-class top-1 accuracy on unseen classes ($\mathcal{Y}^u$) denoted as $\mathbf{u}$ and their harmonic mean, i.e. $H = 2 * (\mathbf{s} * \mathbf{u})/(\mathbf{s} + \mathbf{u})$.

Implementation details. In all f-xGAN models, both the generator and the discriminator are MLP with LeakyReLU activation. The generator consists of a single hidden layer with 4096 hidden units. Its output layer is ReLU because we aim to learn the top max-pooling units of ResNet-101. While the discriminator of f-GAN has

[5]as ImageNet is used for pre-training the ResNet (He *et al.*, 2016)

		Zero-Shot Learning				Generalized Zero-Shot Learning											
		CUB	FLO	SUN	AWA	CUB			FLO			SUN			AWA		
Classifier	FG	T1	T1	T1	T1	u	s	H	u	s	H	u	s	H	u	s	H
DEVISE	none	52.0	45.9	56.5	54.2	23.8	53.0	32.8	9.9	44.2	16.2	16.9	27.4	20.9	13.4	68.7	22.4
	f-CLSWGAN	60.3	60.4	60.9	66.9	52.2	42.4	46.7	45.0	38.6	41.6	38.4	25.4	30.6	35.0	62.8	45.0
SJE	none	53.9	53.4	53.7	65.6	23.5	59.2	33.6	13.9	47.6	21.5	14.7	30.5	19.8	11.3	74.6	19.6
	f-CLSWGAN	58.4	67.4	56.5	66.9	48.1	37.4	42.1	52.1	56.2	54.1	36.7	25.0	29.7	37.9	70.1	49.2
LATEM	none	49.3	40.4	55.3	55.1	15.2	57.3	24.0	6.6	47.6	11.5	14.7	28.8	19.5	7.3	71.7	13.3
	f-CLSWGAN	60.8	60.8	61.3	**69.9**	53.6	39.2	45.3	47.2	37.7	41.9	42.4	23.1	29.9	33.0	61.5	43.0
ESZSL	none	53.9	51.0	54.5	58.2	12.6	63.8	21.0	11.4	56.8	19.0	11.0	27.9	15.8	6.6	75.6	12.1
	f-CLSWGAN	54.7	54.3	54.0	63.9	36.8	50.9	43.2	25.3	69.2	37.1	27.8	20.4	23.5	31.1	72.8	43.6
ALE	none	54.9	48.5	58.1	59.9	23.7	62.8	34.4	13.3	61.6	21.9	21.8	33.1	26.3	16.8	76.1	27.5
	f-CLSWGAN	**61.5**	**71.2**	**62.1**	68.2	40.2	59.3	47.9	54.3	60.3	57.1	41.3	31.1	35.5	47.6	57.2	52.0
Softmax	none	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	f-CLSWGAN	57.3	67.2	60.8	68.2	43.7	57.7	**49.7**	59.0	73.8	**65.6**	42.6	36.6	**39.4**	57.9	61.4	**59.6**

Table 5.2: ZSL measuring per-class average Top-1 accuracy (T1) on $\mathcal{Y}^u$ and GZSL measuring **u** = T1 on $\mathcal{Y}^u$, **s** = T1 on $\mathcal{Y}^s$, H = harmonic mean (FG=feature generator, none: no access to generated CNN features, hence softmax is not applicable). f-CLSWGAN significantly boosts both the ZSL and GZSL accuracy of all classification models on all four datasets.

one hidden layer with 1024 hidden units in order to stabilize the GAN training, the discriminators of f-WGAN and f-CLSWGAN have one hidden layer with 4096 hidden units as WGAN (Gulrajani *et al.*, 2017) does not have instability issues thus a stronger discriminator can be applied here. We do not apply batch normalization our empirical evaluation showed a significant degradation of the accuracy when batch normalization is used. The noise z is drawn from a unit Gaussian with the same dimensionality as the class embedding. We use $\lambda = 10$ as suggested in (Gulrajani *et al.*, 2017) and $\beta = 0.01$ across all the datasets.

5.4.1 Comparing with State-of-the-Art

In a first set of experiments, we evaluate our f-xGAN features in both the ZSL and GZSL settings on four challenging datasets: CUB, FLO, SUN and AWA. Unless it is stated otherwise, we use att for CUB, SUN, AWA and stc for FLO (as att are not available). We compare the effect of our feature generating f-xGAN to 6 recent state-of-the-art methods (Xian *et al.*, 2017).

ZSL with f-CLSWGAN. We first provide ZSL results with our f-CLSWGAN in Table 5.2 (left). Here, the test-time search space is restricted to unseen classes $\mathcal{Y}^u$. First, our f-CLSWGAN in all cases improves the state of the art that is obtained without feature generation. The overall accuracy improvement on CUB is from 54.9% to 61.5%, on FLO from 53.4% to 71.2%, on SUN from 58.1% to 62.1% and on AWA from 65.6% to 69.9%, i.e. all quite significant. Another observation is that feature generation is applicable to all the multimodal embedding models and softmax. These

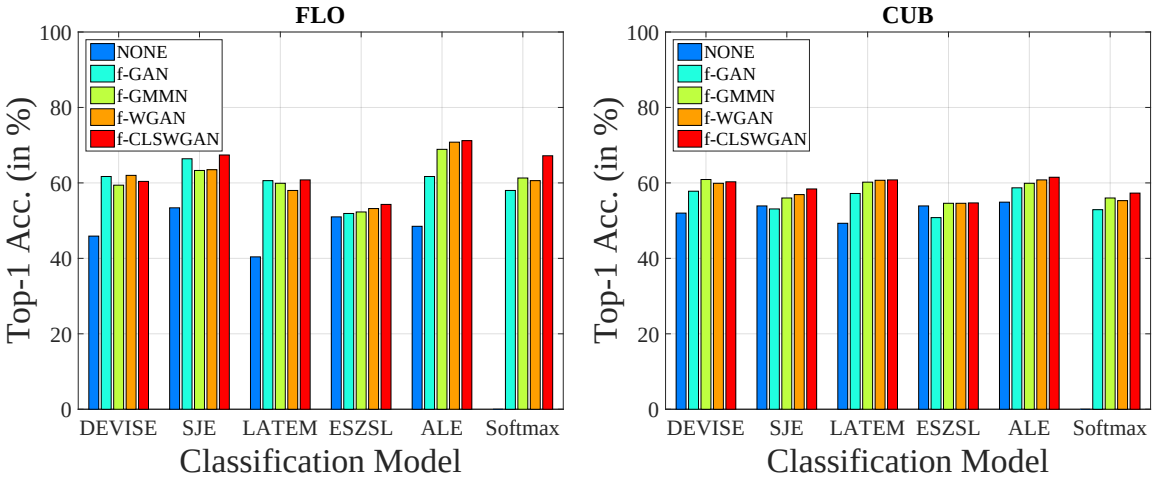

Figure 5.3: Zero-shot learning results when comparing f-xGAN versions with f-GMMN as well as comparing multimodal embedding methods with softmax.

results demonstrate that indeed our f-CLSWGAN generates generalizable and strong visual features of previously unseen classes.

GZSL with f-CLSWGAN. Our main interest is GZSL where the test time search space contains both seen and unseen classes, $\mathcal{Y}^s \cup \mathcal{Y}^u$, and at test time the images come both from seen and unseen classes. Therefore, we evaluate both seen and unseen class accuracy, i.e. **s** and **u**, as well as their harmonic mean (H). The GZSL results with f-CLSWGAN in Table 5.2 (right) demonstrate that for all datasets our f-xGAN significantly improves the H-measure over the state-of-the-art. On CUB, f-CLSWGAN obtains 49.7% in H measure, significantly improving the state of the art (34.4%), on FLO it achieves 65.6% (vs. 21.9%), on SUN it reaches 39.4% (vs. 26.3%), and on AWA it achieves 59.6% (vs. 27.5%). The accuracy boost can be attributed to the strength of the f-CLSWGAN generator learning to imitate CNN features of unseen classes although not having seen any real CNN features of these classes before.

We also observe that without feature generation on all models the seen class accuracy is significantly higher than unseen class accuracy, which indicates that many samples are incorrectly assigned to one of the seen classes. Feature generation through f-CLSWGAN finds a balance between seen and unseen class accuracies by improving the unseen class accuracy while maintaining the accuracy on seen classes. Furthermore, we would like to emphasize that the simple softmax classifier beats all the models and is now applicable to GZSL thanks to our CNN feature generation. This shows the true potential and generalizability of feature generation to various tasks.

ZSL and GZSL with f-xGAN. The generative model is an important component of our framework. Here, we evaluate all versions of our f-xGAN and f-GMMN for it being a strong alternative. We show ZSL and GZSL results of all classification models in Figure 5.3 and Figure 5.4 respectively. We selected CUB and FLO for them

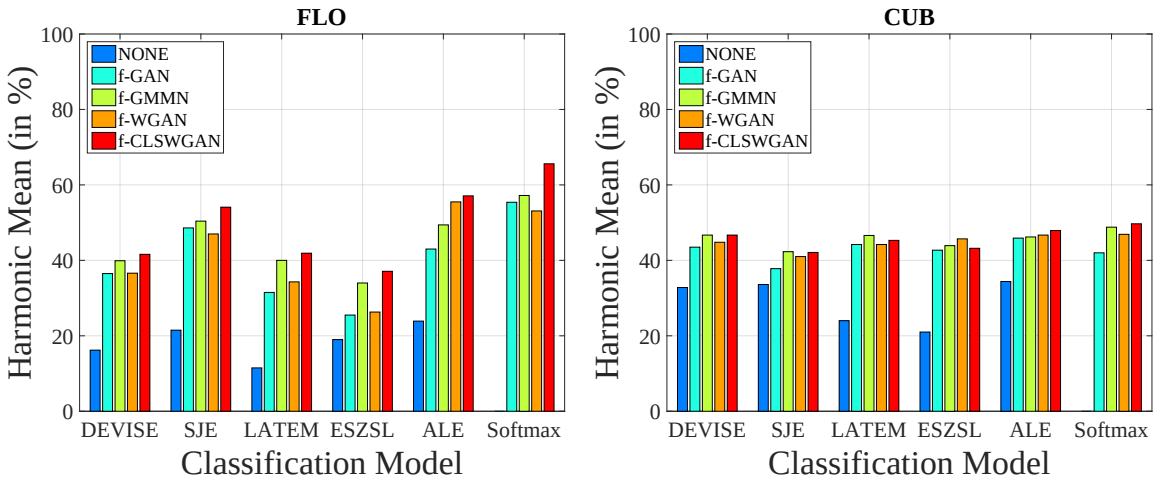

Figure 5.4: Generalized zero-shot learning results when comparing f-xGAN versions with f-GMMN as well as comparing multimodal embedding methods with softmax.

being fine-grained datasets, however we provide full numerical results and plots in the supplementary which shows that our observations hold across datasets. Our first observation is that for both ZSL and GZSL settings all generative models improve in all cases over "none" with no access to the synthetic CNN features. This applies to the GZSL setting and the difference between "none" and f-xGAN is strikingly significant. Our second observation is that our novel f-CLSWGAN model is the best performing generative model in almost all cases for both datasets. Our final observation is that although f-WGAN rarely performs lower than f-GMMN, e.g. ESZL on FLO, our f-CLSWGAN which uses a classification loss in the generator recovers from it and achieves the best result among all these generative models. We conclude from these experiments that generating CNN features to support the classifier when there is missing data is a technique that is flexible and strong.

5.4.2 Analyzing f-xGAN Under Different Conditions

In this section, we analyze f-xGAN in terms of stability, generalization, CNN architecture used to extract real CNN features and the effect of class embeddings on two fine-grained datasets, namely CUB and FLO.

Stability and Generalization. We first analyze how well different generative models fit the seen class data used for training. Instead of using Parzen window-based log-likelihood (Goodfellow *et al.*, 2014) that is unstable, we train a softmax classifier with generated features of seen classes and report the classification accuracy on a held-out test set. Figure 5.5 shows the classification accuracy w.r.t the number of training epochs. On both datasets, we observe a stable training trend. On FLO, compared to the supervised classification accuracy obtained with real images, i.e. the upper bound marked with dashed line, f-GAN remains quite weak even after convergence, which indicates that f-GAN has underfitting issues. A strong alternative is f-GMMN

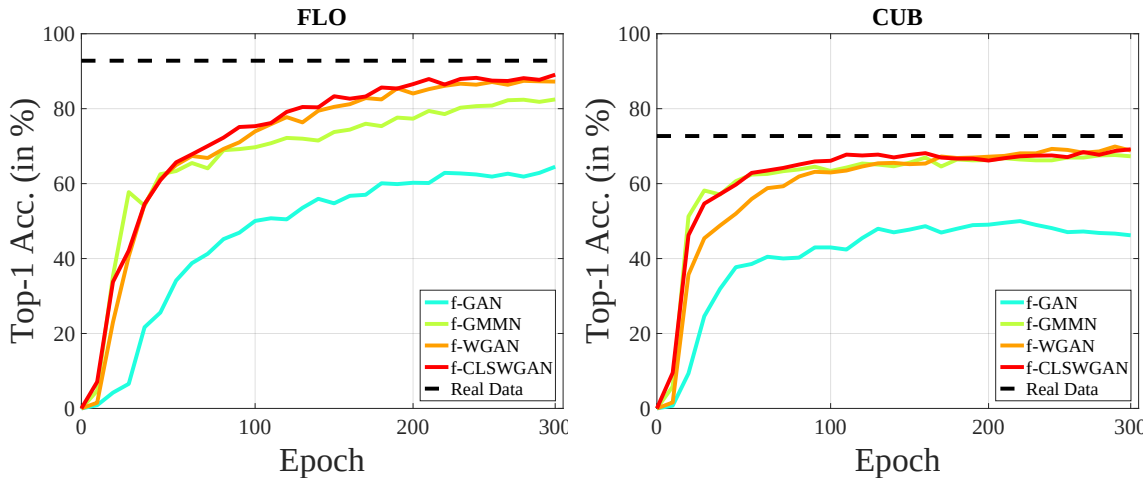

Figure 5.5: Measuring the seen class accuracy of the classifier trained on generated features of seen classes w.r.t. the training epochs (with softmax).

CNN	FG	**u**	**s**	**H**
GoogLeNet	none	20.2	35.7	25.8
	f-CLSWGAN	35.3	38.7	36.9
ResNet-101	none	23.7	62.8	34.4
	f-CLSWGAN	43.7	57.7	49.7

Table 5.3: GZSL results with GoogLeNet vs ResNet-101 features on CUB (CNN: Deep Feature Encoder Network, FG: Feature Generator, **u** = T1 on $\mathcal{Y}^u$, **s** = T1 on $\mathcal{Y}^s$, H = harmonic mean, "none"= no generated features).

leads to a significant accuracy boost while our f-WGAN and f-CLSWGAN improve over f-GMMN and almost reach the supervised upper bound.

After having established that our f-xGAN leads to a stable training performance and generating highly descriptive features, we evaluate the generalization ability of the f-xGAN generator to unseen classes. Using the pre-trained model, we generate CNN features of unseen classes. We then train a softmax classifier using these synthetic CNN features of unseen classes with real CNN features of seen classes. On the GZSL task, Figure 5.6 shows that increasing the number of generated features of unseen classes from 1 to 100 leads to a significant boost of accuracy, e.g. 28.2% to 56.5% on CUB and 37.9% to 66.5% on FLO. As in the case for generating seen class features, here the ordering is f-GAN $<$ f-WGAN $<$ f-GMMN $<$ f-CLSWGAN on CUB and f-GAN $<$ f-GMMN $<$ f-WGAN $<$ f-CLSWGAN on FLO. With these results, we argue that if the generative model can generalize well to previously unseen data distributions, e.g. perform well on GZSL task, they have practical use in a wide range of real-world applications. Hence, we propose to quantitatively evaluate the performance of generative models on the GZSL task.

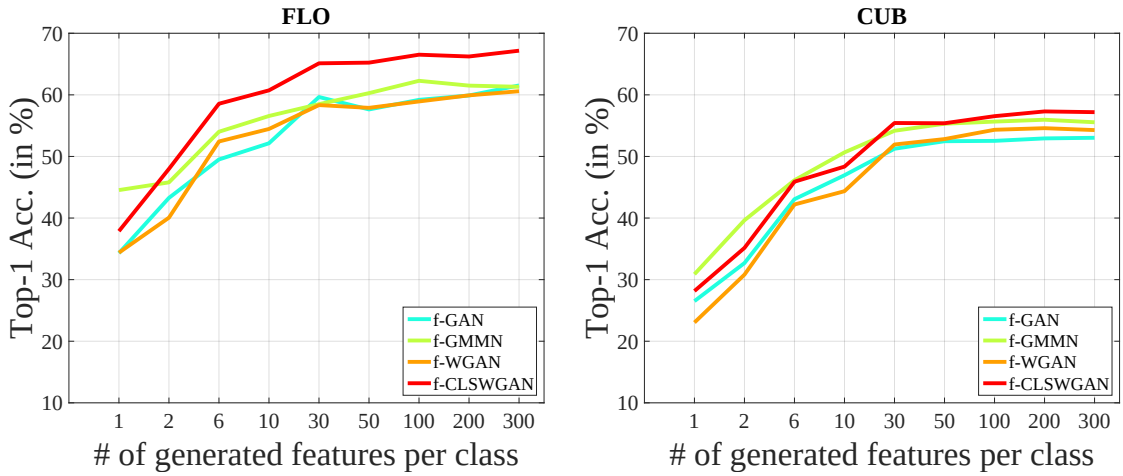

Figure 5.6: Increasing the number of generated f-xGAN features wrt unseen class accuracy (with softmax) in ZSL.

C	FG	**u**	**s**	**H**
Attribute (att)	none	23.7	62.8	34.4
	f-CLSWGAN	43.7	57.7	49.7
Sentence (stc)	none	38.8	53.8	45.1
	f-CLSWGAN	50.3	58.3	54.0

Table 5.4: GZSL results with conditioning f-xGAN with stc and att on CUB (C: Class embedding, FG: Feature Generator, **u** = T1 on $\mathcal{Y}^u$, **s** = T1 on $\mathcal{Y}^s$, H = harmonic mean, "none"= no generated features).

Effect of CNN Architectures. The aim of this study is to determine the effect of the deep CNN encoder that provides real features to our f-xGAN discriminator. In Table 5.3, we first observe that with GoogLeNet features, the results are lower compared to the ones obtained with ResNet-101 features. This indicates that ResNet-101 features are stronger than GoogLeNet, which is expected. Besides, most importantly, with both CNN architectures we observe that our f-xGAN outperforms the "none" by a large margin. Specifically, the accuracy increases from 25.8% to 36.9% for GoogleNet features and 34.4% to 49.7% for ResNet-101 features. Those results are encouraging as they demonstrate that our f-xGAN is not limited to learning the distribution of ResNet-101 features, but also able to learn other feature distributions.

Effect of Class Embeddings. The conditioning variable, i.e. class embedding, is an important component of our f-xGAN. Therefore, we evaluate two different class embeddings, per-class attributes (att) and per-class sentences (stc) on CUB as this is the only dataset that has both. In Table 5.4, we first observe that f-CLSWGAN features generated with att not only lead to a significantly higher result (49.7% vs 34.4%), **s** and **u** are much more balanced (57.7% and 43.7% vs. 62.8% and 23.7%)

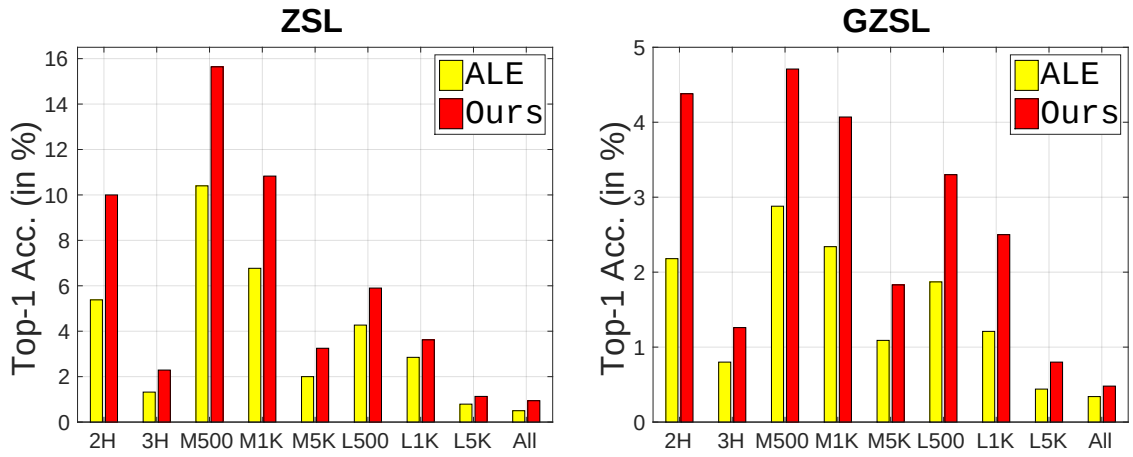

Figure 5.7: ZSL and GZSL results on ImageNet (ZSL: T1 on $\mathcal{Y}^u$, GZSL: T1 on $\mathcal{Y}^u$). The splits, ResNet features and Word2Vec are provided by (Xian *et al.*, 2017). "Ours" = feature generator: f-CLSWGAN, classifier: softmax.

compared to the state-of-the-art, i.e. "none". This is because generated CNN features help us explore the space of unseen classes whereas the state of the art learns to project images closer to seen class embeddings.

Finally, f-CLSWGAN features generated with per-class stc significantly improve results over att, achieving 54.0% in H measure, and also leads to a notable **u** of 50.3% without hurting **s** (58.3%). This is due to the fact that stc leads to high quality features (Reed *et al.*, 2016a) reflecting the highly descriptive semantic content language entails and it shows that our f-CLSWGAN is able to learn higher quality CNN features given a higher quality conditioning signal.

5.4.3 Large-Scale Experiments

Our large-scale experiments follow the same zero-shot data splits of (Xian *et al.*, 2017) and serve two purposes. First, we show the generalizability of our approach by conducting ZSL and GZSL experiments on ImageNet (Deng *et al.*, 2009) for it being the largest-scale single-label image dataset, i.e. with 21K classes and 14M images. Second, as ImageNet does not contain att, we use as a (weak) conditioning signal Word2Vec (Mikolov *et al.*, 2013b) to generate f-CLSWGAN features. Figure 6.3 shows that softmax as a classifier obtains the state-of-the-art of ZSL and GZSL on ImageNet, significantly improving over ALE (Akata *et al.*, 2015a). These results show that our f-CLSWGAN is able to generate high quality CNN features also with Word2Vec as the class embedding.

For ZSL, for instance, with the 2H split "Ours" almost doubles the performance of ALE (5.38% to 10.00%) and in one of the extreme cases, e.g. with L1K split, the accuracy improves from 2.85% to 3.62%. For GZSL the same observations hold, i.e. the gap between ALE and "Ours" is 2.18 vs 4.38 with 2H split and 1.21 vs 2.50 with L1K split. Note that, (Xian *et al.*, 2017) reports the highest results with

	CUB			FLO		
Generated Data	**u**	**s**	**H**	**u**	**s**	**H**
none	38.8	53.8	45.1	13.3	61.6	21.9
Image (with (Zhang *et al.*, 2017a))	0.2	69.4	0.4	10.5	95.4	18.9
CNN feature (Ours)	50.3	58.3	**54.0**	59.0	73.8	**65.6**

Table 5.5: Summary Table (**u** = T1 on $\mathcal{Y}^u$, **s** = T1 accuracy on $\mathcal{Y}^s$, H = harmonic mean, class embedding = stc). "none": ALE with no generated features.

SYNC (Changpinyo *et al.*, 2016) and "Ours" improves over SYNC as well, e.g. 9.26% vs 10.00% with 2H and 3.23% vs 3.56% with L1K. With these results we emphasize that with a supervision as weak as a Word2Vec signal, our model is able to generate CNN features of unseen classes and operate at the ImageNet scale. This does not only hold for the ZSL setting which discards all the seen classes from the test-time search space assuming that the evaluated images will belong to one of the unseen classes. It also holds for the GZSL setting where no such assumption has been made. Our model generalizes to previously unseen classes even when the seen classes are included in the search space which is the most realistic setting for image classification.

5.4.4 Feature vs Image Generation

As our main goal is solving the GZSL task which suffers from the lack of visual training examples, one naturally thinks that image generation serves the same purpose. Therefore, here we compare generating images and image features for the task of GZSL. We use the StackGAN (Zhang *et al.*, 2017a) to generate 256×256 images conditioned on sentences.

In Table 5.5, we compare GZSL results obtained with "none", i.e. with an ALE model trained on real images of seen classes, Image, i.e. image features extracted from 256×256 synthetic images generated by StackGAN (Zhang *et al.*, 2017a) and CNN feature, i.e. generated by our f-CLSWGAN. Between "none" and "Image", although the seen class accuracy improves, the unseen class accuracy is extremely low (0.2% for CUB and 10.5% for FLO) which shows that the generated images do not generalize to unseen classes. On average, i.e. the H measure, generating images of unseen classes leads to 0.4% on CUB and 18.9% accuracy on FLO whereas "none" leads to 45.1% on CUB and 21.9% accuracy on FLO. Upon visual inspection, we have observed that although many images have an accurate visual appearance as birds or flowers, they lack the necessary discriminative details to be classified correctly and the generated images are not class-consistent. On the other hand, generating CNN features leads to a significant boost of accuracy, e.g. 54.0% on CUB and 65.6% on FLO which is clearly higher than having no generation, i.e. "none", and image generation.

We argue that image feature generation has the following advantages. First, the

number of generated image features is limitless. Second, the image feature generation learns from compact invariant representations obtained by a deep network trained on a large-scale dataset such as ImageNet, therefore the feature generative network can be quite shallow and hence computationally efficient. Third, generated CNN features are highly discriminative, i.e. they lead to a significant boost in performance of both ZSL and GZSL. Finally, image feature generation is a much easier task as the generated data is much lower dimensional than high quality images necessary for discrimination.

5.5 CONCLUSION

In this work, we propose f-CLSWGAN, a learning framework for feature generation followed by classification, to tackle the generalized zero-shot learning task. Our f-CLSWGAN model adapts the conditional GAN architecture that is frequently used for generating image pixels to generate CNN features. In f-CLSWGAN, we improve WGAN by adding a classification loss on top of the generator, enforcing it to generate features that are better suited for classification. In our experiments, we have shown that generating features of unseen classes allows us to effectively use softmax classifiers for the GZSL task.

Our framework is generalizable as it can be integrated to various deep CNN architectures, i.e. GoogleNet and ResNet as a pair of the most widely used architectures. It can also be deployed with various classifiers, e.g. ALE, SJE, DEVISE, LATEM, ESZSL that constitute the state of the art for ZSL but also the GZSL accuracy improvements obtained with softmax is important as it is a simple classifier that could not be used for GZSL before this work. Moreover, our features can be generated via different sources of class embeddings, e.g. Sentence, Attribute, Word2vec, and applied to different datasets, i.e. CUB, FLO, SUN, AWA being fine and coarse-grained ZSL datasets and ImageNet being a truly large-scale dataset.

Finally, based on the success of our framework, we motivated the use of GZSL tasks as an auxiliary method for evaluation of the expressive power of generative models in addition to manual inspection of generated image pixels which is tedious and prone to errors. For instance, WGAN (Gulrajani *et al.*, 2017) has been proposed and accepted as an improvement over GAN (Goodfellow *et al.*, 2014). This claim is supported with evaluations based on manual inspection of the images and the inception score. Our observations in Figure 5.4 and in Figure 5.6 support this and follow the same ordering of the models, i.e. WGAN improves over GAN in ZSL and GZSL tasks. Hence, while not being the primary focus of this chapter, we strongly argue, that ZSL and GZSL are suited well as a testbed for comparing generative models.

6 ENHANCED FEATURE GENERATION FRAMEWORKS FOR LOW-SHOT LEARNING

Contents

In Chapter 5, we show that feature generation is an effective way to tackle the data imbalance issue. Therefore in this chapter, we extend this idea to any-shot learning i.e., few-shot and zero-shot learning. We improve the feature generator f-CLSWGAN of Chapter 5 in two ways. First, our combine GANs and VAE to construct a stronger generative model. Second, our model additionally adds a descriminator that learns marginal distribution of novel classes from their unlabeled examples. Our proposed approach achieves the SOTA on the zero-shot learning benchmark introduced in Chapter 4.

The previous chapters including this chapter are all about image classification. In the next two chapters, we will move our attention to more complicated tasks including the semantic segmentation in Chapter 7 and video classification in Chapter 8 in the context of zero-shot and few-shot learning.

6.1 INTRODUCTION

Learning with limited labels has been an important topic of research as it is unrealistic to collect sufficient amounts of labeled data for every object. Recently, generating visual features of previously unseen classes (e.g. Xian *et al.*, 2018; Bucher *et al.*, 2017; Kumar Verma *et al.*, 2018b; Felix *et al.*, 2018a) has shown its potential to perform well on extremely imbalanced image collections. However, current feature generation approaches have still shortcomings. First, they rely on simple generative models which are not able to capture complex data distributions. Second, in many cases, they do not truly generalize to the under represented classes. Third, although classifiers

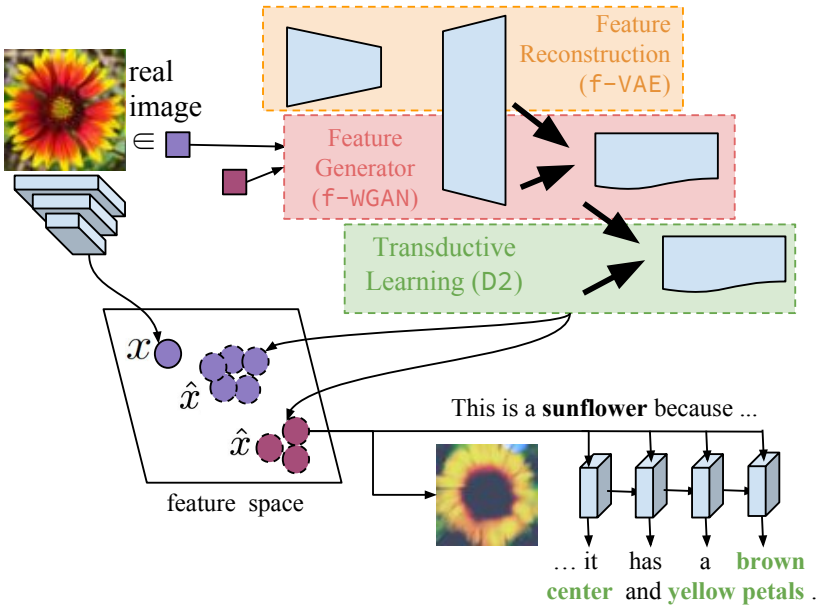

Figure 6.1: Our any-shot feature generating framework learns discriminative and interpretable CNN features from both labeled data of seen and unlabeled data of novel classes.

trained on a combination of real and generated features obtain state-of-the-art results, generated features may not be easily interpretable.

Our main focus in this work is a new model that generates visual features of any class, utilizing labeled samples when they are available and generalizing to unknown concepts whose labeled samples are not available. Prior work used GANs for this task (Xian *et al.*, 2018; Felix *et al.*, 2018a) as they directly optimize the divergence between real and generated data, but they suffer from mode collapse issues (Arjovsky and Bottou, 2017). On the other hand, feature generation with VAE (Kumar Verma *et al.*, 2018b) is more stable. However, VAE optimizes the lower bound of log likelihood rather than the likelihood itself (Kingma and Welling, 2014). Our model combines the strengths of VAE and GANs by assembling them to a conditional feature generating model, called f-VAEGAN-D2, that synthesizes CNN image features from class embeddings, i.e. class-level attributes or word2vec (Mikolov *et al.*, 2013b). Thanks to its additional discriminator that distinguishes real and generated features, our f-VAEGAN-D2 is able to use unlabeled data from previously unseen classes without any condition. The features learned by our model, e.g. Figure 8.1, are disciminative in that they boost the performance of any-shot learning as well as being visually and textually interpretable.

Our main contributions are as follows. (1) We propose the f-VAEGAN-D2 model that consists of a conditional encoder, a shared conditional decoder/generator, a conditional discriminator and a non-conditional discriminator. The first three networks aim to learn the conditional distribution of CNN image features given class embeddings optimizing VAE and WGAN losses on labeled data of seen classes. The

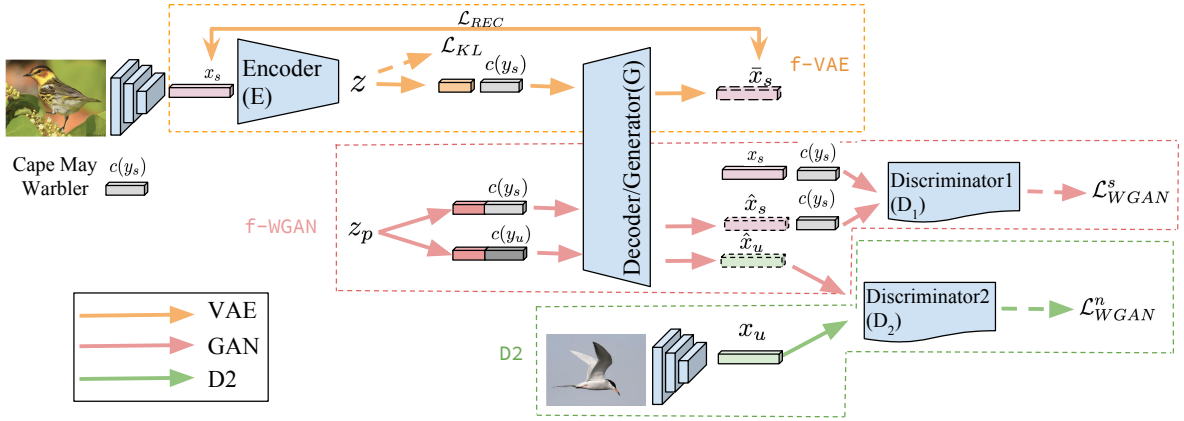

Figure 6.2: Our any-shot feature generating network (f-VAEGAN-D2) consist of a feature generating VAE (f-VAE), a feature generating WGAN (f-WGAN) with a conditional discriminator (D_1) and a transductive feature generator with a non-conditional discriminator (D_2) that learns from both labeled data of seen classes and unlabeled data of novel classes.

last network learns the marginal distribution of CNN image features on the unlabeled features of novel classes. Once trained, our model synthesizes discriminative image features that can be used to augment softmax classifier training. (2) Our empirical analysis on CUB, AWA2, SUN, FLO, and large-scale ImageNet shows that our generated features improve the state-of-the-art in low-shot regimes, i.e. (generalized) zero- and few shot learning in both the inductive and transductive settings. (3) We demonstrate that our generated features are interpretable by inverting them back to the raw pixel space and by generating visual explanations.

6.2 RELATED WORK

In this section, we discuss related works on generative models. We will not repeat the zero-shot and few-shot learning works that have been discussed in Chapter 2.

Generative Models. Generative modeling aims to learn the probability distribution of data points such that we can randomly sample data from it that can be used as a data augmentation mechanism. Generative Adversarial Networks (GANs)(Goodfellow *et al.*, 2014; Mirza and Osindero, 2014; Radford *et al.*, 2016) consist of a generator that synthesizes fake data and a discriminator that distinguishes fake and real data. The instable training issues of GANs have been studied by (Gulrajani *et al.*, 2017; Arjovsky and Bottou, 2017; Miyato *et al.*, 2018). An interesting application of GANs is CycleGAN (Zhu *et al.*, 2017) that translates an image from one domain to another domain. (Reed *et al.*, 2016c) generates natural images from text descriptions, and SRGAN(Ledig *et al.*, 2017) solves single image super-resolution. Variational Autoencoder (VAE) (Kingma and Welling, 2014) employs an encoder that represents the input as a latent variable with Gaussian distribution assumption and a decoder that

reconstructs the input from the latent variable. GMMN (Li *et al.*, 2015) optimizes the maximum mean discrepancy (MMD) (Gretton *et al.*, 2007) between real and generated distribution. Recently, generative models (Bucher *et al.*, 2017; Zhu *et al.*, 2018b; Kumar Verma *et al.*, 2018b; Xian *et al.*, 2018) have been applied to solve generalized zero-shot learning by synthesizing CNN features of unseen classes from semantic embeddings. Among those, (Bucher *et al.*, 2017) uses GMMN (Li *et al.*, 2015), (Zhu *et al.*, 2018b; Xian *et al.*, 2018) use GANs(Goodfellow *et al.*, 2014) and (Kumar Verma *et al.*, 2018b) employs VAE (Kingma and Welling, 2014). Our model combines the advantages of both VAE and GAN with an additional discriminator to use unlabeled data of unseen classes which lead to more discriminative features.

6.3 f-VAEGAN-D2 model

Existing models that operate on sparse data regimes are either trained with labeled data from a set of classes which is disjoint from the set of classes at test time, i.e. inductive zero-shot setting (e.g. Lampert *et al.*, 2013; Frome *et al.*, 2013), or the samples can come from all classes but then their labels are not known, i.e. transductive zero-shot setting (e.g. Fu *et al.*, 2015a; Rohrbach *et al.*, 2013). Recent works (e.g. Xian *et al.*, 2018; Kumar Verma *et al.*, 2018b; Felix *et al.*, 2018a) address generalized zero-shot learning by generating synthetic CNN features of unseen classes followed by training softmax classifiers, which alleviates the imbalance between seen and unseen classes. However, we argue that those feature generating approaches are not expressive enough to capture complicated feature distributions in real world. In addition, since they have no access to any real unseen class features, there is no guarantee on the quality of generated unseen class features. As shown in Figure 7.2, we proposes to enhance the feature generator by combining VAE and GANs with shared decoder and generator, and adding another discriminator (D_2) to distinguish real or generated features without applying any condition. Intuitively, in transductive zero-shot setting, by feeding real unlabeled features of unseen classes, D_2 will be able to learn the manifold of unseen class such that more realistic features can be generated. Hence, the key to our approach is the ability to generate semantically rich CNN feature distributions, which is generalizes to any-shot learning scenarios ranging from (generalized) zero-shot to (generalized) few-shot to (generalized) many-shot learning.

Setup. We are given a set of images $X = \{x_1, \ldots, x_l\} \cup \{x_{l+1}, \ldots, x_t\}$ encoded in the image feature space $\mathcal{X}$, a seen class label set Y^s, a novel label set Y^n, a.k.a unseen class label set Y^u in the zero-shot learning literature. The set of class embeddings $C = \{c(y) | \forall y \in Y^s \cup Y^n\}$ are encoded in the semantic embedding space $\mathcal{C}$ that defines high level semantic relationships between classes. The first l points $x_s (s \leq l)$ are labeled as one of the seen classes $y_s \in Y^s$ and the remaining points $x_n (l+1 \leq n \leq t)$ are unlabeled, i.e. may come from seen or novel classes.

In the inductive setting, the training set contains only labeled samples of seen class images, i.e. $\{x_1, \ldots, x_l\}$. On the other hand, in the transductive setting, the

training set contains both labeled and unlabeled samples, i.e. $\{x_1, \ldots, x_l, x_{l+1}, \ldots, x_t\}$. For both inductive and transductive settings the inference is the same. In zero-shot learning, the task is to predict the label of those unlabeled points that belong to novel classes, i.e. $f_{zsl} : \mathcal{X} \rightarrow \mathcal{Y}^n$, while in the generalized zero-shot learning, the goal is to classify those unlabeled points that can be either from seen or novel classes, i.e. $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^n$. Few-shot and generalized few-shot learning are defined similarly.

Our framework can be thought of as a data augmentation scheme where arbitrarily many synthetic features of sparsely populated classes aid in improving the disciminative power of classifiers. In the following, we only detail our feature generating network structure as the classifier is unconstrained (we use linear softmax classifiers).

6.3.1 Baseline Feature Generating Models

In feature generating networks (f-WGAN) (Xian *et al.*, 2018) the generator $G(z, c)$ generates a CNN feature $\hat{x}$ in the input feature space $\mathcal{X}$ from random noise z_p and a condition c, and the discriminator $D(x, c)$ takes as input a pair of input features x and a condition c and outputs a real value, optimizing:

$$\begin{aligned} \mathcal{L}^s_{WGAN} =& \mathbb{E}[D(x, c)] - \mathbb{E}[D(\tilde{x}, c)] \\ & - \lambda \mathbb{E}[(||\nabla_{\hat{x}} D(\hat{x}, c)||_2 - 1)^2], \end{aligned} \tag{6.1}$$

where $\tilde{x} = G(z, c)$ is the generated feature and $\hat{x} = \alpha x + (1 - \alpha x)$ with $\alpha \sim U(0, 1)$ and λ is the penalty coefficient.

The feature generating VAE (Kingma and Welling, 2014) (f-VAE) consists of an encoder $E(x, c)$, which encodes an input feature x and a condition c to a latent variable z, and a decoder $Dec(z, c)$, which reconstructs the input x from the latent z and condition c optimizing:

$$\begin{aligned} \mathcal{L}^s_{VAE} =& KL(q(z|x, c)||p(z|c)) \\ & - \mathbb{E}_{q(z|x,c)}[\log p(x|z, c)], \end{aligned} \tag{6.2}$$

where the conditional distribution $q(z|x, c)$ is modeled as $E(x, c)$, $p(z|c))$ is assumed to be $\mathcal{N}(0, 1)$, KL is the Kullback-Leibler divergence, and $p(x|z, c)$ is equal to $Dec(z, c)$.

6.3.2 Our f-VAEGAN-D2 Model

It has been shown that ensembling a VAE and a GAN leads to better image generation results (Larsen *et al.*, 2016). We hypothesize that VAE and GAN learn complementary information for feature generation as well. This is likely when the target data follows a complicated multi-modal distribution where two losses are able to capture different modes of the data.

To combine f-VAE and f-WGAN, we introduce an encoder $E(x, c) : \mathcal{X} \times \mathcal{C} \rightarrow \mathcal{Z}$, which encodes a pair of feature and class embedding to a latent representation, and a discriminator $D_1 : \mathcal{X} \times \mathcal{C} \rightarrow \mathbb{R}$ maps this embedding pair to a compatibility score, optimizing:

$$\mathcal{L}^s_{VAEGAN} = \mathcal{L}^s_{VAE} + \gamma \mathcal{L}^s_{WGAN} \tag{6.3}$$

where the generator $G(z, c)$ of the GAN and decoder $Dec(z, c)$ of the VAE share the same parameters. The superscript s indicates that the loss is applied to feature and class embedding pair of seen classes. γ is a hyperparameter to control the weighting of VAE and GAN losses.

Furthermore, when unlabeled data of novel classes becomes available, we propose to add a non-conditional discriminator D_2 (D2 in f-VAEGAN-D2) which distinguishes between real and generated features of novel classes. This way D_2 learns the feature manifold of novel classes. Formally, our additional non-conditional discriminator $D_2 : \mathcal{X} \rightarrow R$ distinguishes real and synthetic unlabeled samples using a WGAN loss:

$$\begin{aligned} \mathcal{L}^n_{WGAN} =& \mathbb{E}[D_2(x_n)] - \mathbb{E}[D_2(\tilde{x}_n)] - \\ & \lambda \mathbb{E}[(||\nabla_{\hat{x}_n} D_2(\hat{x}_n)||_2 - 1)^2], \end{aligned} \tag{6.4}$$

where $\tilde{x}_n = G(z, y_n)$ with $y_n \in Y^n$, $\hat{x}_n = \alpha x_n + (1 - \alpha x_n)$ with $\alpha \sim U(0, 1)$. Since $\mathcal{L}^s_{WGAN}$ is trained to learn CNN features using labeled data conditioned on class embeddings of seen classes and class embeddings encode shared properties across classes, we expect these CNN features to be transferable across seen and novel classes. However, this heavily relies on the quality of semantic embeddings and suffers from domain shift problems. Intuitively, $\mathcal{L}^n_{WGAN}$ captures the marginal distribution of CNN features and provides useful signals of novel classes to generate transferable CNN features. Hence, our unified f-VAEGAN-D2 model optimizes the following objective function:

$$\min_{G,E} \max_{D_1,D_2} \mathcal{L}^s_{VAEGAN} + \mathcal{L}^n_{WGAN} \tag{6.5}$$

Implementation Details. Our generator (G) and discriminators (D_1 and D_2) are implemented as multilayer perceptron (MLP). The random Gaussian noise $z \sim N(0, 1)$ and class embedding $c(y)$ are concatenated and fed into the generator, which is composed of 2 fully connected layers with 4096 hidden units. We find dimension of noise $d_z = d_c$, i.e. dimension of class embeddings, works well. Similarly, the discriminators take input as the concatenation of image feature and class embedding and have 2 fully connected layers with 4096 hidden units. We use LeakyReLU as the nonlinear activation function except for the output layer of G, for which Sigmoid is used because we apply binary cross entropy loss as $\mathcal{L}_{REC}$ and input features are rescaled to be in $[0, 1]$. We find $\beta = 1$ and $\gamma = 1000$ works well across all the datasets. Gradient penalty coefficient is set to $\lambda = 10$ and generator is updated every 5 discriminator iterations as suggested in WGAN paper (Arjovsky *et al.*, 2017). As for the optimization, we use Adam optimizer with constant learning rate 0.001 and early stopping on the validation set.

	Model	ZSL	GZSL
INDUCTIVE	GAN	59.1	52.3
	VAE	58.4	52.5
	VAE-GAN	61.0	53.7
TRANSDUCTIVE	GAN	67.3	61.6
	VAE	68.9	59.6
	VAE-GAN	**71.1**	**63.2**

Table 6.1: Ablating different generative models on CUB (using attribute class embedding and image features with no fine-tuning). ZSL: top-1 accuracy on unseen classes, GZSL: harmonic mean of seen and unseen class accuracies.

6.4 EXPERIMENTS

In this section, we validate our approach in both zero-shot and few-shot learning. The details of the settings are provided in their respective sections.

6.4.1 (Generalized) Zero-shot Learning

We validate our model on five widely-used datasets for zero-shot learning, i.e. Caltech-UCSD-Birds (CUB) (Welinder *et al.*, 2010), Oxford Flowers (FLO) (Nilsback and Zisserman, 2008), SUN Attribute (SUN) (Patterson and Hays, 2012) and Animals with Attributes2 (AWA2) (Xian *et al.*, 2019b). Among those, CUB, FLO and SUN are medium scale, fine-grained datasets. AWA2, on the other hand, is a coarse-grained dataset. Finally we evaluate our model also on ImageNet (Deng *et al.*, 2009) with more than 14 million images and 21K classes as a large-scale and fine-grained dataset.

We follow the exact ZSL and GZSL splits as well as the evaluation protocol of (Xian *et al.*, 2019b) and for fair comparison we use the same image and class embeddings for all models. Briefly, image (with no image cropping or flipping) features are extracted from the 2048-dim top pooling units of 101-layer ResNet pretrained on ImageNet 1K. For comparative studies, we also fine-tune ResNet-101 on the seen class images of each dataset. As for class embeddings, unless otherwise specified, we use class-level attributes for CUB (312-dim), AWA2 (85-dim) and SUN(102-dim). For CUB and FLO, we also extract 1024-dim sentence embeddings of character-based CNN-RNN model (Reed *et al.*, 2016a) from fine-grained visual descriptions (10 sentences per image).

Ablation study. We ablate our model with respect to the generative model, i.e. using GAN, VAE or VAE-GAN in both inductive and transductive settings. Our conclusions from Table 8.4, are as follows. In the inductive setting VAE-GAN has an edge over both VAE and GAN, i.e. 59.1% and 58.4% vs 61.0% in ZSL setting.

		Zero-Shot Learning				Generalized Zero-Shot Learning											
		CUB	FLO	SUN	AWA	CUB			FLO			SUN			AWA		
	Method	**T1**	**T1**	**T1**	**T1**	**u**	**s**	**H**	**u**	**s**	**H**	**u**	**s**	**H**	**u**	**s**	**H**
IND	ALE	54.9	48.5	58.1	59.9	23.7	62.8	34.4	13.3	61.6	21.9	21.8	33.1	26.3	16.8	76.1	27.5
	CLSWGAN	57.3	67.2	60.8	68.2	43.7	57.7	49.7	59.0	73.8	65.6	42.6	36.6	39.4	57.9	61.4	59.6
	SE-GZSL	59.6	-	63.4	69.2	41.5	53.3	46.7	-	-	-	40.9	30.5	34.9	58.3	68.1	62.8
	Cycle-CLSWGAN	58.6	70.3	59.9	66.8	47.9	59.3	53.0	61.6	69.2	65.2	47.2	33.8	39.4	**59.6**	63.4	59.8
	Ours	61.0	67.7	64.7	**71.1**	48.4	60.1	53.6	56.8	74.9	64.6	45.1	**38.0**	41.3	57.6	70.6	63.5
	Ours-finetuned	**72.9**	**70.4**	**65.6**	70.3	**63.2**	**75.6**	**68.9**	**63.3**	**92.4**	**75.1**	**50.1**	37.8	**43.1**	57.1	**76.1**	**65.2**
TRAN	ALE-tran	54.5	48.3	55.7	70.7	23.5	45.1	30.9	13.6	61.4	22.2	19.9	22.6	21.2	12.6	73.0	21.5
	GFZSL	50.0	85.4	64.0	78.6	24.9	45.8	32.2	21.8	75.0	33.8	0.0	41.6	0.0	31.7	67.2	43.1
	DSRL	48.7	57.7	56.8	72.8	17.3	39.0	24.0	26.9	64.3	37.9	17.7	25.0	20.7	20.8	74.7	32.6
	UE-finetune	72.1	-	58.3	79.7	74.9	71.5	73.2	-	-	-	33.6	**54.8**	41.7	**93.1**	66.2	77.4
	Ours	71.1	89.1	70.1	**89.8**	61.4	65.1	63.2	78.7	87.2	82.7	**60.6**	41.9	**49.6**	84.8	88.6	86.7
	Ours-finetuned	**82.6**	**95.4**	**72.6**	89.3	**73.8**	**81.4**	**77.3**	**91.0**	**97.4**	**94.1**	54.2	41.8	47.2	86.3	**88.7**	**87.5**

Table 6.2: Comparing with the-state-of-the-art. Top: inductive methods (IND), Bottom: transductive methods (TRAN). Fine tuning is performed only on seen class images as this does not violate the zero-shot condition. We measure top-1 accuracy (**T1**) in ZSL setting, Top-1 accuracy on seen (**s**) and unseen (**s**) classes as well as their harmonic mean (**H**) in GZSL setting.

Adding unlabeled samples to the training set, i.e. transductive learning setting, is beneficial for all the generative models. As in the inductive setting VAE and GAN achieve similar results, i.e 67.3% and 68.9% for ZSL. Our VAE-GAN model leads to the state-of-the-art results, i.e. 71.1% in ZSL and 63.2% in GZSL confirming that VAE and GAN learn complementary representations. As VAE-GAN gives the highest accuracy in all settings, it is employed in all remaining results of the chapter.

Comparing with the state-of-the-art. In Table 6.2 we compare our model with the best performing recent methods on four zero-shot learning datasets on ZSL and GZSL settings.

In the inductive ZSL setting, our model both with and without fine-tuning outperforms the state-of-the art for all datasets. Our model with fine-tuned features establishes the new state-of-the-art, i.e. 72.9% on CUB, 70.4% on FLO, 65.6% on SUN and 70.3% on AWA. For the transductive ZSL setting, our model without fine-tuning on CUB is surpassed by UE-finetune of (Song *et al.*, 2018), i.e. 71.1% vs 72.1%. However, when we also fine-tune our features, we establish the new state-of-the-art on the transductive ZSL setting as well, i.e. 82.6% on CUB, 95.4% on FLO, 72.6% on SUN and 89.3% on AWA.

In the GZSL setting, we observe that feature generating methods, i.e. our model, CLSWGAN (Xian *et al.*, 2018), SE-GZSL (Kumar Verma *et al.*, 2018b), Cycle-CLSWGAN (Felix *et al.*, 2018a) achieve better results than others. This is due to the fact that data augmentation through feature generation leads to a more balanced data distribution such that the learned classifier is not biased to seen classes. Note that although UE (Song *et al.*, 2018) is not a feature generating method, it leads to strong results as this model uses additional information, i.e. it assumes that unlabeled test samples always come from unseen classes. Nevertheless, our model

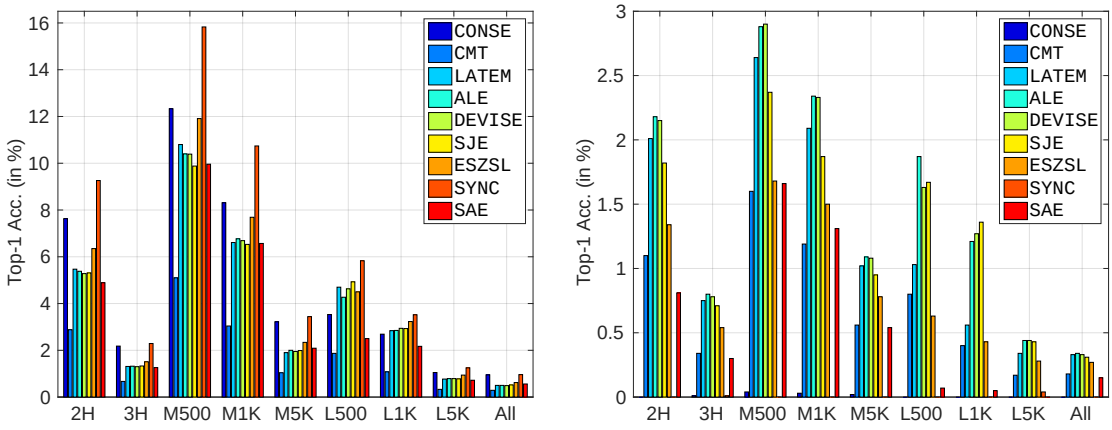

Figure 6.3: Top-1 ZSL results on ImageNet. We follow the splits in (Xian *et al.*, 2019b) and compare our results with the state-of-the-art feature generating model CLSWGAN (Xian *et al.*, 2018).

with fine-tuning leads to 77.3% harmonic mean (H) on CUB, 94.1% H on FLO, 47.2% H on SUN and 87.5% H on AWA achieving significantly higher results than all the prior works.

Large-scale experiments. Although most of the prior work presented in Table 6.2 has not been evaluated in ImageNet, this dataset serves a challenging and interesting test bed for (G)ZSL research. Hence, we compare our model with CLSWGAN (Xian *et al.*, 2018) on ImageNet using the same evaluation protocol. As shown in Figure 6.3 our model significantly improves over the state-of-the-art in both ZSL and GZSL settings in 2H, 3H and All splits determined by considering the classes 2 hops or 3 hops away from 1000 classes of Imagenet as well as all the remaining classes. These experiments are important for two reasons. First, they show that our feature generation model is scalable to the largest scale setting available. Second, our model is applicable to the situations even when human annotated attributes are not available, i.e. for ImageNet classes attributes are not available hence we use per-class word2vec representations.

6.4.2 (Generalized) Few-shot Learning

In few-shot or low-shot learning scenarios, classes are divided into base classes that have a large number of labeled training samples and novel classes that contain only few labeled samples per category. In the plain FSL setting, the goal is to achieve good performance on novel classes whereas in GFSL setting good performance must generalize to all classes.

Among the classic ZSL datasets, CUB has been used for few-shot learning in (Qi *et al.*, 2018) by taking the first 100 classes as base classes and the rest as novel classes. However, as ImageNet 1K contains some of those novel classes and feature extractors

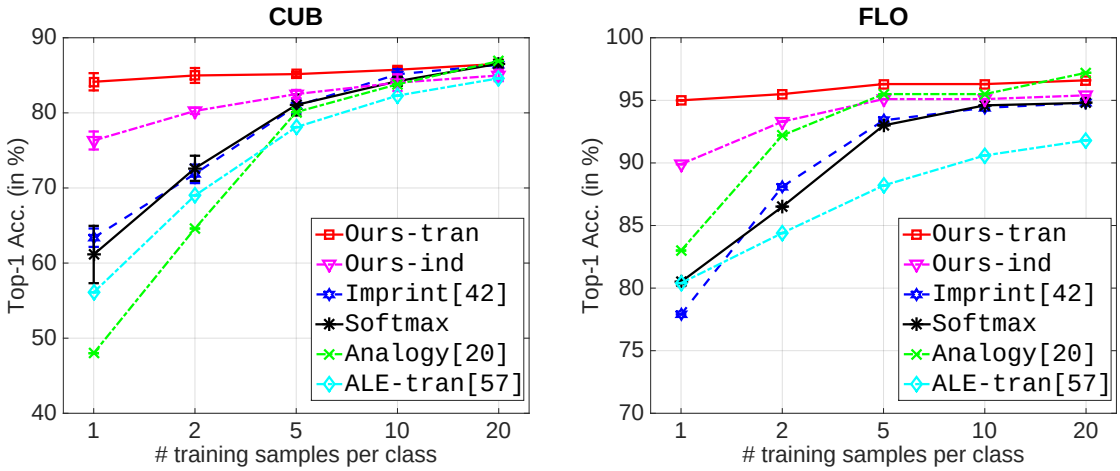

Figure 6.4: Few-Shot Learning (FSL) results on CUB and FLO with increasing number of training samples per novel class. We report the top-1 accuracy on novel classes.

are pretrained on it, we use the class splits from the standard ZSL setting, i.e. 150 base and 50 novel. For FLO we also follow the same class splits as in ZSL. As for features, we use the same fine-tuned ResNet-101 features and attribute class embeddings used in zero-shot learning experiments. For fairness, we repeat all the experiments for (Qi *et al.*, 2018) and (Hariharan and Girshick, 2017) with the same image features.

Comparing with the state-of-the-art. As shown in Figure 6.4 and Figure 6.5, both for FSL and GFSL settings and for both datasets, both our inductive and transductive models have a significant edge over all the competing methods when the number of samples from novel classes is small, e.g. 1,2 and 5. This shows that our model generates highly discriminative features even with only few real samples are present. In fact, only with one real sample per class, our model achieves almost the full accuracy obtained with 20 samples per class. Going towards the full supervised learning, e.g. with 10 or 20 samples per class, all methods perform similarly. This is expected since in the setting where a large number of labeled samples per class is available, then a simple softmax classifier that uses real ResNet-101 features achieves the state-of-the-art.

In the inductive FSL setting, our model that uses one labeled sample per class reaches the accuracy as softmax that uses five samples per class. In the transductive FSL setting, our model that uses one labeled sample per class reaches the accuracy of softmax obtained with 10 samples per class. Furthermore, the inductive GFSL setting, our model with two samples per class achieves the same accuracy as softmax trained with ten samples per class on CUB. In the transductive GFSL setting, for FLO, for our model only one labeled sample is enough to reach the accuracy obtained with 20 labeled samples with softmax. Note that the same behavior is observed on SUN and AWA as well. Due to space restrictions we present them in the supplementary material.

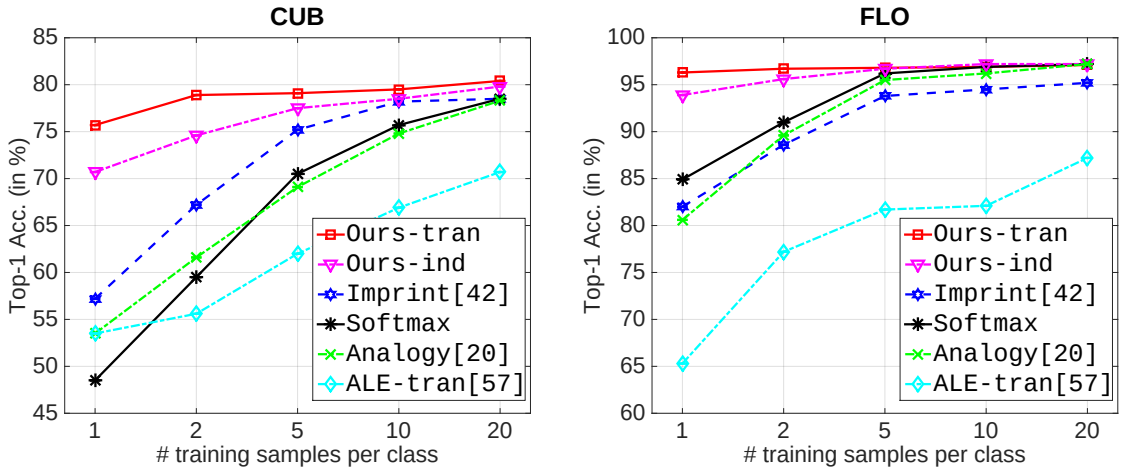

Figure 6.5: Generalized Few-Shot Learning (GFSL) results on CUB and FLO with increasing number of training samples per novel class. We report the top-1 accuracy on all classes.

Large-scale experiments. Regarding few-shot learning results on ImageNet, we follow the procedure in (Hariharan and Girshick, 2017) where 1K ImageNet categories are randomly divided into 389 base and 611 novel classes. To facilitate cross validation, base classes are further split into C^1_{base} (193 classes) and C^2_{base} (196 classes), and novel classes into C^1_{novel} (300 classes) and C^2_{novel} (311 classes). The cross validation of hyperparameters is performed on C^1_{base} and C^1_{novel} and the final results are reported on C^2_{base} and C^2_{novel}. Here, we extract image features from the ResNet-50 pretrained on $C^1_{base} \cup C^2_{base}$, which is provided by the benchmark (Hariharan and Girshick, 2017). Since there is no attribute annotation on ImageNet, we use 300-dim word2vec (Mikolov *et al.*, 2013b) embeddings as the class embedding. Following (Wang *et al.*, 2018c), we measure the averaged top-5 accuracy on test examples of novel classes with the model restricted to only output novel class labels, and the averaged top-5 accuracy on test examples of all classes with the model that predicts both base and novel classes.

Our baselines are PMN w/G* (Wang *et al.*, 2018c) combining meta-learning and feature generation, analogy generator (Hariharan and Girshick, 2017) learning an analogy-based feature generator and softmax classifier learned with uniform class sampling. For, few-shot learning results in Figure 6.6(left), we observe that our model in the transductive setting, i.e. Ours-tran improves the state-of-the-art PMN w/G* (Wang *et al.*, 2018c) significantly when the number of training samples is small, i.e. 1,2 and 5. Notably, we achieve 60.6% vs 54.7% state-of-the art at 1 shot, 70.3 vs 66.8% at 2 shots. This indicates that our model generates highly discriminative features by leveraging unlabeled data and word embeddings. In the challenging generalized few-shot learning setting (Figure 6.6 right), although PMN /G* (Wang *et al.*, 2018c) is quite strong by applying meta-learning (Snell *et al.*, 2017), our model still achieves comparable results with the state-of-the-art. It is also worth noting that PMN w/G* (Wang *et al.*, 2018c) cannot be directly applied to zero-shot learning.

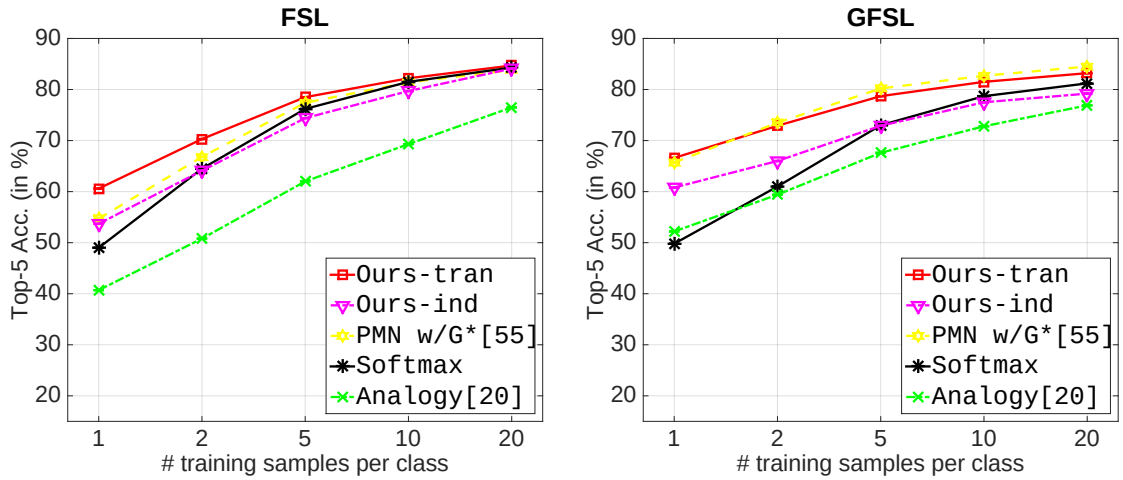

Figure 6.6: Few Shot Learning results on ImageNet with increasing number of training samples per novel class (Top-5 Accuracy). Left: FSL setting, Right: GFSL setting.

Hence, our approach is more versatile.

6.4.3 Interpreting Synthesized Features

In this section, we show that our generated features on FLO are visually discriminative and textually explainable.

Visualising generated features. A number of methods (Dosovitskiy and Brox, 2016a; Mahendran and Vedaldi, 2015; Dosovitskiy and Brox, 2016b) have explored strategies to generate images by inverting feature embeddings. We follow a strategy similar to (Dosovitskiy and Brox, 2016a) and train a deep upconvolutional neural network to invert feature embeddings to the image pixel space. We impose a L1 loss between the ground truth image and the inverted image, as well as a perceptual loss, by passing both images through a pre-trained Resnet101, and taking an L2 loss on the feature vectors at conv5_4 and average pooling layers. We also utilize an adversarial loss, by feeding the image and feature embedding to a discriminator, to improve our image quality. Our generator consists of a fully connected layer followed by 5 upconvolutional blocks. Each upconvolutional block contains an Upsampling layer, a 3x3 convolution, BatchNorm and ReLu non-linearity. The final size of the reconstructed image is 64x64. The discriminator processes the image through 4 downsampling blocks, the feature embedding is sent to a linear layer and spatially replicated and concatenated with the image embedding, and this final embedding is passed through a convolutional and sigmoid layer to get the probability that the sample is real or fake. We train this model on all the real feature-image pairs of the 102 classes, and use the trained generator to invert images from synthetic features.

In Figure 6.7, we show generated images from real and synthetic features for comparison. We observe that images generated from synthetic features contain the

Figure 6.7: Interpretability: visualizations by generating images and textual explanations from real or synthetic features. For every block, the top is the target, the middle is reconstructed from the real feature (R) of the target, the bottom is reconstructed from a synthetic feature (S) from the same class. We also generate visual explanations conditioned with the predicted class and the reconstructed real or synthetic images. Top (Middle): Features come from seen (unseen) classes. Bottom: classes with a large inter-class variation lead to poorer visualizations and explanations.

essential attributes required for classification, such as the general color distribution and sometimes even features like the petal and stamen are visible. Also, the image quality is similar for the images generated from real and synthetic features. Interestingly, the synthetic features of unseen classes generated by our model without observing any real features from that class, i.e. "Unseen classes" and "S" row, also yield pleasing reconstructions.

As shown in "Challenging Classes" of Figure 6.7, in some cases the generated images from synthetic features lack a certain level of detail, e.g. see images for "Balloon Flower" and in some cases the colors do not match with the real image, e.g. see images for "Sweat Pea". We noticed that these correspond to classes with high inter class variation.

Explaining visual features. We also explore generating textual explanations of our synthetic features. For this, we choose a language model (Hendricks *et al.*, 2016), that produces an explanation of why an image belongs to a particular class, given a feature embedding and a class label. The architecture of our model is similar to (Hendricks *et al.*, 2016), we use a linear layer for the feature embedding, and feed it as the start token for a LSTM. At every step in the sequence, we also feed the class embedding, to produce class relevant captions. The class embedding is obtained by

training a LSTM to generate captions from images, and taking the average hidden state for images of that class. A softmax cross entropy loss is imposed on the output using the ground truth caption. Also, a discriminative loss that encourages the generated sentence to belong to the relevant class is imposed by sampling a sentence from the LSTM and sending it to a pre-trained sentence classifier. The model is trained on the dataset from (Reed *et al.*, 2016a). As before, we train this model on all the real feature-caption pairs, and use it to obtain explanations for synthetic features.

In Figure 6.7, we show explanations obtained from real and synthetic features. We observe that the model generates image relevant and class specific explanations for synthetic features of both seen and unseen classes. For instance, a "King Protea" feature contains information about "red petals and pointy tips" while "Purple Coneflower" feature has information on "pink in color and petals that are drooping downward" which are the most visually distinguishing properties of this flower.

On the other hand, as shown at the bottom of the figure, for classes where image features lack a certain level of detail, the generated explanations have some issues such as repetitions, e.g. "trumpet shaped" and "star shape" in the same sentence and unknown words, e.g. see the explanation for "Balloon Flower".

6.5 CONCLUSION

In this work, we develop a transductive feature generating framework that synthesizes CNN image features from a class embedding. Our generated features circumvent the scarceness of the labeled training data issues and allow us to effectively train softmax classifiers. Our framework combines conditional VAE and GAN architectures to obtain a more robust generative model. We further improve VAE-GAN by adding a non-conditional discriminator that handles unlabeled data from unseen classes. The second discriminator learns the manifold of unseen classes and backpropagates the WGAN loss to feature generator such that it generalizes better to generate CNN image features for unseen classes.

Our feature generating framework is effective across zero-shot (ZSL), generalized zero-shot (GZSL), few-shot (FSL) and generalized few-shot learning (GFSL) tasks on CUB, FLO, SUN, AWA and large-scale ImageNet datasets. Finally, we show that our generated features are visually interpretable, i.e. the generated images by by inverting features into raw image pixels achieve an impressive level of detail. They are also explainable via language, i.e. visual explanations generated using our features are class-specific.

7 ZERO-LABEL AND FEW-LABEL SEMANTIC SEGMENTATION

Contents

In Chapters 3, 4, 5, and 6, we develop methods and define evaluation protocols for the image classification tasks. However, in fact, the long-tail issue almost appear in many computer vision applications. Semantic segmentation is one of the most fundamental problems in computer vision. As pixel-level labelling in this context is particularly expensive, there have been several attempts to reduce the annotation effort, e.g. by learning from image level labels and bounding box annotations. In this chapter, we take this one step further and propose zero- and few-label learning for semantic segmentation as a new task and propose a benchmark on the challenging COCO-Stuff and PASCAL VOC12 datasets. In the task of zero-label semantic image segmentation no labeled sample of that class was present during training whereas in few-label semantic segmentation only a few labeled samples were present. Solving this task requires transferring the knowledge from previously seen classes to novel classes. Our proposed semantic projection network (SPNet) achieves this by incorporating class-level semantic information into any network designed for semantic segmentation, and is trained in an end-to-end manner. Our model is effective in segmenting novel classes, i.e. alleviating expensive dense annotations, but also in adapting to novel classes without forgetting its prior knowledge, i.e. generalized zero- and few-label semantic segmentation.

In Chapter 8, we will take a further step to address the few-shot learning challenges in the video domain.

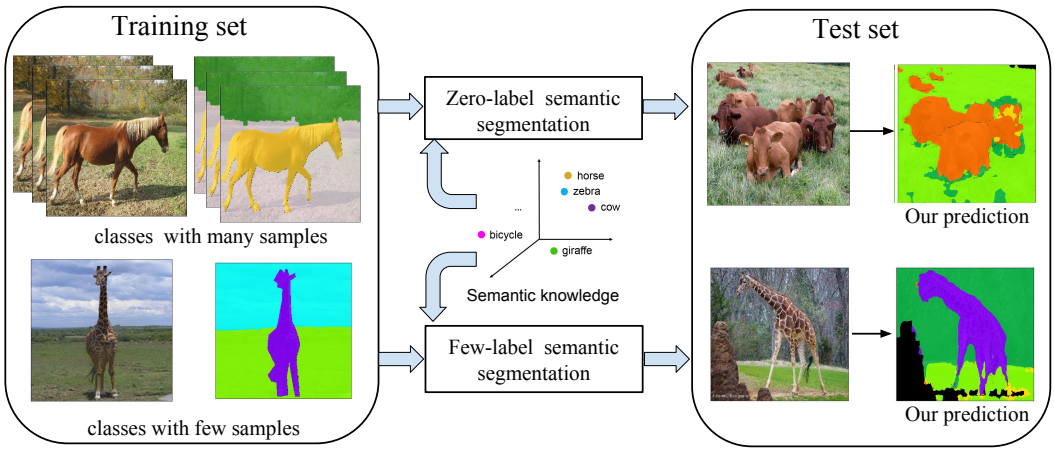

Figure 7.1: We propose (generalized) zero- and few-label semantic segmentation tasks, i.e. segmenting classes whose labels are not seen by the model during training or the model has a few labeled samples of those classes. To tackle these tasks, we propose a model that transfers knowledge from seen classes to unseen classes using side information, e.g. semantic word embedding trained on free text corpus.

7.1 INTRODUCTION

In semantic image segmentation the aim is assign a label to every pixel in an image by partitioning it into several semantic regions and then learning the appearance of various classes as well as the background. Although deep CNN-based approaches have achieved good performance for this task, they require costly dense annotations to learn their numerous parameters. Hence, leveraging weak annotations via image-level labels (Pathak *et al.*, 2015; Papandreou *et al.*, 2015; Oh *et al.*, 2017) or point (Bearman *et al.*, 2016), bounding box (Khoreva *et al.*, 2017), scribble-level annotations (Lin *et al.*, 2016) recently gained interest. On the other hand, as humans, we easily learn to recognize a previously unseen, i.e. novel, class by associating it with classes that we know. However, segmenting such novel classes via modern machine learning techniques is still an open problem as this process requires knowledge transfer from known classes to previously unseen ones.

Knowledge transfer to novel classes is not a new task. Learning to predict novel classes has been studied extensively in the context of image classification, i.e. zero-shot learning (Lampert *et al.*, 2013; Zhang and Saligrama, 2016; Changpinyo *et al.*, 2016; Akata *et al.*, 2015b). In zero-label semantic segmentation (ZLSS), our aim is to segment previously unseen, i.e. novel, classes, in few-label semantic segmentation (FLSS) these novel classes have a small number of labeled training examples (see Figure 7.1). In this work, we also aim for learning without forgetting the previously seen classes, i.e. generalized ZLSS and FLSS. To achieve these aims, we propose Semantic Projection Network (SPNet) that incorporates semantic word embeddings to an arbitrary semantic segmentation network inspired by the success of zero-shot learning. Prior models that tackle few-shot semantic segmentation (Shaban *et al.*,

2017; Dong and Xing, 2018) operate in the foreground-background segmentation setting. However, in our definition of FLSS the model has to predict all the classes in an image separately, which is more challenging and realistic. Our framework utilizes the similarity between different categories in a semantic segmentation network, enabling it to transfer learned representations to other classes. Consequently, our model is able to segment scenes containing novel classes.

Our main contributions are as follows. (1) We introduce the (generalized) zero-label and few-label semantic image segmentation task in a realistic settings inspired by zero-shot learning for image classification. (2) We propose semantic projection network (SPNet), an end-to-end semantic segmentation model which maps each image pixel to a semantic word embedding space where it is projected with a fixed word embedding to class probabilities optimizing the cross-entropy loss. (3) We create a benchmark for (generalized) zero- and few-label semantic image segmentation with two challenging datasets, i.e. COCO-Stuff and PASCAL-VOC. Our analysis shows that the SPNet model achieves impressive results both quantitatively and qualitatively in (generalized) zero-label and few-label tasks. Furthermore, as a side-product, our model improves the state of the art in zero-shot image classification demonstrating that it successfully generalizes to other tasks.

7.2 RELATED WORKS

In this section, we review prior work on semantic segmentation and its combination with zero-shot learning. Related works on zero-shot learning have been extensively discussed in Chapter 2 and will not be repeated here.

Semantic segmentation with weak supervision. Modern semantic segmentation systems (Long *et al.*, 2015; Chen *et al.*, 2018; Badrinarayanan *et al.*, 2017) are built on the encoder-decoder networks and trained with densely labeled annotations. Much efforts focus on improving semantic segmentation under fully supervised settings, e.g. adding global context information (Zhao *et al.*, 2017b; Zhang *et al.*, 2018a; Liu *et al.*, 2016), applying graphical models as a post-processing step to refine the output (Zheng *et al.*, 2015; Chen *et al.*, 2018), etc. On the other hand, weakly supervised semantic segmentation, i.e. reducing the annotation effort, has recently gained momentum. As weak supervision, prior works use image-level annotation (Pathak *et al.*, 2015; Papandreou *et al.*, 2015; Oh *et al.*, 2017), point (Bearman *et al.*, 2016), scribble (Lin *et al.*, 2016) and bounding box (Khoreva *et al.*, 2017) annotations. Those methods propagate the supervision to larger regions by measuring objectness (Bearman *et al.*, 2016) and saliency (Oh *et al.*, 2017), or applying graphical models (Lin *et al.*, 2016). Other methods refine the coarse annotated regions to more accurate ones (Khoreva *et al.*, 2017; Papandreou *et al.*, 2015). However, those models still require all the classes to be seen during training, thus cannot easily be adapted to new classes. In contrast, we focus on segmenting completely novel classes.

Semantic segmentation of novel classes. The term zero-shot semantic segmentation appears in prior works (Ji *et al.*, 2018a; Zhao *et al.*, 2017a). The aim of (Ji *et al.*, 2018a)

is to segment novel actor-action patterns during test time. While (Zhao *et al.*, 2017a) proposes open-vocabulary scene parsing task that segments novel objects by performing hierarchical parsing, we leverage word embeddings to predict the exact unseen classes and address the few-label problem in a unified framework. For few-shot semantic segmentation, previous approaches (Shaban *et al.*, 2017; Rakelly *et al.*, 2018; Dong and Xing, 2018; Zhang *et al.*, 2018b) follow the meta-learning setup (Vinyals *et al.*, 2016; Snell *et al.*, 2017), which uses a support set to predict an query image. However, those approaches are restricted to output a binary mask and fail to segment an image with multiple classes. In contrast, our approach is operating in the more realistic (generalized) few-label semantic segmentation setting, i.e. pixel-level labeling of an image where labels come from both base and novel classes.

Semantic embeddings. In learning with limited labels, some form of side information is required to transfer the knowledge learned from seen classes to unseen classes. One popular form of side information is attributes (Lampert *et al.*, 2013) that, however, require costly expert annotation. Thus, there has been a large group of studies (Akata *et al.*, 2015b; Reed *et al.*, 2016a; Qiao *et al.*, 2016; Ding *et al.*, 2017) utilizing other sources such as Word2vec (Mikolov *et al.*, 2013b), fastText (Joulin *et al.*, 2016a), or hierarchies (Miller, 1995) for building semantic embeddings. In this work, we utilize Word2Vec and fastText as they do not require dataset specific human annotation.

7.3 APPROACH

Modern semantic segmentation models are built on fully convolutional encoder-decoder architectures (Chen *et al.*, 2018; Long *et al.*, 2015) that output intermediate feature maps and posteriors for individual classes. However, to segment novel classes these models need to be adapted to transfer knowledge from one class to the other. Such knowledge can be obtained from class-level semantic embeddings associating different classes. Hence, the main insight of our approach is to leverage semantic word embeddings, i.e. word2vec (Mikolov *et al.*, 2013b) or fast-text (Joulin *et al.*, 2016a), to transfer knowledge learned from base classes to novel classes in a two-step process. First, we propose to learn a visual-semantic embedding module that produces intermediate feature maps in the word embedding space. Second, we project those feature maps into class probabilities via a fixed word embedding projection matrix. At test time, by replacing the projection matrix with word embeddings of novel classes, our model is able to segment unseen categories. Our model is trained end-to-end and can be incorporated into any semantic segmentation network, i.e. FCN (Long *et al.*, 2015) and deeplab (Chen *et al.*, 2018). We illustrate our overall pipeline in Figure 7.2.

Task formulation. We denote the set of seen classes as $\mathcal{S}$ and a disjoint set of unseen classes as $\mathcal{U}$. Let $\mathcal{D}_s = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}^s\}$ be our labeled training data of seen classes where x is an image in the image space $\mathcal{X}$, y is its corresponding label mask

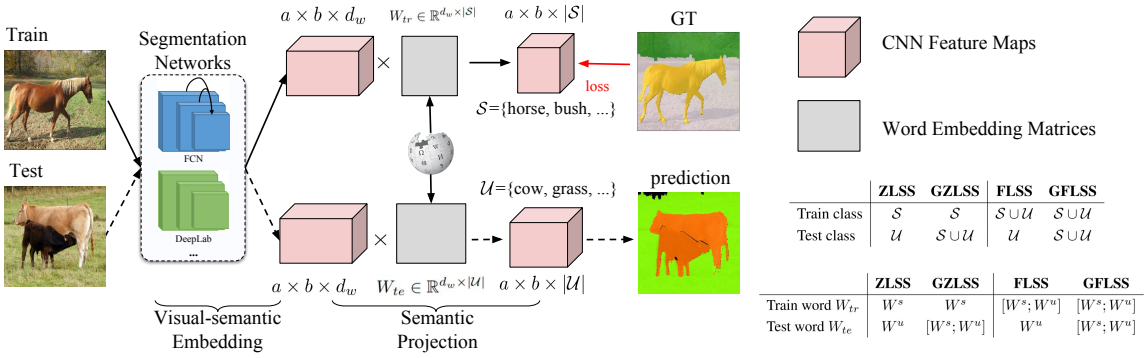

Figure 7.2: Our zero-label and few-label semantic segmentation model, i.e. SPNet, consists of two steps: visual semantic embedding and semantic projection. Zero-label semantic segmentation is drawn as an instance of our model. Replacing different components of SPNet, four tasks are addressed (Solid/dashed lines show the training/test procedures respectively).

in the dense label mask space $\mathcal{Y}^s \subset \mathcal{S}^{a*b}$ of seen classes with a and b being the height and the width of the image respectively. Similarly, we define the label mask space of unseen classes as $\mathcal{Y}^u \subset \mathcal{U}^{a*b}$. In addition, $W^s \in \mathbb{R}^{d_w \times |\mathcal{S}|}$ and $W^u \in \mathbb{R}^{d_w \times |\mathcal{U}|}$ denote the word embedding matrices of seen and unseen classes where d_w is the word embedding dimension. Given $\mathcal{D}_s$, W^s, and W^u, the task of zero-label semantic segmentation (ZLSS) is to learn a model that takes an image as an input and predicts the label of each pixel among unseen classes. A more realistic setting is generalized zero-label semantic segmentation (GZLSS) where the learned model predicts both seen and unseen classes. As for the (generalized) few-label semantic segmentation task, a few labeled samples from unseen classes $\mathcal{D}_u = \{(x, y) | x \in \mathcal{X}, y \in \mathcal{Y}^u\}$ are provided to the model during training. The test time target classes include only seen classes in few-label semantic segmentation (FLSS) whereas they include both seen and unseen classes in generalized few-label semantic segmentation (GFLSS). Here, we refer to the classes with a few labeled samples as unseen or novel, interchangeably. We summarize train class, test class and word embeddings used in different settings in Figure 7.2.

7.3.1 Semantic Projection Network (SPNet)

We address all four tasks with an unified model SPNet, which consists of two parts: visual-semantic embedding module and semantic projection layer.

i. Visual-semantic embedding module. This module is parameterized by a CNN and maps an input image $x \in \mathcal{X}$ into d_w feature maps via $\phi : \mathcal{X} \rightarrow \mathcal{R}^{a \times b \times d_w}$ of size $a \times b$. This is equivalent to embedding each pixel at (i, j) into a d_w dimensional class embedding vector $\phi(x)_{ij}$ that lies in the semantic embedding space shared by all the classes. The semantic embedding space constrains the output of the

visual-semantic embedding extractor ϕ and transfers knowledge from seen to unseen classes. Note that this is different from a standard CNN where pixels are mapped into an unconstrained feature space.

ii. Semantic projection layer. The semantic projection layer maps the feature embedding $\phi(x)_{ij}$ into unnormalized logit scores followed by a softmax activation that outputs the probability distribution over each training category,

$$p(\hat{y}_{ij} = s|x; W^s) = \frac{\exp\left(w_s^\top \phi(x)_{ij}\right)}{\sum_{c \in \mathcal{S}} \exp\left(w_c^\top \phi(x)_{ij}\right)} \tag{7.1}$$

where $\hat{y}_{ij}$ represents the prediction for pixel (i, j), w_c is the c-th column of W^s normalized to have unit length.

In contrast to standard CNNs that predict the class posterior by adding 1×1 convolution layer or fully connected layer with learnable weights, our classifier weights W^s are predefined by a word embedding model, e.g. word2vec (Mikolov *et al.*, 2013b), and then fixed during training. The W^s and the semantic projection layer estimate the compatibility between class prototypes and a feature embedding in terms of inner product similarity. Our proposed semantic projection layer is easy to implement by computing the tensor product between feature maps $\phi(x)$ and word embedding matrix W^s followed by the softmax activation function. After this layer, we directly optimize the standard cross-entropy loss over the spatial dimensions $(i, j) \in \mathcal{I}$,

$$\sum_{(i,j) \in \mathcal{I}} -\log p(\hat{y}_{ij} = y_{ij}|x) \tag{7.2}$$

which can be viewed as maximizing the negative log likelihood of predicting each pixel as its true label y_{ij}. Since there are no learnable parameters at the semantic projection layer, the optimization is over parameters of the visual-semantic embedding extractor ϕ. Compared to the standard semantic segmentation network, we have made subtle yet critical changes, i.e. mapping pixels to the semantic word embedding space followed by stacking a projection layer.

Inference. At the test time, in ZLSS and FLSS, we predict unseen classes by replacing the word embedding matrix in Eq. (7.1) with W^u. Each pixel label is predicted by:

$$\underset{u \in \mathcal{U}}{\operatorname{argmax}}\, p(\hat{y}_{ij} = u|x; W^u). \tag{7.3}$$

On the other hand, for GZLSS and GFLSS, we predict both seen and unseen class labels via their word embedding:

$$\underset{u \in \mathcal{S} \cup \mathcal{U}}{\operatorname{argmax}}\, p(\hat{y}_{ij} = u|x; [W^s; W^u]). \tag{7.4}$$

The extreme case of the imbalanced data problem occurs when there is no labeled training images of unseen classes, and this results in predictions being biased to seen

classes. To fix this issue, we follow (Chao *et al.*, 2016) and calibrate the prediction by reducing the scores of seen classes, which leads to:

$$\underset{u \in \mathcal{S} \cup \mathcal{U}}{\operatorname{argmax}}\, p(\hat{y}_{ij} = u | x; [W^s; W^u]) - \gamma \mathbb{I}[u \in \mathcal{S}] \tag{7.5}$$

where $\mathbb{I} = 1$ if u is a seen class and 0 otherwise, $\gamma \in [0, 1]$ is the calibration factor tuned on a held-out validation set.

Theoretically, the semantic projection layer allows our model to predict any class by simply copying its word embedding to the classifier weights. However, intuitively, the model can only perform well on the classes that share visual similarities with training classes. Hence, the word embedding ought to capture the similarity between classes.

Two-stage training in few-label setting. In our FLSS and GFLSS, we train a model with both D_s that includes a large number of samples per seen class and D_u that has only a few samples per unseen, i.e. novel, class. This is a typical imbalanced learning problem. The naive idea is to learn using both seen and unseen class samples within a mini-batch sampled uniformly from the whole training data. As expected, this leads to good performance on seen classes but inferior performance on unseen classes. Another strategy is to oversample unseen classes by first uniformly sampling a mini-batch of classes and selecting one sample from each of those classes. We found that this strategy remedies the imbalance issues to some extent but the results still remain unsatisfactory. On the other hand, fine-tuning the learned classifier on unseen class samples, i.e. after the initial optimization with only seen class samples, yields better results on unseen classes in FLSS as well as better overall results in GFLSS. Hence, we report our results in this setting.

7.3.2 Baseline: Hinge Visual-Semantic Loss (HVSL)

The choice of the loss function turns out to be important in zero-label semantic segmentation. Hence, in this section, we develop a baseline that shares the same embedding extractor ϕ as our SPNet but adopts the hinge visual-semantic loss instead of cross-entropy loss. Indeed hinge visual-semantic loss constitutes the most widely used loss function for zero-shot image classification (Akata *et al.*, 2015a; Bansal *et al.*, 2018; Frome *et al.*, 2013; Zhang and Saligrama, 2016; Xian *et al.*, 2016). In the context of semantic segmentation, we define the following hinge ranking loss for a single training example (x, y) as,

$$\sum_{(i,j) \in \mathcal{I}} \sum_{s \in \mathcal{S}} [\Delta(s, y_{ij}) + w_s^\top \phi(x)_{ij} - w_{y_{ij}}^\top \phi(x)_{ij}]_+ \tag{7.6}$$

where $\Delta(s, y_{ij}) = 1$ if $s \neq y_{ij}$ otherwise 0, $\phi(x)_{ij}$ is the visual-semantic embedding for pixel (i, j) in image x, y_{ij} is its corresponding ground-truth label. In practice, we follow (Frome *et al.*, 2013) to truncate the sum by randomly sampling one class that is not ground-truth.

7.4 EXPERIMENT

In this section, we present both quantitative and qualitative results of zero-label semantic segmentation and few-label semantic segmentation.

Datasets. We evaluate our model on the challenging COCO-stuff (Caesar *et al.*, 2018) and PASCAL-VOC 2012 (Everingham *et al.*) datasets. COCO-stuff has 164K images with dense pixel-level annotations from 172 classes including 80 thing classes, 91 stuff classes. PASCAL-VOC is a smaller dataset which contains 13K images from 20 classes.

Word embeddings. Encoding the semantic similarity between labels plays an important role in bridging the gap between seen and unseen class predictions. In this work, we study two different word embedding models, i.e. word2vec (Mikolov *et al.*, 2013b) trained on Google News (Wang *et al.*, 2018a) and fastText (Joulin *et al.*, 2016a) trained on Common Crawl (Mikolov *et al.*, 2018). The word embeddings of classes that contain multiple words are obtained by averaging the embeddings of each individual word.

Implementation details. We implement our SPNet model with PyTorch (Paszke *et al.*, 2017). We apply ImageNet pretrained VGG-16 (Simonyan and Zisserman, 2014b) and ResNet-101 (He *et al.*, 2016) as our backbone to extract features, and our model is built on the DeepLab-v2 (Chen *et al.*, 2018) that first extract features and apply atrous spatial pyramid pooling layer to produce the visual features, whose dimension is the same as the dimension of the semantic embedding space (i.e., 300 for fast-text and word2vec; 600 for their concatenation). In this work, for VGG backbone we apply Adam solver (Kingma and Ba, 2014) with initial learning rate 1.0×10^{-4}, and for ResNet we use SGD with initial learning rate 2.5×10^{-4}. Following (Chen *et al.*, 2018), we use the "poly" learning rate policy where current learning rate is the initial one multiplied by $(1 - \frac{iter}{max\_iter})^{power}$, and we set power to 0.9. Momentum and weight decay are set to 0.9 and .0005.

7.4.1 Zero-Label Semantic Segmentation Task

One of the contributions of our work is to propose a new task of zero-label semantic segmentation (ZLSS). In this section, we propose two benchmarks with zero-label data splits and detail the zero-label evaluation protocol.

Proposed zero-label dataset splits. The zero-label assumption, i.e. similar to the zero-shot assumption (Xian *et al.*, 2019b), states that none of the pixel values of the query images are allowed to belong to the classes that were used in any part of the training procedure, i.e. be it the model training or CNN training. This means that as CNNs are commonly trained on ImageNet 1K, none of the test classes should overlap with it. Following this rule, in COCO-Stuff dataset, we create a new zero-label class split by selecting 15 classes as unseen and the rest of the 167 classes as seen classes as they appear in ImageNet 1K which was used to pretrain ResNet.

	# classes		# images	
	train+val	**test**	**train+val**	**test**
COCO-Stuff	155+12	15	116287+2000	5000
PASCAL-VOC	12+3	5	11185 + 500	1449

Table 7.1: Statistics of data splits for COCO-Stuff and PASCAL-VOC datasets in terms of the number of classes and the number of images in the training and test splits.

In contrast to zero-shot image classification, we do not remove images that contain unseen classes from the training set, otherwise most of training images will be eliminated because seen and unseen classes co-occur frequently. Instead, we utilize the whole training set but ignore the labels of pixels belonging to unseen classes during training, i.e. these pixels do not effect the loss we optimize in any stage of the training. For PASCAL-VOC, since (a) only 4 classes are unseen in ImageNet 1K, (b) one of the candidate class 'person' has no semantically similar class present in the dataset, (c) all vehicles appear in ImageNet thus reducing candidate diversity - we simply take the first 15 classes as seen classes and the last 5 classes as unseen classes. We use the train/val split provided by the COCO-Stuff dataset: 118K training images as our training set and 5K validation images as our test set, and PASCAL-VOC: 11K training images and 1.4K test images. Following the cross-validation procedure of (Xian *et al.*, 2019b), we further hold out a subset of training classes as our validation set for tuning hyperparameters. More details about our data splits are shown in Table 7.1.

Evaluation protocol. The intersection-over-union (IoU), i.e. the standard evaluation criteria commonly used in semantic segmentation, quantizes the overlap between the predicted mask and the target mask. It is defined to be the size of the intersection between predicted and target regions divided by the union of them. For each class, its mean IoU is computed by averaging the IoU over all the query images.

In ZLSS, as the test-time search space is restricted to be unseen classes we report the mean IoU averaged over unseen classes. In GZLSS, the search space becomes the union of seen and unseen classes. In analogy to generalized zero-shot image classification (Xian *et al.*, 2019b), we report the mean IoU on seen classes, the mean IoU on unseen classes and the harmonic mean (H) of them, which is defined as,

$$H = \frac{2 * mIoU_{seen} * mIoU_{unseen}}{mIoU_{seen} + mIoU_{unseen}} \tag{7.7}$$

where $mIoU_{seen}$ and $mIoU_{unseen}$ represents the mean IoU of seen classes and unseen classes respectively. Similarly, in few-label semantic segmentation, we report mean IoU on unseen classes, but in generalized few-label semantic segmentation, the mean IoU over all classes is reported.

	fastText (ft)	word2vec (w2v)	ft + w2v
HVSL	25.8	25.3	31.8
SPNet	**33.1**	**32.1**	**35.2**

Table 7.2: Effect of word embeddings: Mean IoU of unseen classes in ZLSS with different word2vec, fastText and their combination on COCO-Stuff. Both HVSL and SPNet are based on ResNet101.

	COCO-Stuff	PASCAL VOC
SPNet-VGG	26.3	47.4
SPNet-ResNet101	**35.2**	**49.5**

Table 7.3: Effect of CNN architectures: ZLSS with different CNN architectures, i.e. VGG and ResNet101 on COCO-Stuff and PASCAL-VOC. Word embedding is the ft + w2v.

7.4.1.1 *SPNet Model Analysis for ZLSS*

In this section, we provide an extensive evaluation for different design choices of our model.

Effect of word embeddings. We compare our SPNet model with HVSL and study the effect of different word embeddings in Table 7.2. We investigate three types of word embeddings, i.e. fastText, word2vec and their concatenation. Our first observation is that SPNet performs significantly better than HVSL wrt. all the word embedding types, e.g. SPNet achieves 33.1 vs 25.8 with fastText, and 32.1 vs 25.3 with word2vec compared to HVSL. This implies that the cross-entropy loss is more suitable to the ZLSS task than hinge loss. Furthermore, we observe that fastText and word2vec achieve comparable results, and combining them significantly boosts the performance, e.g. mean IoU of SPNet are improved from 33.1 and 32.1 to 35.2. This indicates that fastText and word2vec contain complementary information. Hence, for the rest experiments, we use SPNet with fastText and word2vec combined.

Effect of CNN architectures. Our aim here is to compare different CNN architectures that are used as the backbone network to encode images in DeepLab-v2 (Chen *et al.*, 2018). Table 7.3 shows the ZLSS results with VGG16 (Simonyan and Zisserman, 2014b) and ResNet101 (He *et al.*, 2016). We first observe that with VGG16, the results are lower than with ResNet101 on both COCO-Stuff and PASCAL-VOC which implies that ResNet101 generate stronger features than VGG16 for this task. Besides, these results show that our SPNet achieves reasonably good results in ZLSS with both CNN architectures. Specifically, on COCO-stuff, SPNet obtains 26.3% mIoU with VGG16 and 35.2% mIoU with ResNet101. This is promising because our model does not require expensive dense pixel-level annotations for each class, e.g. it is not trained with any of the 15 unseen class labels of COCO-Stuff. This also indicates

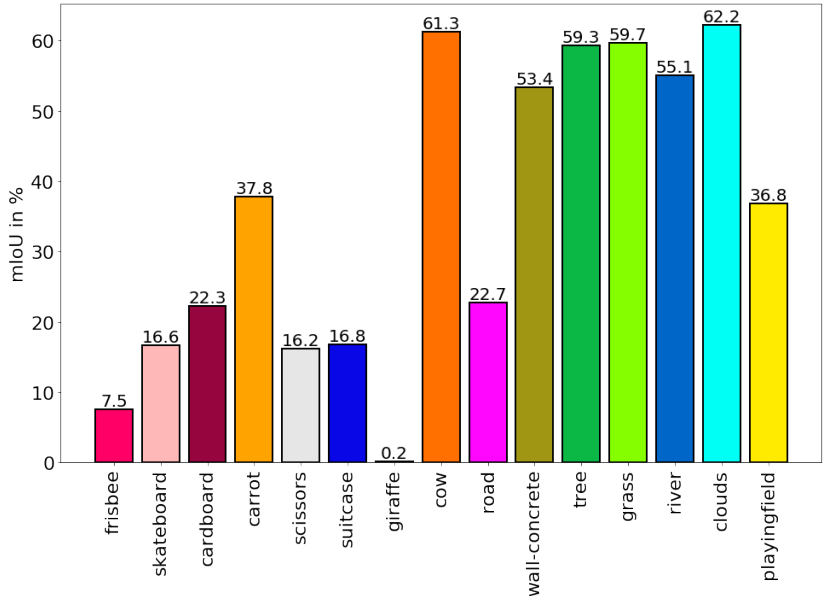

Figure 7.3: mIoU of unseen classes on COCO-Stuff ordered wrt average object size (left to right).

that our model is easily adapted to various semantic segmentation architectures.

Effect of the object size. We study the difficulty of zero-label semantic segmentation as a function of object sizes. Figure 7.3 presents a plot of per class mIoU score for the unseen classes in COCO-Stuff. The classes are ordered according to their average object sizes – with the largest on the right. It shows that there is a tendency that the performance is better for classes with larger objects. The plot also indicates that the knowledge transfer from seen to unseen classes is in general successful for the challenging stuff classes, such as, tree (59.3%), grass (59.7%, clouds (62.2%), considering the fact that they do not have semantically similar classes present in ImageNet 1K. We also observe that our model performs well for cow (61.3%) however the result is quite poor the other unseen animal class giraffe (0.2%).

7.4.1.2 *Generalized Zero-Label Semantic Segmentation*

GZLSS is a practical segmentation setting as the test time search space contains both seen and unseen classes, i.e. the pixel can be assigned to one of the seen or one of the unseen classes. Since the training images contain only labeled pixels of seen classes, at the test time, prediction will be biased to seen classes. Hence, this is a particularly challenging task. We alleviate this issue by using the calibrated classifier formulated in Eq. (7.5), which reduces the prediction scores of seen classes by a calibration factor γ. We select the optimal γ value based on the best harmonic mean IoU on a held-out validation set. Figure 7.4 shows the mean IoU on unseen classes, seen classes and their harmonic mean on COCO-Stuff and PASCAL VOC datasets.

On COCO-Stuff SPNet obtains 0.2% mean IoU on unseen classes while IoU on

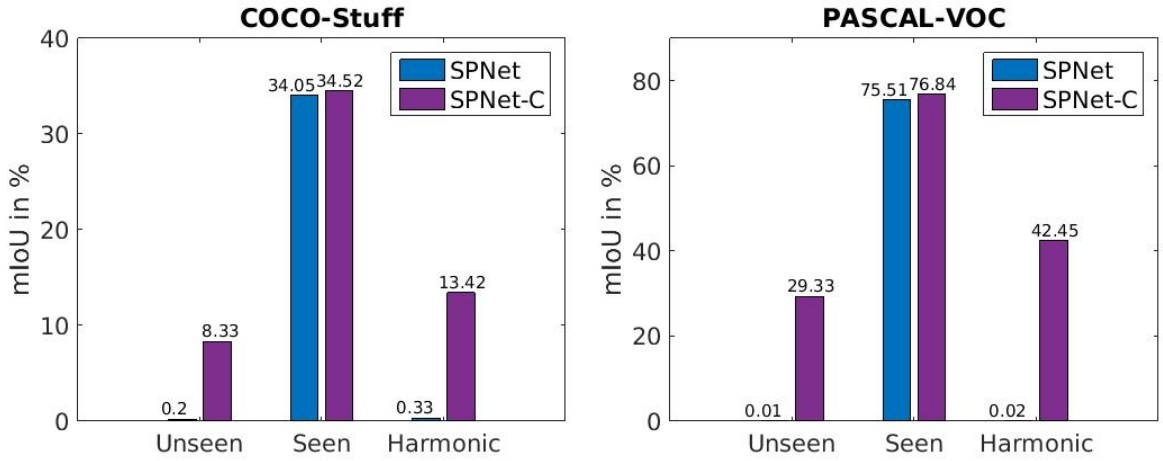

Figure 7.4: GZLSS results on COCO-Stuff and PASCAL-VOC. We report mean IoU of unseen classes, seen classes and their harmonic mean (perception model is based on ResNet101 and the semantic embedding is ft + w2v). SPNet-C represents SPNet with calibration.

	ZSL			GZSL		
	CUB	**SUN**	**AWA**	**CUB**	**SUN**	**AWA**
ALE	54.9	58.1	59.9	34.4	26.3	**27.5**
SJE	53.9	53.7	65.6	33.6	19.8	19.6
SYNC	56.3	55.6	54.0	19.8	13.4	16.2
GFZSL	49.3	60.6	**68.3**	0.0	0.0	3.5
SPNet	**56.5**	**60.7**	66.2	**36.6**	**39.6**	24.7

Table 7.4: SPNet loss on (generalized) zero-shot learning tasks. Top-1 accuracy on unseen classes is reported for ZSL and harmonic mean of seen and unseen classes is for GZSL.

seen classes is high, i.e. 34.05%. This is expected, in fact the same trend is observed in generalized zero-shot image classification task (Xian *et al.*, 2019b; Chao *et al.*, 2016). On the other hand, after calibration i.e. SPNet-C, on COCO-Stuff, mean IoU of unseen classes jumps to 8.33% while maintaining high mIoU on seen classes, i.e. 34.52% and overall SPNet-C achieves a harmonic mean of 13.42%. This is due to the fact that after calibration, i.e. reducing prediction scores of seen classes, pixels get predicted as seen classes less frequently.

On PASCAL-VOC we observe a similar trend. While SPNet performs poorly on unseen classes, i.e. 0.01% mIoU, with calibration this increases to 29.33% mIoU. Accordingly, SPNet-C achieves an impressive 42.45% harmonic mIoU. These results demonstrate that our SPNet does not only tackle ZLSS but also can handle the more practical GZLSS via predictor calibration.

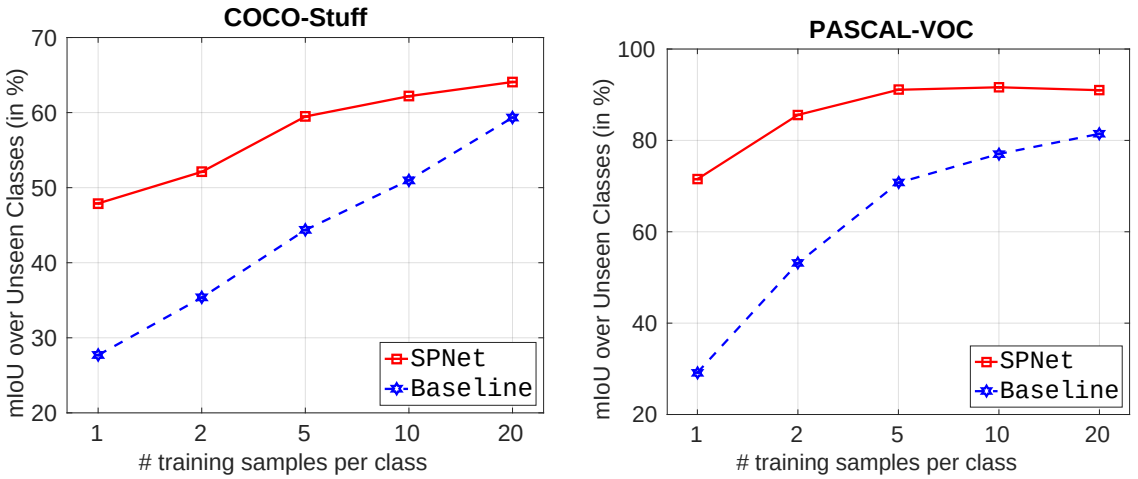

Figure 7.5: Few-label semantic segmentation (FLSS) on COCO-Stuff and PASCAL VOC with increasing number of training samples per class, i.e. $n \in \{1, 2, 5, 10, 20\}$.

7.4.1.3 *(Generalized) Zero-Shot Image Classification*

We evaluate our SPNet on the zero-shot image classification task on three benchmark datasets, i.e. CUB (Welinder *et al.*, 2010) (200 types of birds with 312 attributes), SUN (Patterson and Hays, 2012) (717 scenes with 102 attributes) and AWA (Lampert *et al.*, 2013) (50 classes of animals with 85 attributes) with various sizes and complexities, following the data splits and evaluation protocol of (Xian *et al.*, 2019b). We train SPNet with cross-entropy loss:

$$L(x, y) = -\log \frac{\exp\left(\phi(x)^\top V w_y\right)}{\sum_{c \in \mathcal{S}} \exp\left(\phi(x)^\top V w_c\right)} \tag{7.8}$$

where $\phi(x)$ is 2048-dim image feature extracted from a pre-trained ResNet101 (no fine-tuning on the task), $w_c \in \mathbb{R}^{d_w}$ is the class attribute of class c, $V \in \mathbb{R}^{2048 \times d_w}$ is the linear embedding we aim to learn. Table 7.4 shows that both in ZSL and GZSL settings, our SPNet improves over the state of the art on both CUB and SUN while it obtains the second best results on AWA despite the simplicity of our model. Both ALE (Akata *et al.*, 2015a) and SJE (Akata *et al.*, 2015b) utilize the visual-semantic hinge loss, SYNC (Changpinyo *et al.*, 2016) align visual and semantic embedding space using manifold learning, and GFZSL (Verma and Rai, 2017) learns a generative model to capture the class conditional distribution. However, our SPNet simply projects image feature into the class embedding space and apply the standard softmax classifier with the class embedding being the weights.

7.4.2 Few-Label Semantic Segmentation Task

The (Generalized) few-label semantic segmentation (FLSS and GFLSS) tasks arise in many real-world applications since class distribution in semantic segmentation

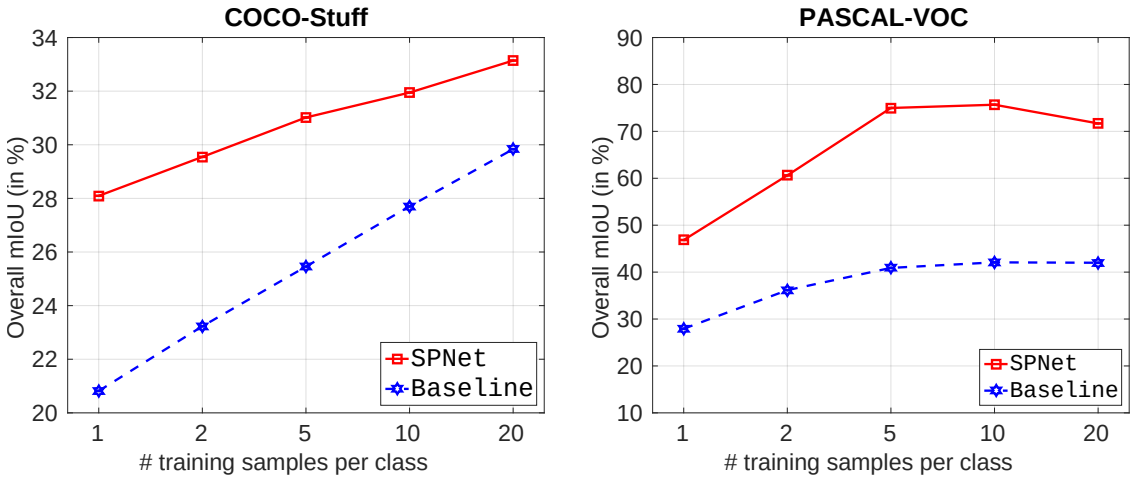

Figure 7.6: Generalized few-label semantic segmentation (GFLSS) on COCO-Stuff and PASCAL VOC with increasing number of training samples per class, i.e. $n \in \{1, 2, 5, 10, 20\}$.

is usually skewed, e.g. there are far more road pixels than bicycles. In contrast to ZLSS where the training set has no labeled example from unseen (novel) classes, in FLSS and GFLSS, the model is trained with all classes. At the evaluation time, the goal of FLSS is to segment only the novel classes, while GFLSS aims to segment both base and novel classes. For each novel class, we randomly draw $n \in \{1, 2, 5, 10, 20\}$ images that contain this class from the training set and disable ignore-label condition for those novel pixels. In addition, we develop a simple baseline based on the original DeepLab-v2 (Chen *et al.*, 2018), which is finetuned on novel classes after an initial optimization on base classes. We carry out experiments in FLSS and GFLSS with the baseline and our SPNet on COCO-Stuff and PASCAL-VOC.

In FLSS task, Figure 7.5 shows the comparison results with the baseline model (Chen *et al.*, 2018). Our SPNet yields significantly better results than the baseline in all cases on both COCO-Stuff and PASCAL VOC. In particular, when there is only 1 labeled example, our SPNet significantly outperforms the baseline, achieving a mean IoU of 47.90% over 27.69% in COCO-Stuff and 71.52% over 29.17% in PASCAL VOC on FZLSS. The accuracy improvement from 1 labeled sample to 5 labeled samples is significant, i.e. $\approx$ 20% mIoU for both COCO-Stuff and PASCAL VOC. These results demonstrate the effectiveness of our SPNet when the training samples are scarce.

As for GFLSS in Figure 7.6, a similar trend is observed. Our SPNet improves over DeepLab in all cases. The accuracy improvement is steady from 1 to $2, 5, 10, 20$ especially on COCO-Stuff. The difference between DeepLab and ours is 21.24% mIoU over both seen and unseen classes on PASCAL VOC when our model has access to only one labeled sample from novel classes.

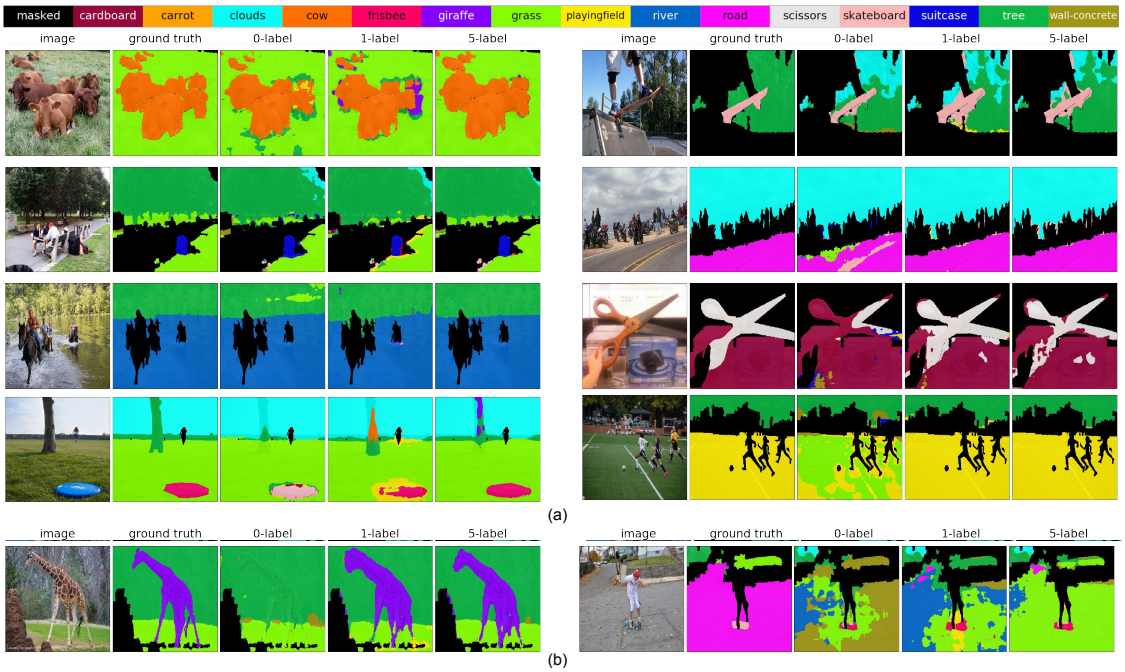

Figure 7.7: Qualitative results of our SPNet in 0-, 1- and 5-label semantic segmentation settings on COCO-Stuff on 15 novel classes (color coded at the top). Base classes are masked out with black color. (a) promising results (b) failure cases.

7.4.3 Qualitative Results

Figure 7.7 shows the qualitative results obtained by our SPNet in ZLSS and FLSS on COCO-Stuff. Our target 15 novel classes are encoded with the colors shown at the top. Base classes are masked out with black color. Some interesting results are as follows. In the first row and left column, our SPNet is already able to segment two previously unseen classes cows and grass at ZLSS, i.e. 0-label, and results get refined after the model sees more examples. It is also worth noting that our SPNet is able to predict stuff classes, such as road, river, clouds etc., in ZLSS setting. For instance, SPNet successfully segments clouds and roads in the image at the second row and right column, and perfectly segments the river in the image at the third row and left column. Another interesting result is in the left column of 4th row where the model correctly segments the frisbee in 0-label setting but incorrectly labels most pixels as 'skateboard' which in fact is another sports category object. On the other hand, some failure cases are shown in the bottom row. Our SPNet fails to predict giraffe at 0-label because shape and appearance of a giraffe vary significantly from seen classes. However, seeing only 1 example is enough to recognize and segment it, which demonstrates the ability of our SPNet in learning from few examples. Again, the result gets refined with 5 labeled examples.

These results support our observations in the previous sections and indicate that our SPNet, although simple, adapts its knowledge attained in previously seen examples to unseen ones.

7.5 CONCLUSIONS

In this work, we propose SPNet to semantically segment novel classes with no labeled examples or with only a few samples, within the new tasks of zero-label semantic segmentation and few-label semantic segmentation respectively. This model consists of a visual-semantic embedding module that encodes images in the word embedding space and a semantic projection layer that produces class probabilities. Our SPNet is both conceptually and computationally simple but surprisingly effective and end-to-end trainable. We have shown its applicability across zero-shot image classification to zero-label and few-label semantic segmentation tasks on various benchmark datasets.

8

GENERALIZED MANY-WAY FEW-SHOT VIDEO CLASSIFICATION

Contents

In Chapters 3, 4 and 5, we show that semantic embeddings can be used as an effective way for knowledge transfer on image classification tasks. We extend this idea to the semantic segmentation field in Chapter 6. While most of works for few-shot learning are in the image domain, there are many real-world applications that takes as input videos e.g., self-driving cars and video surveillance. Therefore, in this chapter, we study how to develop efficient methods for the few-shot video classification task where there are only few training examples per class. Our main idea is to improve the issues of video representation learning and lacking of training data. We argue that existing methods with 2D CNNs are unable to learn temporal information and thus develop a simple 3D CNN baseline, surpassing existing methods by a large margin. To circumvent the need of labelled examples, we propose to leverage weakly-labelled videos from a large dataset using video tag retrieval followed by selection of the best clips with visual similarities, yielding further improvement. Our results saturate current 5-way benchmarks for few-shot video classification and therefore we propose a more challenging benchmark involving more classes and a mixture of classes with varying supervision.

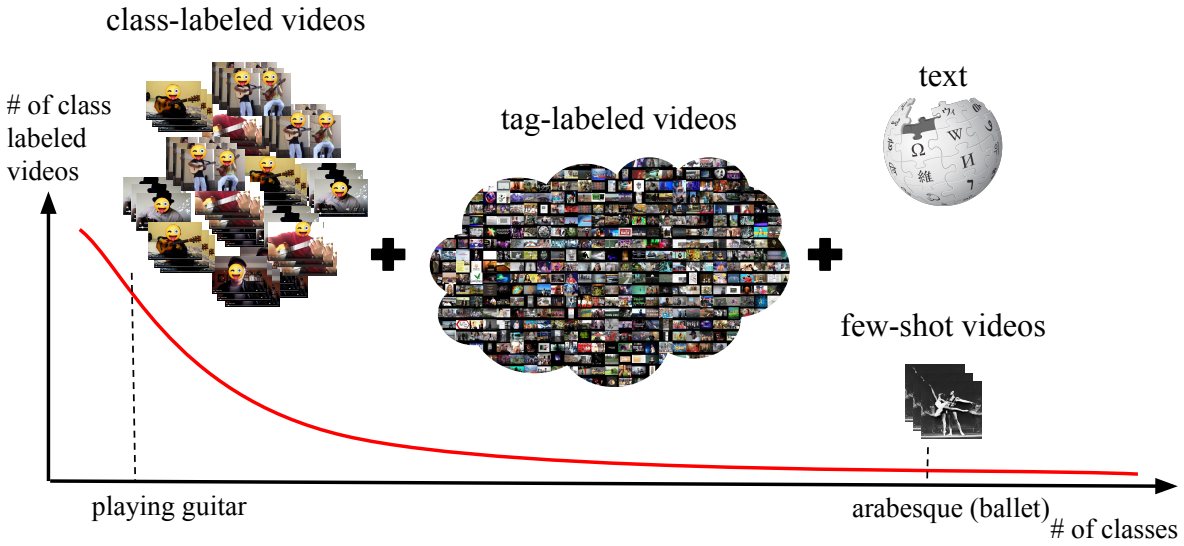

Figure 8.1: Leveraging the lack of class-labeled videos (time-consuming to obtain) with tag-labeled videos, few-shot videos and text, our 3D CNN saturates existing benchmarks and enables the more challenging generalized few-shot multi-way video classification task.

8.1 INTRODUCTION

In the video domain annotating data is very time-consuming due to the additional time dimension. A lack of labelled training data is more prominent in fine-grained scenarios such as action recognition. For some fine-grained action classes at the "tail" of the skewed long-tail distribution (see Figure 8.1 for an illustration), e.g., 'arabesque in ballet", collecting enough training videos is even not possible It is thus of great importance to investigate how to learn to classify videos in the limited labeled training data regime. Visual recognition methods that operate in the few-shot learning setting aim to generalize a classifier trained on known classes (often referred to as base classes) with enough training data to unknown (novel) classes with only a few labelled training examples. While considerable attention has been devoted to the scenario of few-shot image classification (Vinyals *et al.*, 2016; Qi *et al.*, 2018; Ravi and Larochelle, 2016; Chen *et al.*, 2019), few-shot video classification is relatively unexplored.

Existing few-shot video classification approaches (Zhu and Yang, 2018; Cao *et al.*, 2019) are mostly based on frame-level features extracted from a 2D CNN, which essentially ignores the important temporal information. Although additional temporal modules have been added at the top of a pre-trained 2D CNN, necessary temporal cues may be lost when temporal information is learned on top of static image features. We argue that under-representing temporal cues may negatively impact the robustness of the classifier. In fact, in the few-shot scenario it may be

risky for the model to rely exclusively on appearance and context cues extrapolated from the few examples available. In order to make temporal information available we propose to represent the videos by means of a 3D CNN.

While obtaining labelled videos for target classes is time-consuming and challenging, there are many weakly-labelled videos available on the internet, e.g. there are 400,000 tag-labelled videos in the YFCC100M (Thomee *et al.*, 2015) dataset. Our second goal is thus to leverage such tag-labelled videos (Figure 8.1) to alleviate the lack of training data for our few-shot video models.

Existing experimental settings for few-shot video classification (Zhu and Yang, 2018; Cao *et al.*, 2019) are limited. Searching for the label among 5 novel classes, i.e. classes with few-shot videos, in each testing episode is restrictive. Moreover, restricting the search space to novel classes at test time, i.e. test set consists of only videos from novel classes and models only have to predict novel classes, and ignoring the base classes is unrealistic because in real-world applications test videos are expected to belong to any class.

In this work, our goal is to push the progress of few-shot video classification in three ways: 1) To learn the temporal information, we revisit spatiotemporal CNNs in the few-shot video classification regime. We develop a 3D CNN baseline that maintains significant temporal information within short clips; 2) We propose to retrieve relevant tag-labeled videos from a large video dataset, i.e. YFCC100M, to circumvent the need for class-labeled videos of novel classes; 3) We extend current few-shot video classification evaluation by introducing two challenging experimental settings. In generalized few-shot video classification task, the search space has no restriction in terms of classes. In few-shot video classification with more ways, the search space goes beyond five towards all classes. Our extensive experimental results demonstrate that on existing settings spatiotemporal CNNs outperform the state-of-the-art by a largin margin, and on our proposed settings weakly-labeled videos retrieved using tags successfully tackles both of our new few-shot video classification tasks.

8.2 RELATED WORK

Low-shot learning setup. The low-shot image classification (Mensink *et al.*, 2012; Ravi and Larochelle, 2016; Hariharan and Girshick, 2017) setting uses a large-scale fully labelled dataset for pre-training a DNN and a low-shot dataset with a small number of examples from a disjoint set of classes. The terminology "k-shot n-way classification" means that in the low-shot dataset there are n distinct classes and k examples per class for training. Evaluating with few examples (k small) is bound to be noisy. Therefore, the k training examples are often sampled several times and accuracy results are averaged (Hariharan and Girshick, 2017; Douze *et al.*, 2018). Many authors focus on cases where the number of classes n is small as well, which amplifies the measurement noise. For that case (Ravi and Larochelle, 2016) introduce the notion of "episodes". One episode is one sampling of n classes and k examples

per class.

It is feasible to use distinct datasets for pre-training and low-shot evaluation. However, to avoid dataset bias (Torralba *et al.*, 2011) it is easier to split a large supervised dataset into a set of "base" classes and a set of "novel" classes. The evaluation is most often performed only on novel classes, except (Hariharan and Girshick, 2017; Xian *et al.*, 2019c; Schoenfeld *et al.*, 2019) who evaluate on the combination of base+novel classes.

Recently, a low-shot video classification setup has been proposed (Zhu and Yang, 2018; Dwivedi *et al.*, 2019). They use the same type of decomposition of the dataset as (Ravi and Larochelle, 2016), with learning episodes and random sampling of low-shot classes. In this work, we follow and extend the evaluation protocol of (Zhu and Yang, 2018).

Tackling low-shot learning. The simplest low-shot learning approach is to extract embeddings from the images using the pre-trained trunk and train a linear classifier (Akata *et al.*, 2015a) or logistic regression (Hariharan and Girshick, 2017) on top using the k training available examples. Another approach is to cast low-shot learning as a similarity search problem (Wang *et al.*, 2019b). The "inprinting" approach (Qi *et al.*, 2018), consists in building a linear classifier from the embeddings of training examples, then fine-tune it. It also belongs to this family, since it is equivalent to doing class-mean similarity search with a cosine distance. As a complementary approach, (Joulin *et al.*, 2016b) has looked into exploiting noisy labels to aid classification. By leveraging tags of 100M images from the YFCC100M dataset (Thomee *et al.*, 2015), they show improvements over Imagenet-pretraining. In this work, we use videos from YFCC100M retrieved by tags to augment and improve training of our classifier.

In a meta-learning setup, the the low-shot classifier is assumed to have hyper-parameters or parameters that must be adjusted before training. Thus, there is a preliminary meta-learning step that consists in training those parameters on simulated episodes sampled from the main training data. Matching networks (Vinyals *et al.*, 2016) "meta-learns" an LSTM that maps the low-shot training examples into a classifier. Feature hallucination (Wang *et al.*, 2018c) meta-learns how to generate additional training data for novel classes, directly in the feature space. In MAML (Finn *et al.*, 2017), the embedding classifier is meta-learned to adapt quickly and without overfitting to fine-tuning.

Recent works (Chen *et al.*, 2019; Wang *et al.*, 2019b) suggest that state-of-the-art performance can be obtained by methods that do not need meta learning. In particular, (Chen *et al.*, 2019) show that meta-learning methods are less useful when the image descriptors are expressive enough, which is the case when they are from high-capacity networks trained on large datasets. Therefore, we focus on techniques that do not require a meta-learning stage.

Deep descriptors for videos. Moving from hand-designed descriptors (Dollár *et al.*, 2005; Laptev, 2005; Sadanand and Corso, 2012; Wang and Schmid, 2013) to learned deep-network based descriptors (Feichtenhofer *et al.*, 2016a,b; Karpathy *et al.*, 2014; Simonyan and Zisserman, 2014a; Wang *et al.*, 2016; Tran *et al.*, 2015) has been

enabled by labeled large-scale datasets (Kay *et al.*, 2017; Karpathy *et al.*, 2014), and parallel computing hardware. Deep descriptors are either based on 2D-CNN models operating on a frame-by-frame basis with temporal aggregation (Girdhar *et al.*, 2017; Yue-Hei Ng *et al.*, 2015), or more commonly 3D-CNN models operating on sequential sequences of images we refer to as video-clips (Tran *et al.*, 2015, 2018). Recently, ever-more-powerful descriptors have been developed leveraging two-stream architectures using additional modalities (Feichtenhofer *et al.*, 2016b; Simonyan and Zisserman, 2014a), factorized 3D convolutions (Tran *et al.*, 2018, 2019), or multi-scale approaches (Feichtenhofer *et al.*, 2019). While most of these descriptors are trained in a fully supervised way, advances in learning deep descriptors in either weakly-supervised (Yalniz *et al.*, 2019; Ghadiyaram *et al.*, 2019; Mahajan *et al.*, 2018) or self-supervised fashion have been explored as well (Korbar *et al.*, 2018; Owens and Efros, 2018).

8.3 R-3DFSV APPROACH

In the few-shot learning setting (Zhu and Yang, 2018), classes are split into two disjoint label sets, i.e., base classes (denoted as $\mathcal{C}_b$) that have a large number of training examples, and novel classes (denoted as $\mathcal{C}_n$) that have only a small set of training examples. Let $\mathcal{X}_b$ denote the training videos with labels from the base classes and $\mathcal{X}_n$ be the training videos with labels from the novel classes ($|\mathcal{X}_b| \gg |\mathcal{X}_n|$). Given the training data $\mathcal{X}_b$ and $\mathcal{X}_n$, the goal of the conventional few-shot video classification task (FSV) (Zhu and Yang, 2018; Cao *et al.*, 2019) is to learn a classifier which searches for the labels among novel classes at test time. As the test-time search space is restricted to novel classes, the FSV setting is unrealistic. Thus, in this chapter, we additionally study the generalized few-shot video classification (GFSV) which allows videos at test time to belong to any base or novel class.

8.3.1 3D CNN for FSV (3DFSV)

In this section, we introduce our spatiotemporal CNN baseline for few-shot video classification (3DFSV). Our approach in Figure 8.2 consists of 1) a representation learning stage which trains a spatiotemporal CNN on the base classes, 2) a few-shot learning stage that trains a linear classifier for novel classes with few labelled videos, and 3) a testing stage which evaluates the model on unseen test videos. The details of each of these stages are given below.

Representation learning. Our model uses a spatiotemporal CNN (Tran *et al.*, 2018) $\phi : \mathbb{R}^{F\times 3\times H\times W} \rightarrow \mathbb{R}^{d_v}$, encoding a short, fixed-length video clip of F RGB frames with spatial resolution $H \times W$ to a feature vector in the d_v-dimensional Euclidean space. On top of the feature extractor ϕ, we define a linear classifier $f(\bullet; W_b)$ parameterized by a weight matrix $W_b \in \mathbb{R}^{d_v \times |\mathcal{C}_b|}$, producing a probability distribution over the base classes. The objective is to jointly learn the network ϕ and the classifier W_b by minimizing the cross-entropy classification loss on video clips randomly

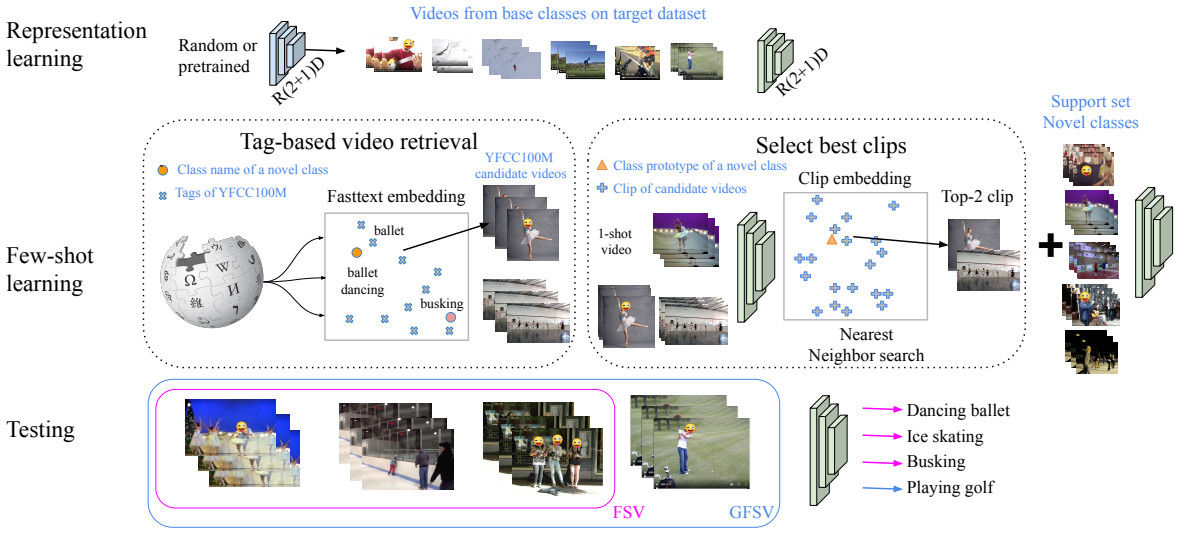

Figure 8.2: Our approach is composed of three steps: representation learning, few-shot learning and testing. In representation learning, we train a R(2+1)D from the random initialization or Sports1M-pretrained model on the base classes of our target dataset. In few-shot learning, given few-shot support videos from novel classes, we first retrieve a list of candidate videos for each class from YFCC100M (Thomee *et al.*, 2015) using their tags, followed by selecting the best matching short clips from the retrieved videos using visual features. Those clips serve as additional training examples to learn classifiers that generalize to novel classes at test time.

sampled from training videos $\mathcal{X}_b$ of base classes. More specifically, given a training video $\mathbf{x} \in \mathcal{X}_b$ with a label $\mathbf{y} \in \mathcal{C}_b$, the loss for a video clip $x_i \in \mathbb{R}^{F\times 3\times H\times W}$ sampled from video x is defined as,

$$\mathcal{L}(\mathbf{x}_i) = -\log \sigma(W_b^T \phi(x_i))_y \tag{8.1}$$

where σ denotes the softmax function that produces a probability distribution and $\sigma(\bullet)_y$ is the probability at class y. Following (Chen *et al.*, 2019), we do not do meta-learning, so we can use all the base classes as a whole to learn the network ϕ.

Few-shot learning. This stage aims to adapt the learned network ϕ to recognize novel classes $\mathcal{C}_n$ with a few training videos $\mathcal{X}_n$. To reduce overfitting, we fix the network ϕ and learn a linear classifier $f(\bullet, W_n)$ by minimizing the cross-entropy loss on video clips randomly sampled from videos in $\mathcal{X}_n$, where $W_n \in \mathbb{R}^{d_v\times|\mathcal{C}_n|}$ is the weight matrix of the linear classifier. Similarly, we define the loss for a video clip $\mathbf{x}_i$ sampled from $\mathbf{x} \in \mathcal{X}_n$ with a label $\mathbf{y}$ as

$$\mathcal{L}(\mathbf{x}_i) = -\log \sigma(W_n^T \phi(x_i))_y \tag{8.2}$$

Testing. The spatiotemporal CNN operates on fixed-length video *clips* of F RGB frames and the classifiers make clip-level predictions. At test time, the model must

predict the label of a test video $\mathbf{x} \in \mathbb{R}^{T\times 3\times H\times W}$ with arbitrary time length T. We achieve this by randomly drawing a set L of clips $\{\mathbf{x}_i\}_{i=1}^{L}$ from video $\mathbf{x}$, where $\mathbf{x}_i \in \mathbb{R}^{F\times 3\times H\times W}$. The video-level prediction is then obtained by averaging the prediction scores after the softmax function over those L clips. For few-shot video classification (FSV), this is:

$$\frac{1}{L}\sum_{i=1}^{L} f(\mathbf{x}_i; W_n). \tag{8.3}$$

For generalized few-shot video classification (GFSV), both base and novel classes are taken into account and we concatenate the base class weight W_b learned in the representation stage with the novel class weight W_n learned in the few-shot learning stage:

$$\frac{1}{L}\sum_{i=1}^{L} f(\mathbf{x}_i; [W_b; W_n]). \tag{8.4}$$

8.3.2 Retrieval-enhanced 3DFSV (R-3DFSV)

During few-shot learning, fine-tuning the network ϕ or learning the classifier $f(\bullet; W_n)$ alone is prone to overfitting. Moreover, class-labeled videos to be used for fine-tuning are scarce. Instead, the hypothesis is that leveraging a massive collection of weakly-labeled real-world videos would improve our novel-class classifier. Thus, for each novel class, we propose to retrieve a subset of weakly-labelled videos, associate pseudo-labels to these retrieved videos and use them to expand the training set of novel classes. For efficiency and to reduce the label noise, we adopt the following two-step retrieval approach.

Tag-based video retrieval. The YFCC100M dataset (Thomee *et al.*, 2015) includes around 800K videos collected from Flickr, with a total length of over 8000 hours. Processing a large collection of videos has a high computational demand and a large portion of them are irrelevant to our target classes. Thus, we restrict ourselves to videos with tags related to those of the target class names. Leveraging information orthogonal with the actual video content increases the visual diversity.

Given a video with user tags $\{t_i\}_{i=1}^{S}$ where $t_i \in \mathcal{T}$ is a word or phrase and S is the number of tags, we represent it with an average tag embedding $\frac{1}{S}\sum_{i=1}^{S}\varphi(t_i)$. The tag embedding $\varphi(.) : \mathcal{T} \to \mathbb{R}^{d_t}$ maps each tag to a d_t dimensional embedding space, e.g., Fasttext (Joulin *et al.*, 2017). Similarly, we can represent each class by the text embedding of its class name and then for each novel class c, we compute its cosine similarity to all the video tags and retrieve the N most similar videos according to this distance.

Selecting best clips. The video tag retrieval selects a list of N candidate videos for each novel class. However, those videos are not yet suitable for training because the annotation may be erroneous, which can harm the performance. Besides, some weakly-labelled videos can last as long as an hour. We thus propose to select the

best short clips of F frames from those candidate videos using the few-shot videos of novel classes.

Given a set of few-shot videos $\mathcal{X}_n^c$ from novel class c, we randomly sample L video clips from each video. We then extract features from those clips with the spatiotemporal CNN ϕ and compute the class prototype by averaging over clip features. Similarly, for each retrieved candidate video of novel class c, we also randomly draw L video clips and extract clip features from ϕ. Finally, we perform a nearest neighbour search with cosine distance to find the M best matching clips of the class prototype. This can be formulated as

$$\max_{\mathbf{x}_j} \cos(p_c, \phi(\mathbf{x}_j)) \tag{8.5}$$

where p_c denotes the class prototype of class c, $\mathbf{x}_j$ is the clip belonging to the retrieved weakly-labeled videos. After repeating this process for each novel class, we obtain a collection of pseudo-labeled video clips $\mathcal{X}_p = \{\mathcal{X}_p^c\}_{c=1}^{|C_n|}$ where $\mathcal{X}_p^c$ indicates the best M video clips from YFCC100M for novel class c.

Batch denoising. The retrieved video clips contribute to learning a better novel class classifier $f(\bullet; W_n)$ in the few-shot learning stage by expanding the training set of novel classes from $\mathcal{X}_n$ to $\mathcal{X}_n \bigcup \mathcal{X}_p$. $\mathcal{X}_p$ may inevitably include noisy video clips with wrong labels. During the optimization, we adopt a simple strategy to alleviate the noise: we construct a mini-batch with half video clips from $\mathcal{X}_n$ and another half video clips from $\mathcal{X}_p$ at each iteration. The purpose is to reduce the gradient noise in each mini-batch by enforcing that half of the samples are correct.

8.4 EXPERIMENTS

In this section, we first describe the existing experimental settings and our proposed setting for few-shot video recognition. We then present the results comparing our approaches with the state-of-the-art methods in the existing setting on two datasets, the results of our approach in our proposed settings, model analysis and qualitative results.

8.4.1 Experimental settings

Here we describe the four datasets we use, previous few-shot video classification protocols and our settings.

Datasets. Kinetics (Kay *et al.*, 2017) is a large-scale video classification dataset which covers 400 human action classes including human-object and human-human interactions. Its videos are collected from Youtube and trimmed to include only one action class. The UCF101 (Soomro *et al.*, 2012) dataset is also collected from Youtube videos, consisting of 101 realistic human action classes, with one action label in each video. SomethingV2 (Goyal *et al.*, 2017) is a fine-grained human action recognition dataset, containing 174 action classes, in which each video shows a

	# classes			# videos		
	train	**val**	**test**	**train**	**val**	**test**
Kinetics	64	12	24	6400	1200	2400+2288
UCF101	64	12	24	5891	443	971+1162
SomethingV2	64	12	24	67013	1926	2857+5243

Table 8.1: Statistics of our data splits on Kinetics, UCF101 and SomethingV2 datasets. We follow the train, val, and test class splits of (Zhu and Yang, 2018) and (Cao *et al.*, 2019) on Kinetics and SomethingV2 respectively. In addition, we add test videos (the second number under the second test column) from train classes for GFSV. We also introduce a new data split on UCF101 and for all datasets we propose 5-,10-,15-,24-way (the maximum number of test classes) and 1-,5-shot setting.

human performing a predefined basic action, such as "picking something up" and "pulling something from left to right". We use the second release of the dataset. YFCC100M (Thomee *et al.*, 2015) is the largest publicly available multimedia collection with about 99.2 million images and 800k videos from Flickr. Although none of these videos are annotated with a class label, half of them (400k) have at least one user tag. We use the tag-labeled videos of YFCC100M to improve the few-shot video classification.

Prior setup. The existing practice of (Zhu and Yang, 2018) and (Cao *et al.*, 2019) indicates randomly selecting 100 classes on Kinetics and on SomethingV2 datasets respectively. Those 100 classes are then randomly divided into 64, 12, and 24 non-overlapping classes to construct the meta-training, meta-validation and meta-testing sets. The meta-training and meta-validation sets are used for training models and tuning hyperparameters. In the testing phase of this meta-learning setting (Zhu and Yang, 2018; Cao *et al.*, 2019), each episode simulates a n-way, k-shot classification problem by randomly sampling a support set consisting of k samples from each of the n classes, and a query set consisting of one sample from each of the n classes. While the support set is used to adapt the model to recognize novel classes, the classification accuracy is computed at each episode on the query set and mean top-1 accuracy over 20,000 episodes constitutes the final accuracy.

Proposed setup. The prior experimental setup is limited to $n = 5$ classes in each episode, even though there are 24 novel classes in the test set. As in this setting the performance saturates quickly, we extend it to 10-way, 15-way and 24-way settings. Similarly, the previous meta-learning setup assumes that test videos all come from novel classes. On the other hand, it is important in many real-world scenarios that the classifier does not forget about previously learned classes while learning novel classes. Thus, we propose the more challenging generalized few-shot video classification (GFSV) setting where the model needs to predict both base and novel classes.

To evaluate a n-way k-shot problem in GFSV, in addition to a support and a query

set of novel classes, at each test episode we randomly draw an additional query set of 5 samples from each of the 64 base classes. We do not sample a support set for base classes because base class classifiers have been learned during the representation learning phase. We report the mean top-1 accuracy of both base and novel classes over 500 episodes.

Kinetics, UCF101 and SomethingV2 datasets are used as our few-shot video classification datasets with disjoint sets of train, validation and test classes (see Table 8.1 for details). Here we refer to base classes as train classes. Test classes include the classes we sample novel classes from in each testing episode. For Kinetics and SomethingV2, we follow the splits proposed by (Zhu and Yang, 2018) and (Cao *et al.*, 2019) respectively for a fair comparison. It is worth noting that 3 out of 24 test classes in Kinetics appear in Sports1M, which is used for pretraining our 3D ConvNet. But the performance drop is negligible if we replace those 3 classes with other 3 random kinetics classes that are not present in Sports1M (more details can be found in the supplementary material). Following the same convention, we randomly select 64, 12 and 24 non-overlapping classes as train, validation and test classes from UCF101 dataset, which is widely used for video action recognition. We ensure that in our splits the novel classes do not overlap with the classes of Sports1M. For the GFSV setting, in each dataset the test set includes samples from base classes coming from the validation split of the original dataset.

Implementation details. Unless otherwise stated our backbone is a 34-layer R(2+1)D (Tran *et al.*, 2018) pretrained on Sports1M (Karpathy *et al.*, 2014) which takes as input video clips consisting of $F = 16$ RGB frames with spatial resolution of $H = 112 \times W = 112$. We extract clip features from the $d_v = 512$ dimensional top pooling units of the R(2+1)D.

In the representation learning stage, we fine-tune the R(2+1)D with a constant learning rate 0.001 on all datasets and stop training when the validation accuracy of base classes saturates. We perform standard spatial data augmentation including random cropping and horizontal flipping. We also apply temporal data augmentation by randomly drawing 8 clips from a video in one epoch. In the few-shot learning stage, the same data augmentation is applied and the novel class classifier is learned with a constant learning rate 0.01 for 10 epochs on all the datasets. At test time, we randomly draw $L = 10$ clips from each video and average their predictions for a video-level prediction.

As for the retrieval approach, we use the 400 dimensional ($d_t = 400$) fasttext (Joulin *et al.*, 2016a) embedding trained with GoogleNews. We first retrieve $N = 20$ candidate videos for each class with video tag retrieval and then select $M = 5$ best clips among those videos with visual similarities.

8.4.2 Comparing with the state-of-the-art

In this section, we compare our model with the state-of-the-art in existing evaluation settings which mainly consider 1-shot, 5-way and 5-shot, 5-way problems and

	Kinetics		SomethingV2	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
CMN (Zhu and Yang, 2018)	60.5	78.9	-	-
CMN++ (Cao *et al.*, 2019)	65.4	78.8	34.4	43.8
TAM (Cao *et al.*, 2019)	73.0	85.8	42.8	52.3
3DFSV (ours, scratch)	48.9	67.8	57.9	75.0
3DFSV (ours, pretrained)	92.5	**97.8**	**59.1**	**80.1**
R-3DFSV (ours, pretrained)	**95.3**	**97.8**	-	-

Table 8.2: Comparing with the state-of-the-art few-shot video classification methods. We report top-1 accuracy on the novel classes of Kinetics and SomethingV2 for 1-shot and 5-shot tasks (both in 5-way). 3DFSV (ours, scratch): our R(2+1)D is trained from scratch; 3DFSV (ours, pretrained): our model is trained from the Sports1M-pretrained R(2+1)D. R-3DFSV (ours, pretrained): our model with retrieved videos, trained from the Sports1M-pretrained R(2+1)D.

evaluate only on novel classes, i.e., FSV. The baselines CMN (Zhu and Yang, 2018) and TAM (Cao *et al.*, 2019) are considered as the state-of-the-art in few-shot video classification. CMN (Zhu and Yang, 2018) proposes a multi-saliency embedding function to extract video descriptor, and few-shot classification is then done by the compound memory network (Kaiser *et al.*, 2017). TAM (Cao *et al.*, 2019) proposes to leverage the long-range temporal ordering information in video data through temporal alignment. They additionally build a stronger CMN, namely CMN++, by using the few-shot learning practices from (Chen *et al.*, 2019). We use their reported numbers for fair comparison. The results are shown in Table 8.2. As the code from CMN (Zhu and Yang, 2018) and TAM (Cao *et al.*, 2019) is not available at the time of submission we do not include UCF101 results.

On Kinetics, we observe that our 3DFSV (pretrain) approach, i.e. without retrieval, outperforms the previous best results by over 19% in 1-shot case (73.0% of TAM vs 92.5% of ours), and by 12% in 5-shot case (85.8.0% of TAM vs 97.8% of ours). On SomethingV2 dataset, we would like to first highlight that our 3DFSV (scratch) significantly improves over TAM by 15.1% in 1-shot (42.8% of TAM vs 57.9% of ours) and by surprisingly 22.7% in 5-shot (52.3% of TAM vs 75.0% of ours). This is encouraging because the 2D CNN backbone of TAM is pretrained on ImageNet, while our R(2+1)D backbone is trained from random initialization.

Our 3DFSV (pretrain) yields further improvement after using the Sports1M-pretrained R(2+1)D. We observe that the effect of the Sports1M-pretrained model on SomethingV2 is not as significant as on Kinetics because there is a large domain gap between Sports1M to SomethingV2 datasets. Those results show that a simple linear classifier on top of a pretrained 3D CNN, e.g. R(2+1)D (Tran *et al.*, 2018), performs better than sophisticated methods with a pretrained 2D ConvNet as a backbone.

Although as shown in C3D (Tran *et al.*, 2015), I3D (Carreira and Zisserman, 2017), R(2+1)D (Tran *et al.*, 2018), spatiotemporal CNNs have an edge over 2D spatial

ConvNet (He *et al.*, 2016) in the fully supervised video classification with enough annotated training data, we are the first to apply R(2+1)D in the few-shot video classification with limited labeled data. It is worth noting that our R(2+1)D is pretrained on the Sports1M while the 2D ResNet backbone of CMN (Zhu and Yang, 2018) and TAM (Cao *et al.*, 2019) is pretrained on ImageNet. A direct comparison between 3D CNNs and 2D CNNs is hard because they are designed for different input data. While it is standard to use an ImageNet-pretrained 2D CNN in image domains, it is common to apply a Sports1M-pretrained 3D CNN in video domains. One of our goals is to establish a strong few-shot video classification baseline with 3D CNNs. Intuitively, the temporal cue of the video is better preserved when clips are processed directly by a spatiotemporal CNN as opposed to processing them as images via a 2D ConvNet. Indeed, even though we train our 3DFSV from the random initialization on SomethingV2 dataset which requires strong temporal information, our results still remain promising. This confirms the importance of 3D CNNs for few-shot video classification.

Our R-3DFSV (pretrain) approach, i.e. with retrieved weakly-labeled video clips, lead to further improvements in 1-shot case (3DFSV (pretrain) 92.5% vs R-3DFSV (pretrain) 95.3) on Kinetics dataset. This implies that weakly-labeled videos retrieved from the YFCC100M dataset include discriminative cues for Kinetics tasks. In 5-shot, our R-3DFSV (pretrain) approach achieves similar performance as our 3DFSV (pretrain) approach however with an 97.8% this task is almost saturated. We do not retrieve any weakly-labeled videos for the SomethingV2 dataset because it is a fine-grained dataset of basic actions and it is unlikely that YFCC100M includes any relevant video for that dataset. As a summary, although 5-way classification setting is still challenging to those methods with 2D ConvNet backbone, the results saturate with the stronger spatiotemporal CNN backbone.

8.4.3 Increasing the number of classes in FSV

Although prior works evaluated few-shot video classification on 5-way, i.e. the number of novel classes at test time is 5, our 5-way results are already saturated. Hence, in this section, we go beyond 5-way classification and extensively evaluate our approach in the more challenging, i.e., 10-way, 15-way and 24-way few-shot video classification (FSV) setting. Note that from every class we use one sample per class during training, i.e. one-shot video classification.

As shown in Figure 8.3, our R-3DFSV method exceeds 95% accuracy both in Kinetics and UCF101 datasets for 5-way classification. With the increasing number of novel classes, e.g. 10, 15 and 24, as expected, the performance degrades. Note that, our R-3DFSV approach with retrieval consistently outperforms our 3DFSV approach without retrieval and the more challenging the task becomes, e.g. from 5-way to 24-way, the larger improvement retrieval approach can achieve on Kinetics, i.e. our retrieval-based method is better than our baseline method by 2.8% in 5-way (ours 3DFSV 92.5% vs our R-3DFSV 95.3%) and the gap becomes 4.3% in 24-way (our 3DFSV 82.0% vs our R-3DFSV 86.3%).

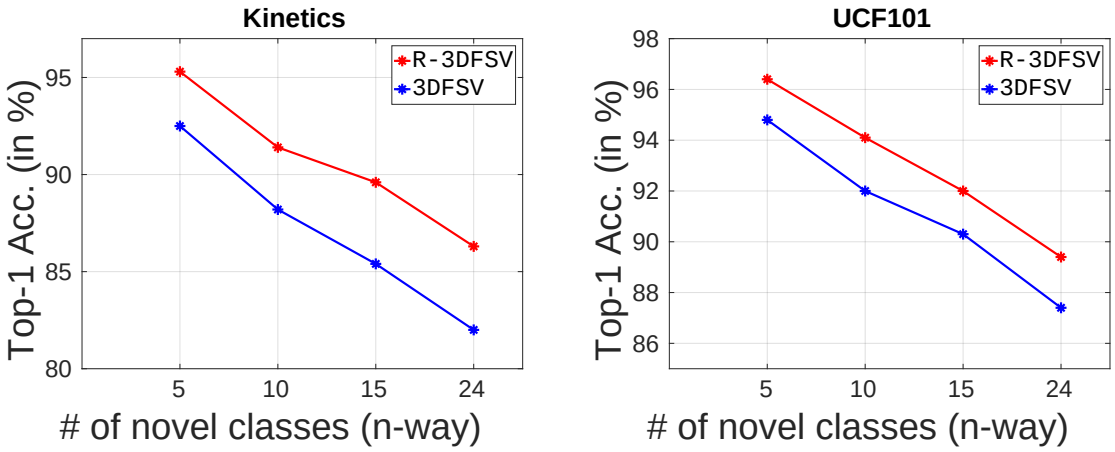

Figure 8.3: Results of 3DFSV and R-3DFSV on both Kinetics and UCF101 in the one-shot video classification setting (FSV). In this experiment we go beyond the classical 5-way classification setting. We use 5, 10, 15 and 24 (all) of the novel classes in each testing episode. We report the top-1 accuracy of novel classes.

The trend with a decreasing accuracy by going from 5-way to 24-way indicates that the more realistic task on few-shot video classification has not yet been solved even with a spatiotemporal CNN. We hope that these results will encourage more progress in this challenging setting of many-way few-shot video classification setting.

8.4.4 Evaluating base and novel classes in GFSV

The FSV setting has a strong assumption that test videos all come from novel classes. In contrast to the FSV, GFSV is more realistic and requires models to predict both base and novel classes in each testing episode. In other words, 64 base classes become distracting classes when predicting novel classes which makes the task more challenging. Intuitively, distinguishing novel and base classes is a challenging task because there are severe imbalance issues between the base classes with a large number of training examples and the novel classes with only few-shot examples. In this section, we evaluate our methods in the more realistic and challenging generalized few-shot video classification (GFSV) setting.

In Table 8.3, on the Kinetics dataset, we observe a large performance gap between base and novel classes in both 1-shot and 5-shot cases, i.e., 3DFSV only achieves 7.5% on novel classes vs 88.7% on base classes. The reason is that predictions of novel classes are dominated by the base classes. Interestingly, our R-3DFSV improves 3DFSV on novel classes in both 1-shot and 5-shot cases, e.g., 7.5% of 3DFSV vs 13.7% of R-3DFSV in 1-shot. A similar trend can be observed on the UCF101 dataset. Those results demonstrate that our retrieval-based approach can alleviate the imbalance issues to some extent. At the same time, we find that generalized few-shot video classification (GFSV) setting, e.g. not restricting the test time search space only to novel classes but considering all of the classes even though base classes are

		Kinetics		UCF101	
	Method	novel	base	novel	base
1-shot	3DFSV	7.5	88.7	3.5	97.1
	R-3DFSV	13.7	88.7	4.9	97.1
5-shot	3DFSV	20.5	88.7	10.1	97.1
	R-3DFSV	22.3	88.7	10.4	97.1

Table 8.3: Generalized few-shot video classification results on Kinetics and UCF101 in 5-way tasks. We report top-1 accuracy on both base and novel classes.

PR	**SS**	**RL**	**VR**	**BD**	**BC**	**Acc**
✓						27.1
		✓				48.9
	✓	✓				51.9
✓		✓				92.5
✓		✓	✓			91.4
✓		✓	✓	✓		93.2
✓		✓	✓		✓	95.3

Table 8.4: Ablation study on 5-way 1-shot video classification task on the meta-test set of Kinetics. **PR**: pretrain R(2+1)D on Sports1M; **SS**: self-supervised model of AVTS (Korbar *et al.*, 2018); **RL**: representation learning on base classes; **VR**: retrieve unlabeled videos with tags (Thomee *et al.*, 2015); **BD**: batch denoising. **BC**: best clip selection.

distracting, is still a challenging task and hope that this setting will attract interest of a wider community for future research.

8.4.5 Ablation study and retrieved clips

In this section, we perform an ablation study to understand the importance of each component of our approach. After the ablation study, we evaluate the importance of the number of retrieved clips to the few-shot video classification (FSV) performance.

Ablation study. We ablate our model in the 1-shot, 5-way video classification task on Kinetics dataset with respect to six critical parts including pretraining R(2+1)D on Sports1M (**PR**), self-supervised model of (Korbar *et al.*, 2018) as the backbone (**SS**), representation learning on base classes (**RL**), video retrieval with tags (**VR**), batch denoising (**BD**) and best clip selection (**BC**). Table 8.4 shows the results.

We start from a model with only a few-shot learning stage on novel classes. If a **PR** component is added to the model (first result row in Table (8.4), the newly-obtained model can achieve 27.1% accuracy which is only slightly better than random guessing performance (20%). It demonstrates that a pretrained 3D CNN alone is

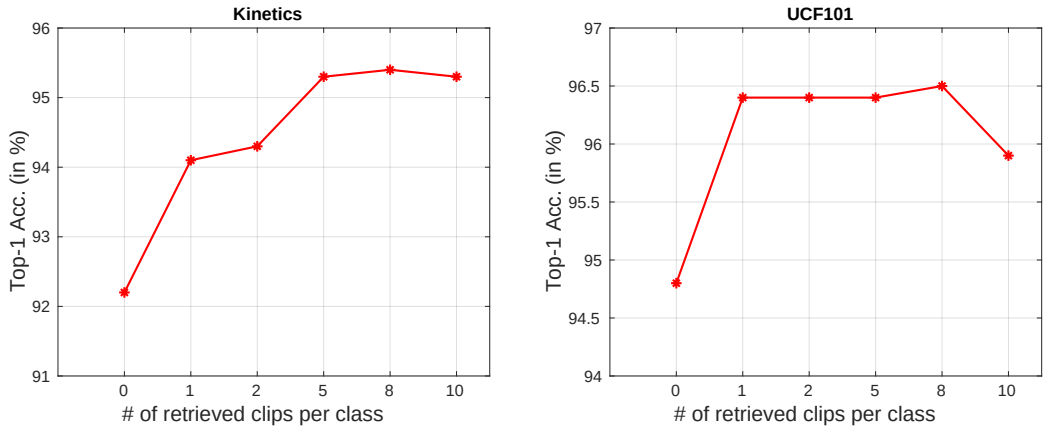

Figure 8.4: The effect of increasing the number of retrieved clips, **left**: on Kinetics, **right**: on UCF101. Both experiments are conducted on the one-shot, five-way classification task, reporting top-1 accuracy in the few-shot video classification (FSV) setting.

not sufficient for a good performance. Besides, it also indicates that there exists a domain shift between the pretraining dataset, i.e. Sports1M, and our target Kinetics dataset.

Adding **RL** component to the model (the second result row) means to train representation on base classes from scratch, which results in a worse accuracy of 48.9% compared to our full model. The primary reason for worse results is that optimizing the massive number of parameters of R(2+1)D is difficult on a train set consisting of only 6400 videos. Interestingly, if we adopt the self-supervised pretrained 3D CNN (MC3 pretrained on Kinetics without using any label) of (Korbar *et al.*, 2018), i.e., **SS**, we immediate get 3.0% performance gains (the third result row) over training from random initialization. Adding both **PR** and **RL** components (the fourth row) obtains an accuracy of 92.5 which significantly boosts adding **PR** and **RL** components alone.

Next, we study two critical components proposed in our retrieval approach. Comparing to our approach without retrieval (the fourth row), directly appending retrieved videos from YFCC100M (**VR**) to the few-shot training set of novel classes (the fifth result row) leads to 0.9% performance drop, while performing the batch denoising (the sixth row) in addition to **VR** obtains 0.7% gain. This implies that noisy labels from retrieved videos may hurt the performance but our batch denoising technique handles the noise well. Finally, adding the best clip selection (**BC**, the last row) after **VR** and **BD** gets a big boost of 2.8% accuracy. In summary, those ablation studies demonstrate the effectiveness of the six different critical parts in our approach.

Influence of the number of retrieved clips. Intuitively, when the number of retrieved clips increases, the retrieved videos become more diverse, but at the same time, the risk of obtaining negative videos becomes higher. We show the effectiveness

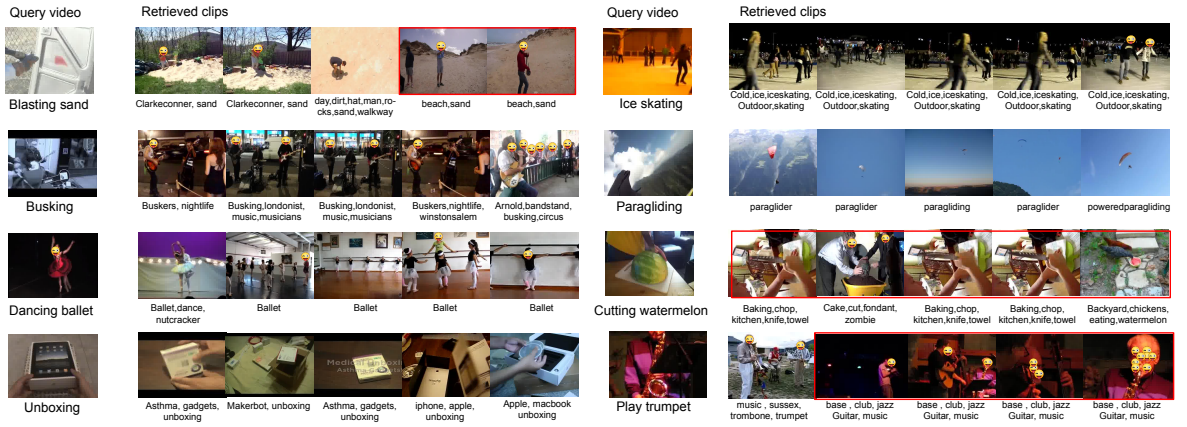

Figure 8.5: Top-5 retrieved video clips from YFCC100M for 8 novel classes on Kinetics. The left column is the class name with its one-shot query video and the right column shows the retrieved 16-frame video clips (middle frame is visualized) together with their users tags. Negative retrievals are marked in red.

of our R-3DFSV with the increasing number of retrieved clips in Figure 8.4.

On the Kinetics dataset (left of Figure 8.4), without retrieving any videos, the performance is 92.5%. As we increase the number of retrieved video clips for each novel class, the performance keeps improving and saturates at retrieving 8 clips per class, reaching an accuracy of 95.4%. On the UCF101 dataset (right of Figure 8.4), retrieving 1 clip gives us 1.6% gain. Retrieving more clips does not further improve the results, indicating more negative videos are retrieved. On the other hand, our batch denoising strategy is able to tolerate the noise to some extent. We observe a slight performance drop at retrieving 10 clips because the noise level becomes too high, i.e. there are 10 times more noisy labels than clean labels.

8.4.6 Qualitative results

In Figure 8.5, we visualize the top-5 video clips we retrieve from YFCC100M dataset with video tag retrieval followed by the best clips selection. Here we only show 8 novel classes of Kinetics dataset due to the space limitation and visualization of other classes are in supplementary.

We observe that the retrieved video clips of some classes are of high quality, meaning that those videos truly reveal the target novel classes. For instance, retrieved clips of class "Busking" are all correct because user tags of those videos consist of words like "buskers", "busking" that are close to the class name, and the best clip selection can effectively filter out the irrelevant clips. It is intuitive those clips can potentially help to learn better novel class classifiers by supplementing the limited training videos.

Failure cases are also common. For example, videos from the class "Cutting

watermelon" do not retrieve any positive videos. The reasons can be that there are no user tags of cutting watermelon or our tag embeddings are not good enough. Those negative videos might hurt the performance if we treat them equally, which is why the batch denoising is critical to reduce the effect of negative videos.

8.5 CONCLUSION

In this work, we point out that a spatiotemporal CNN trained on a large-scale video dataset saturates existing few-shot video classification benchmarks. Hence, we propose new more challenging experimental settings, namely generalized few-shot video classification (GFSV) and few-shot video classification with more ways than the classical 5-way setting. We further improve spatiotemporal CNNs by leveraging the weakly-labelled videos from YFCC100M using weak-labels such as tags for text-supported and video-based retrieval. Our results show that generalized more-way few-shot video classification is challenging and we encourage future research in this setting.

9 CONCLUSIONS AND FUTURE PERSPECTIVES

Contents

SIGNIFICANT progress has been made across various computer vision tasks in recent years. Deep neural networks have achieved great breakthrough in reliable object recognition for up to 1000 object categories (He *et al.*, 2016), in widely-applicable activity recognition (Carreira and Zisserman, 2017) and in robust semantic image segmentation (Chen *et al.*, 2018) for autonomous driving. Despite the success, training a deep neural network always requires a massive amount of labeled instances. In real world applications, labeled instances are often expensive and difficult to obtain because annotating data requires expert knowledge. Training a standard deep neural network on a small training set will lead to overfitting. It is thus of great importance to study the problems of learning with limited labeled data. This thesis aims to push the progress of the field by exploring how to transfer knowledge from known classes with enough labeled instances to novel classes with only limited labeled instances. More specifically, we focus on the following three directions, (1) zero-shot image classification where novel object classes have zero training examples, (2) few-shot image classification where each novel object class has only a few training examples, and (3) zero-shot and few-shot learning for semantic image segmentation and video action recognition. After a summary of the thesis with respect to the three directions in the following, we discuss our contributions and future perspectives.

First, we examined zero-shot image classification. The goal of the task is to recognize novel object classes without observing any image instances of them by transferring knowledge from known to novel classes. In order to capture the complex correlation between image and semantic embedding spaces, we propose a piece-wise linear label embedding approach called LatEm that learns multiple linear transformation from image embedding space to the semantic embedding space. As there is no agreed upon zero-shot image classification benchmark, we first define a new benchmark by unifying both the evaluation protocols and data splits of publicly available datasets. We re-evaluate a significant number of methods on our

benchmark. Our analysis shows the status of the field and advocates to study the realistic generalized zero-shot learning problem where both known and novel classes are predicted during the test phase. To tackle the extreme data imbalance issue in generalized zero-shot learning, we introduce a feature generation framework, namely f-CLSWGAN, that synthesizes visual features for novel classes. We empirically show that f-CLSWGAN is effective to balance the base and novel class performance and the generated features can be applied to any zero-shot learning methods. Additionally, we extend f-CLSWGAN to a stronger version called f-VAEGAN-D2, which combines VAE and GANs for a better generative model and can learn from unlabeled data as well.

The second direction of this thesis is concerned with few-shot image classification. The goal of the task is to recognize novel object classes after observing only a few instances of them. While human beings naturally have such ability, deep neural networks are difficult to be trained on a small training set due to the high risk of overfitting. While most of few-shot learning methods only rely on images of base class for knowledge transfer, we argue that semantic embeddings e.g., attributes, word embeddings and class hierarchy, provide complementary information that would benefit novel classes. Therefore, we extend our zero-shot learning approaches i.e., LatEm and f-VAEGAN-D2, to work in the few-shot learning setting. To this end, we generate few-shot learning splits on public datasets what are widely used for zero-shot learning. We show that our approaches have an edge over the standard linear classifier in few-shot image classification, indicating the benefits of using semantic embeddings. In addition, it is encouraging that our f-VAEGAN-D2 outperforms the state-of-the-art few-shot approaches on challenging large-scale few-shot benchmarks as well. Our experimental results also demonstrate that f-VAEGAN-D2 is able to obtain further improvement from unlabeled data .

The third part of the thesis looks at zero-shot and few-shot learning tasks beyond the image classification. More specifically, we tackle both semantic image segmentation and video action recognition with limited training examples. While most few-shot and zero-shot works are tackling image classification, there are little works on other computer vision tasks. To this end, we introduce the zero-label and few-label semantic segmentation problems and new data splits on public semantic segmentation datasets i.e., COCO-Stuff and Pascal-VOC. The task is to segment novel classes with few or zero instances. Inspired by our previous experience in zero-shot image classification, we develop a novel approach called SPNet, that projects each pixel into the semantic embedding space for knowledge transfer. Our SPNet can be incorporated into any semantic segmentation networks. We empirically show that it achieves decent results in zero-label setting and outperforms the state-of-the-art methods in the few-label setting. In addition, we study the few-shot video classification problem. We found that previous methods focus only on developing complicated few-shot methods but fail to adopt strong video representation that captures better temporal information. Our work shows that video representation with strong temporal modeling is critical for few-shot video classification. Moreover, we propose to leverage weakly-labeled videos from a large-scale video dataset to

expand the few-shot training set, leading to further improvement.

In summary, this thesis defines a new zero-shot image classification benchmark. In order to improve the benchmark performance as well as few-shot image classification, we present a multi-modal learning approach and another two methods that generate synthetic visual features. We further tackle few-shot and zero-shot learning challenges for semantic segmentation and video action classification tasks.

9.1 DISCUSSION OF CONTRIBUTIONS

The goal of this thesis is to develop efficient methods to improve the performance of learning with limited labeled data. To this end, we study zero-shot and few-shot learning problems which aim to learn novel classes with zero or only a few training examples. In the following we will discuss the contributions and steps we made towards these goals and tasks with respect to the individual chapters.

First, we presented a novel latent variable model, Latent Embeddings (LatEm), for learning a nonlinear (piece-wise linear) compatibility function for the task of zero-shot classification in Chapter 3 . LatEm is a multi-modal method, it uses images and class-level side-information either obtained through human annotation or in an unsupervised way from a large text corpus. LatEm incorporates multiple linear compatibility units and allows each image to choose one of them – such choices being the latent variables. We proposed a ranking based objective to learn the model using an efficient and scalable SGD based solver. We empirically validated our model on three challenging benchmark datasets for zero-shot classification of Birds, Dogs and Animals. We improved the state-of-the-art for zero-shot learning using unsupervised class embeddings i.e., word embeddings, on AWA and on two fine-grained datasets (CUB and Stanford Dogs). On AWA, we also improve the accuracy obtained with supervised class embeddings i.e., human-annotated attributes. This demonstrates quantitatively that our method learns a latent structure in the embedding space through multiple compatibility units. We also presented a qualitative analysis of our results and showed that the latent embeddings learned with our method leads to visual consistencies. We proposed a new method for selecting the number of latent variables automatically from the data by pruning. Such pruning based method speeds up the training and leads to models with competitive space-time complexities compared to the cross-validation based method. We further extended our application domain to generalized zero-shot and generalized few-shot learning setting where at training time we assume the availability of either no or a few labeled samples from unseen classes. On the other hand, both at training and test time the search space includes all the class embeddings from seen and unseen classes. As expected, our evaluation on generalized zero-shot learning setting showed a significant loss of accuracy compared to the standard zero-shot learning setting which we analyzed through visualizations and quantitative results. Our evaluation on generalized few-shots setting showed that with as few as two to ten samples from unseen classes, unsupervised class embeddings can outperform the supervised

attributes. Therefore, with increasing number of additional training samples, the difference between different class embeddings are reduced.

Second, in Chapter 4, we evaluated a significant number of state-of-the-art zero-shot learning methods, i.e. (Lampert *et al.*, 2013; Zhang and Saligrama, 2015; Xian *et al.*, 2016; Akata *et al.*, 2015c; Romera-Paredes *et al.*, 2015; Changpinyo *et al.*, 2016; Socher *et al.*, 2013; Norouzi *et al.*, 2014; Frome *et al.*, 2013; Akata *et al.*, 2015a; Kodirov *et al.*, 2017; Verm and Rai, 2017; Ye and Guo, 2017), on several datasets, i.e. SUN, CUB, AWA1, AWA2, aPY and ImageNet, within a unified evaluation protocol both in zero-shot and generalized zero-shot settings. Our evaluation showed that generative models and compatibility learning frameworks have an edge over learning independent object or attribute classifiers and also over other hybrid models for the classic zero-shot learning setting. We observed that unlabeled data of unseen classes can further improve the zero-shot learning results, thus it is not fair to compare transductive learning approaches with inductive ones. We discovered that some standard zero-shot dataset splits may treat feature learning disjoint from the training stage as several test classes are included in the ImageNet1K dataset that is used to train the deep neural networks that act as feature extractor. Therefore, we proposed new dataset splits making sure that none of the test classes in none of the datasets belong to ImageNet1K. Moreover, disjoint training and validation class split is a necessary component of parameter tuning in zero-shot learning setting. In addition, we introduced a new Animal with Attributes (AWA2) dataset. AWA2 inherits the same 50 classes and attributes annotations from the original Animal with Attributes (AWA1) dataset, but consists of different 37,322 images with publicly available redistribution license. Our experimental results showed that the 12 methods that we evaluated perform similarly on AWA2 and AWA1. Moreover, our statistical consistency test indicated that AWA1 and AWA2 are compatible with each other. Finally, including training classes in the search space while evaluating the methods, i.e. generalized zero-shot learning, provides an interesting playground for future research. Although the generalized zero-shot learning accuracy obtained with 13 models compared to their zero-shot learning accuracy is significantly lower, the relative performance comparison of different models remain the same. In summary, our work extensively evaluated the good and bad aspects of zero-shot learning while sanitizing the ugly ones.

Third, in Chapter 5, we propose f-CLSWGAN, a learning framework for feature generation followed by classification, to tackle the generalized zero-shot learning task. Our f-CLSWGAN model adapts the conditional GAN architecture that is frequently used for generating image pixels to generate CNN features. In f-CLSWGAN, we improve WGAN by adding a classification loss on top of the generator, enforcing it to generate features that are better suited for classification. In our experiments, we have shown that generating features of unseen classes allows us to effectively use softmax classifiers for the GZSL task. Our framework is generalizable as it can be integrated to various deep CNN architectures, i.e. GoogleNet and ResNet as a pair of the most widely used architectures. It can also be deployed with various classifiers, e.g. ALE, SJE, DEVISE, LATEM, ESZSL that constitute the state of the

art for ZSL but also the GZSL accuracy improvements obtained with softmax is important as it is a simple classifier that could not be used for GZSL before this work. Moreover, our features can be generated via different sources of class embeddings, e.g. Sentence, Attribute, Word2vec, and applied to different datasets, i.e. CUB, FLO, SUN, AWA being fine and coarse-grained ZSL datasets and ImageNet being a truly large-scale dataset. Finally, based on the success of our framework, we motivated the use of GZSL tasks as an auxiliary method for evaluation of the expressive power of generative models in addition to manual inspection of generated image pixels which is tedious and prone to errors. For instance, WGAN (Gulrajani *et al.*, 2017) has been proposed and accepted as an improvement over GAN (Goodfellow *et al.*, 2014). This claim is supported with evaluations based on manual inspection of the images and the inception score. Our observations in Figure 5.4 and in Figure 5.6 support this and follow the same ordering of the models, i.e. WGAN improves over GAN in ZSL and GZSL tasks. Hence, while not being the primary focus of this chapter, we strongly argue, that ZSL and GZSL are suited well as a testbed for comparing generative models.

Fourth, in Chapter 6 , we develop a transductive feature generating framework that synthesizes CNN image features from a class embedding. Our generated features circumvent the scarceness of the labeled training data issues and allow us to effectively train softmax classifiers. Our framework combines conditional VAE and GAN architectures to obtain a more robust generative model. We further improve VAE-GAN by adding a non-conditional discriminator that handles unlabeled data from unseen classes. The second discriminator learns the manifold of unseen classes and backpropagates the WGAN loss to feature generator such that it generalizes better to generate CNN image features for unseen classes. Our feature generating framework is effective across zero-shot (ZSL), generalized zero-shot (GZSL), few-shot (FSL) and generalized few-shot learning (GFSL) tasks on CUB, FLO, SUN, AWA and large-scale ImageNet datasets. Finally, we show that our generated features are visually interpretable, i.e. the generated images by by inverting features into raw image pixels achieve an impressive level of detail. They are also explainable via language, i.e. visual explanations generated using our features are class-specific.

Fifth, in Chapter 7, we propose SPNet to semantically segment novel classes with no labeled examples or with only a few samples, within the new tasks of zero-label semantic segmentation and few-label semantic segmentation respectively. This model consists of a visual-semantic embedding module that encodes images in the word embedding space and a semantic projection layer that produces class probabilities. Our SPNet is both conceptually and computationally simple but surprisingly effective and end-to-end trainable. We have shown its applicability across zero-shot image classification to zero-label and few-label semantic segmentation tasks on various benchmark datasets.

Finally, in Chapter 8, we point out that a spatiotemporal CNN trained on a large-scale video dataset saturates existing few-shot video classification benchmarks. Hence, we propose new more challenging experimental settings, namely generalized few-shot video classification (GFSV) and few-shot video classification with more

ways than the classical 5-way setting. We further improve spatiotemporal CNNs by leveraging the weakly-labelled videos from YFCC100M using weak-labels such as tags for text-supported and video-based retrieval. Our results show that generalized more-way few-shot video classification is challenging and we encourage future research in this setting.

9.2 FUTURE PERSPECTIVES

The content of this thesis mainly focuses on establishing benchmark and tackling imbalanced issues for few-shot and zero-shot learning in various computer vision applications. Despite the progress we achieved, few-shot and zero-shot learning are still not saturating. In the following we first discuss items of future work with respect to the different directions of the thesis. In the last section we give a broader outlook for the field.

9.2.1 Zero-shot image classification

Most of zero-shot learning methods as well as proposed approaches in this thesis rely on deep representation that is pretrained or finetuned following the standard supervised learning setting. We postulate there exists special image representation that is more efficient for zero-shot learning. In addition, as semantic embeddings play an important role in zero-shot learning, it is promising to explore better unsupervised semantic embeddings rather than annotating attributes. We layout the following directions for future work.

Explainable zero-shot learning. This thesis has been adopting human annotated attributes for several datasets i.e. CUB, AWA and SUN. While decent zero-shot results have been achieved with the attributes, we still lack an explainable approach that tells us how the zero-shot prediction is made. One possible way to improve the visual explainability is by localizing semantic parts i.e., "head of a bird", "beak of a bird", etc. Previous works (e.g. Zhang *et al.*, 2016b, 2014) directly tackle the bird part detection problem by using the part annotation, which is expensive to obtain. In a future work, we are interested in introducing new intermediate layers into a CNN architecture such that bird parts can be localised using only class-level attributes. We believe such representation network will naturally have better interpretibility and potentially lead to better fine-grained zero-shot learning performance due to its better locality.

Improving locality and compositionality of the image representation Zero-shot learning aims to achieve generalization on novel tasks. However, most of existing zero-shot learning works rely on the standard CNNs, which has a different goal of achieving the same task generalization. In a future work, we are interested in exploring special representation learning framework for zero-shot learning. We are inspired by (Sylvain *et al.*, 2019) which points out that

locality and compositionality are the two representation learning principles that attribute to a good performance in zero-shot learning. Local features have been widely used in computer vision for a long history. The traditional hand-crafted features e.g., SIFT (Lowe, 2004), SURF (Bay *et al.*, 2008), extract statistics within local patches in an image and aggregate them to form a global image representation. Similarly, CNNs (LeCun *et al.*, 2015) perform convolution operation on local patches in the images followed by some non-liearity and pooling. By stacking multiple such convolutional layers, CNNs increase its receptive and get more global features. Local features can beneficial to novel task generalization because local information is often shared by many classes. On the contrary, global information is often category-specific and requires a lot of training examples to learn the within-class variations. Another direction is to explore compositionality of the representation. The key insight is that the representation will be able to encode classes more efficiently if representation is compositional of visual primitives. The challenge is that how we define the compositional function and how we learn visual primitives.

Compositional zero-shot learning. Most of the existing zero-shot learning works rely on attribute annotation to achieve the best performance. In real-world applications, attribute annotation is often not available. Compositional zero-shot learning (Purushwalkam *et al.*, 2019) is a special zero-shot learning problem where attribute annotation is not available, but visual concepts are assumed to be composed by an adjective and an object e.g. "red apple" and "green apple". The goal is to predict novel visual concepts that are unseen compositions of existing adjective and objects. Interesting research ideas could be to explore how our feature generation idea can be adapted to this problem and how we learn compositional representation.

Graph convolutional networks (GCN) for large-scale zero-shot image classification. The zero-shot learning performance on the large-scale ImageNet is limited by the weakness of noisy word embeddings. Recently (Wang *et al.*, 2018b) significantly improves the large-scale zero-shot learning performance by adopting a graph CNN (Kipf and Welling, 2017) on the wordnet hierarchy. But (Wang *et al.*, 2018b) simply takes as input the original class hierarchy, ignoring the special tree structure of the wordnet and visual similarities. The GCN used in (Wang *et al.*, 2018b) also has over-smoothing issues. Therefore, we are interesting in exploring a better graph construction method and a new graph convolutional neural network technique for the large-scale zero-shot learning performance.

Learning unsupervised semantic embeddings. It is clear in this thesis that the semantic embeddings play a critical role in zero-shot learning performance. Attributes often achieve the best results but they require expert knowledge to annotate. Unsupervised word embeddings i.e., word2vec (Mikolov *et al.*, 2013b) and glove (Pennington *et al.*, 2014), are easier to obtain but it has a big performance gap behind the attributes. Recently, a new language model

called BERT (Devlin *et al.*, 2018) has created new state-of-the-art on a wide range of NLP tasks. We believe it is promising to enhance the unsupervised embeddings by incorporating BERT (Devlin *et al.*, 2018).

9.2.2 Few-shot image classification

Both zero-shot and few-shot learning share the same goal of novel task generalization. Therefore, we believe technique that work for zero-shot learning can potentially work well in few-shot learning as well. For this reason, it is interesting to investigate image representation with better locality and compositionality for few-shot learning. In addition to that, we would consider the following topics as promising directions.

Cross-domain few-shot learning. Significant improvement has been made in the few-shot learning setting where both base and novel classes belong to the same dataset i.e., Mini-ImageNet and Omniglot. However, in many real-world applications, novel classes are likely from a different domain. For example, if the target novel classes belong to the medical image domains, it is difficult to collect sufficient amount of base class data from the same domain. Therefore, we consider that learning to learn adaptation with limited labeled data would be an important direction for future few-shot learning research. Unlabeled data from novel classes could potentially help to domain adaptation.

Generalized few-shot learning. Majority of few-shot learning methods are evaluated in the meta-learning setup where a new set of classes is sampled from all the novel classes in each episode and the goal is to improve the novel class accuracy over many episodes. However, such evaluation protocol is not realistic because it ignores the base classes. In real-world applications, we are interested in the generalized few-shot learning where the model has to predict both base and novel classes. Similar setting in zero-shot learning has attracted increasing attention, but there are not much few-shot learning works that tackle this problem. We believe it is an important direction as well.

Semi-supervised few-shot learning. While obtaining labeling data is difficult, unlabeled data is often easy to collect. Therefore, it is of great importance to study semi-supervised few-shot learning field where the training set consists of few-shot labeled examples and a large number of unlabeled examples. Previous approaches are limited to adopt the classical semi-supervised learning technics like label propogation or semi-SVM. We are interested in combining a few-shot learning objects on the labeled data with self-supervised learning objectives on unlabeled data. Given the success of recent self-supervised learning approaches (e.g. Chen *et al.*, 2020; He *et al.*, 2019), we believe those technics would benefit few-shot learning.

Meta-learning. Meta-learning or learning to learn, is a popular subfield of few-shot learning. The key insight is to exploit training classes for the purpose

of learning "a meta procedure", e.g., initialization, optimization algorithm, that generalizes well to novel classes. This concept sounds appealing, but we concern the limitation of their evaluation setting. More specifically, most of papers are only evaluated on 5 classes with 1 or 5 samples per class in each episode. Recently, (Triantafillou *et al.*, 2019) proposes a new large-scale meta-dataset that addresses those issues. We think it is interesting to work on meta-learning field on this more realistic benchmark.

Bayesian few-shot learning. Most of few-shot learning approaches produce a single model after learning from only a small amount of training examples. However, there are a lot of uncertainties about the novel classes due to the small training set, resulting ambiguous description of novel classes. It is impossible that a single model could achieve accurate results on those novel classes. We believe that Bayesian learning could address the ambiguity issues by learning a distribution of models for novel classes. Unfortunately, previous Bayesian few-shot approaches (e.g. Gordon *et al.*, 2018; Yoon *et al.*, 2018; Finn *et al.*, 2018) still do not achieve state-of-the-art results on Mini-ImageNet and recent realistic meta-learning benchmark (Triantafillou *et al.*, 2019). It would be important to further push the performance of Bayesian approaches such that they are more appealing in practice.

9.2.3 Zero-shot and few-shot learning beyond image classification

In addition to the image classification, there are many other computer vision applications naturally facing the few-shot learning problems. Here we list a few applications we are interested in.

Learning stronger temporal information for few-shot videos classification. Our approach for few-shot video classification does not capture long-term temporal information, which can be critical for recognizing actions. We are currently working on a project that aims to learn long-term temporal correlation in video through self-attention (Vaswani *et al.*, 2017). Although the self-attention has been well established in the standard setting, it is not trivial on how to extend it to the few-shot learning setting.

Few-shot learning for medical image analysis. Medical image analysis has always been an important field of computer vision research. The tasks for medical images analysis include image segmentation, computer-aided disease diagnosis, and image registration for scanned images from CT, fMRI, and X-ray. CheXNet (Rajpurkar *et al.*, 2017) achieves radiologists-level pneumonia detection performance by learning a deep CNN on a large-scale chest X-ray dataset. However, such large-scale medical image dataset is not always feasible due to the huge cost of collecting medical images. For novel diseases or other medical image tasks, the few-shot learning challenges remain there. We are excited to extend our expertise in few-shot learning to disease diagnosis from medical

images. In particular, we plan to investigate knowledge transfer technics for novel diseases.

Improving zero-label and few-label semantic segmentation. This thesis has made the first step towards the zero-label and few-label semantic segmentation problems. While we have shown that a semantic project layer followed by the cross-entropy loss works well, we believe that exploring better loss functions is likely to lead to big improvements in the predictions. Furthermore, we found that the performance of generalized zero-label semantic segmentation is still unsatisfied, we believe that exploring better semantic embeddings and special normalization technics are promising directions for this issue.

Few-shot 3D computer vision. 3D computer vision is a critical field for virtual reality, robotics and autonomous driving because the real world is obviously in 3D. Typical 3D vision tasks include 3D reconstruction, 3D human body modeling and 3D scene understanding like detection and tracking problems. Although deep learning has achieved big breakthrough in 2D vision , we have not seen the same progress in 3D vision because collecting and processing 3D training data are difficult. We do not have much expertise in 3D vision and it is hard to suggest any good ideas but we are definitely interested in studying it in the near future.

9.2.4 A broader view on the topic

Our long-term goal is to develop machine perception that can generalized well after observing only limited labeled examples of novel tasks. Few-shot and and zero-shot learning are simply two directions towards this goal. From a broader view, topics of learning with limited labeled data include but not limited to self-supervised learning, and long-tailed recognition problem and multi-modal learning.

Semi-supervised and self-supervised learning. Semi-supervised and self-supervised learning are both two practical solutions for learning with limited labeled data. While semi-supervised learning leverages unlabeled data in addition to labeled data, self-supervised learning learns from a completely unlabeled dataset by solving other proxy tasks that make use of the structure of the input data. I am interested in develop an efficient learning algorithm that combines low-shot learning, semi-supervised learning and self-supervised learning.

Long-tailed recognition problem. Real-world datasets inherently follow a long-tail distribution i.e., the number of samples per class is decreasing exponentially. A reliable visual recognition system should perform well on all the classes by balancing the dataset and transferring knowledge from known classes to novel classes. This is a very challenging task because it must handle imbalanced classification and low-shot learning at the same time. I believe developing robust novelty detection algorithms, special sampling methods, and normalization technics to calibrate the prediction are promising directions.

Multi-modal learning. Learning from multiple modalities of data has been shown to the amount of necessary training instances because different modalities often contain complementary information. In fact, human beings learn from multiple sensory modalities i.e., the five classic types of human perception are senses of vision (sight), audition (hearing), tactile stimulation (touch), olfaction (smell), and gustation (taste). While there have been a lot of studies in learning with vision and language, little research has been done in combining those five sensory modalities (or subsets of them). I feel it hold the potential to improve self-supervised learning by predicting the correspondence between two or multiple modalities.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

Z. Akata, M. Malinowski, M. Fritz, and B. Schiele (2016). Multi-Cue Zero-Shot Learning with Strong Supervision, in *CVPR 2016*. Cited on page 28.

Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid (2013). Label embedding for attribute-based classification, in *CVPR 2013*. Cited on pages 4, 17, 18, 52, and 64.

Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid (2015a). Label-Embedding for Image Classification, *IEEE TPAMI*. Cited on pages 27, 28, 29, 30, 31, 32, 36, 48, 55, 56, 57, 59, 60, 61, 63, 67, 68, 69, 71, 72, 75, 77, 84, 92, 115, 121, 128, and 146.

Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele (2015b). Evaluation of output embeddings for fine-grained image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*. Cited on pages 4, 16, 18, 27, 110, 112, and 121.

Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele (2015c). Evaluation of Output Embeddings for Fine-Grained Image Classification, in *CVPR 2015*. Cited on pages 28, 29, 30, 31, 32, 34, 35, 36, 37, 38, 39, 40, 41, 52, 55, 56, 63, 66, 67, 68, 70, 71, 72, 75, 77, 84, 146, 155, and 161.

Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen (2016). Recovering the Missing Link: Predicting Class-Attribute Associations for Unsupervised Zero-Shot Learning, in *CVPR 2016*. Cited on page 18.

R. Arandjelovic and A. Zisserman (2013). All about VLAD, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition 2013*. Cited on page 15.

M. Arjovsky and L. Bottou (2017). Towards principled methods for training generative adversarial networks, *ICLR*. Cited on pages 20, 81, 83, 96, and 97.

M. Arjovsky, S. Chintala, and L. Bottou (2017). Wasserstein gan, *ICML*. Cited on pages 20, 81, 82, and 100.

V. Badrinarayanan, A. Kendall, and R. Cipolla (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *TPAMI*. Cited on page 111.

A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran (2018). Zero-Shot Object Detection, in *ECCV 2018*. Cited on pages 24 and 115.

E. Bart and S. Ullman (2005). Single-example learning of novel classes using representation by similarity, in *BMVC 2005*. Cited on page 28.

R. H. Bartels and G. Stewart (1972). Solution of the matrix equation AX+ XB= C [F4], *Commun. ACM*, vol. 15(9), pp. 820–826. Cited on page 56.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool (2008). Speeded-up robust features (SURF), *Computer vision and image understanding*, vol. 110(3), pp. 346–359. Cited on pages 14 and 149.

A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei (2016). What's the point: Semantic segmentation with point supervision, in *ECCV 2016*. Cited on pages 110 and 111.

A. Bendale and T. E. Boult (2016). Towards Open Set Deep Networks, in *CVPR 2016*. Cited on page 54.

A. Berg, J. Deng, and L. Fei-Fei (). *ILSVRC 2010*, *http://www.image-net.org/challenges/LSVRC/2010/index*. Cited on page 47.

C. M. Bishop (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg. Cited on page 3.

M. Bucher, S. Herbin, and F. Jurie (2016). Improving Semantic Embedding Consistency by Metric Learning for Zero-Shot Classiffication, in *ECCV 2016*. Cited on page 28.

M. Bucher, S. Herbin, and F. Jurie (2017). Generating Visual Representations for Zero-Shot Classification, *ICCV Workshop*. Cited on pages 20, 82, 95, and 98.

M. Bucher, V. Tuan-Hung, M. Cord, and P. Pérez (2019). Zero-Shot Semantic Segmentation, in *Advances in Neural Information Processing Systems 2019*. Cited on page 25.

H. Caesar, J. Uijlings, and V. Ferrari (2016). Region-based semantic segmentation with end-to-end training, in *ECCV 2016*. Cited on page 24.

H. Caesar, J. Uijlings, and V. Ferrari (2018). COCO-Stuff: Thing and Stuff Classes in Context, in *CVPR 2018*. Cited on page 116.

K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles (2019). Few-shot video classification via temporal alignment, *arXiv preprint arXiv:1906.11415*. Cited on pages 25, 126, 127, 129, 133, 134, 135, 136, and 163.

J. Carreira and A. Zisserman (2017). Quo vadis, action recognition? a new model and the kinetics dataset, in *CVPR 2017*. Cited on pages 135 and 143.

S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha (2016). Synthesized Classifiers for Zero-Shot Learning, in *CVPR 2016*. Cited on pages 19, 52, 58, 59, 63, 64, 66, 67, 68, 69, 70, 71, 72, 73, 75, 77, 93, 110, 121, and 146.

S. Changpinyo, W.-L. Chao, and F. Sha (2017). Predicting visual exemplars of unseen classes for zero-shot learning, in *Proceedings of the IEEE international conference on computer vision 2017*. Cited on page 19.

W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha (2016). An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild, in *ECCV 2016*. Cited on pages 54, 80, 115, and 120.

O. Chapelle, B. Scholkopf, and A. Zien (2009). Semi-supervised learning, *IEEE Transactions on Neural Networks*, vol. 20(3), pp. 542–542. Cited on page 59.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). SMOTE: synthetic minority over-sampling technique, *Journal of artificial intelligence research*. Cited on pages 5 and 79.

N. V. Chawla, N. Japkowicz, and A. Kotcz (2004). Editorial: Special Issue on Learning from Imbalanced Data Sets, *SIGKDD Explor. Newsl.*. Cited on page 5.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *TPAMI*. Cited on pages 24, 111, 112, 116, 118, 122, and 143.

Q. Chen and V. Koltun (2017). Photographic Image Synthesis with Cascaded Refinement Networks, in *ICCV 2017*. Cited on page 80.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton (2020). A simple framework for contrastive learning of visual representations, *arXiv preprint arXiv:2002.05709*. Cited on pages 4 and 150.

W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang (2019). A Closer Look at Few-shot Classification, in *International Conference on Learning Representations 2019*. Cited on pages 22, 126, 128, 130, and 135.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, in *NIPS 2016*. Cited on page 81.

K. Crammer and Y. Singer (2002). On the Learnability and Design of Output Codes for Multiclass Problems, *ML*. Cited on page 35.

J. Deng, W. Dong, R., L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database, in *CVPR 2009*. Cited on pages 7, 47, 52, 60, 62, 86, 92, and 101.

J. Deng, J. Krause, and L. Fei-Fei (2013). Fine-Grained Crowdsourcing for Fine-Grained Recognition, in *CVPR 2013*. Cited on page 35.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*. Cited on page 150.

Z. Ding, M. Shao, and Y. Fu (2017). Low-Rank Embedded Ensemble Semantic Dictionary for Zero-Shot Learning, in *CVPR 2017*. Cited on pages 52 and 112.

P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie (2005). Behavior recognition via sparse spatio-temporal features. Cited on page 128.

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell (2014). DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition., in *ICML 2014*. Cited on page 52.

N. Dong and E. P. Xing (2018). Few-Shot Semantic Segmentation with Prototype Learning. Cited on pages 24, 25, 111, and 112.

A. Dosovitskiy and T. Brox (2016a). Generating images with perceptual similarity metrics based on deep networks, in *NIPS 2016*. Cited on page 106.

A. Dosovitskiy and T. Brox (2016b). Inverting visual representations with convolutional networks, in *CVPR 2016*. Cited on page 106.

M. Douze, A. Szlam, B. Hariharan, and H. Jégou (2018). Low-shot learning with large-scale diffusion, in *CVPR 2018*. Cited on page 127.

K. Duan, D. Parikh, D. J. Crandall, and K. Grauman (2012). Discovering localized attributes for fine-grained recognition, in *CVPR 2012*. Cited on pages 28 and 35.

S. K. Dwivedi, V. Gupta, R. Mitra, S. Ahmed, and A. Jain (2019). ProtoGAN: Towards Few Shot Learning for Action Recognition, *arXiv preprint arXiv:1909.07945*. Cited on page 128.

M. Elhoseiny, B. Saleh, and A. Elgammal (). Write a classifier: Zero-shot learning using purely textual descriptions. Cited on page 17.

M. Elhoseiny, B. Saleh, and A. Elgammal (2013). Write a classifier: Zero-shot learning using purely textual descriptions, in *Proceedings of the IEEE International Conference on Computer Vision 2013*. Cited on page 19.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*, *http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html*. Cited on page 116.

A. Farhadi, I. Endres, and D. Hoiem (2010). Attribute-centric recognition for cross-category generalization, in *CVPR 2010*. Cited on page 28.

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth (2009). Describing objects by their attributes, *CVPR*. Cited on pages 7, 18, 28, 30, 52, 53, 60, 63, and 161.

C. Feichtenhofer, H. Fan, J. Malik, and K. He (2019). Slowfast networks for video recognition, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. Cited on page 129.

C. Feichtenhofer, A. Pinz, and R. Wildes (2016a). Spatiotemporal residual networks for video action recognition, in *Advances in neural information processing systems 2016*. Cited on page 128.

C. Feichtenhofer, A. Pinz, and R. P. Wildes (2017). Spatiotemporal multiplier networks for video action recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2017*. Cited on page 25.

C. Feichtenhofer, A. Pinz, and A. Zisserman (2016b). Convolutional two-stream network fusion for video action recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2016*. Cited on pages 24, 25, 128, and 129.

R. Felix, V. K. B. G, I. Reid, and G. Carneiro (2018a). Multi-modal Cycle-consistent Generalized Zero-Shot Learning, in *ECCV 2018*. Cited on pages 95, 96, 98, and 102.

R. Felix, V. B. Kumar, I. Reid, and G. Carneiro (2018b). Multi-modal cycle-consistent generalized zero-shot learning, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. Cited on pages 19 and 20.

P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan (2010). Object Detection with Discriminatively Trained Part Based Models, *PAMI*. Cited on pages 29 and 31.

V. Ferrari and A. Zisserman (2007). Learning Visual Attributes, in *NIPS 2007*. Cited on page 28.

C. Finn, P. Abbeel, and S. Levine (2017). Model-agnostic meta-learning for fast adaptation of deep networks, in *ICML 2017*. Cited on pages 22, 23, and 128.

C. Finn, K. Xu, and S. Levine (2018). Probabilistic model-agnostic meta-learning, in *Advances in Neural Information Processing Systems 2018*. Cited on page 151.

A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov (2013). Devise: A deep visual-semantic embedding model, in *NIPS 2013*. Cited on pages 27, 28, 29, 32, 52, 54, 55, 67, 68, 69, 70, 71, 72, 75, 77, 84, 98, 115, and 146.

Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong (2014). Transductive Multi-view Embedding for Zero-Shot Recognition and Annotation, in *ECCV 2014*. Cited on pages 19 and 20.

Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong (2015a). Transductive Multi-view Zero-Shot Learning, *TPAMI*. Cited on page 98.

Y. Fu and L. Sigal (2016). Semi-Supervised Vocabulary-Informed Learning, in *CVPR 2016*. Cited on page 28.

Z. Fu, T. Xiang, E. Kodirov, and S. Gong (2015b). Zero-Shot Object Recognition by Semantic Manifold Distance, in *CVPR 2015*. Cited on page 28.

L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen (2017). Video captioning with attention-based LSTM and semantic consistency, *IEEE Transactions on Multimedia*, vol. 19(9), pp. 2045–2055. Cited on page 25.

S. Garcia and F. Herrera (2008). An Extension on"Statistical Comparisons of Classifiers over Multiple Data Sets"for all Pairwise Comparisons, *JLMR*. Cited on page 69.

A. Geiger, P. Lenz, and R. Urtasun (2012). Are we ready for autonomous driving? the kitti vision benchmark suite, in *2012 IEEE Conference on Computer Vision and Pattern Recognition 2012*. Cited on page 25.

D. Ghadiyaram, D. Tran, and D. Mahajan (2019). Large-scale weakly-supervised pre-training for video action recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. Cited on page 129.

R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell (2017). Actionvlad: Learning spatio-temporal aggregation for action classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. Cited on page 129.

R. Girshick (2015). Fast r-cnn, in *Proceedings of the IEEE international conference on computer vision 2015*. Cited on page 24.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative Adversarial Nets, in *NIPS 2014*. Cited on pages 20, 80, 81, 82, 89, 94, 97, 98, and 147.

J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner (2018). Meta-learning probabilistic inference for prediction, *arXiv preprint arXiv:1805.09921*. Cited on page 151.

R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, *et al.* (2017). The" Something Something" Video Database for Learning and Evaluating Visual Common Sense., in *ICCV 2017*. Cited on page 132.

A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola (2007). A kernel method for the two-sample-problem, in *NIPS 2007*. Cited on page 98.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville (2017). Improved training of wasserstein gans, *arXiv preprint arXiv:1704.00028*. Cited on pages 81, 83, 87, 94, 97, and 147.

B. Hariharan and R. Girshick (2017). Low-shot Visual Recognition by Shrinking and Hallucinating Features, *ICCV*. Cited on pages 21, 23, 82, 104, 105, 127, and 128.

T. Hastie, R. Tibshirani, and J. Friedman (2008). *The Elements of Statistical Learning (2nd Ed.)*, Springer. Cited on page 28.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick (2019). Momentum contrast for unsupervised visual representation learning, *arXiv preprint arXiv:1911.05722*. Cited on page 150.

K. He, G. Gkioxari, P. Dollár, and R. Girshick (2017). Mask r-cnn, in *Proceedings of the IEEE international conference on computer vision 2017*. Cited on page 24.

K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition, in *CVPR 2016*. Cited on pages 2, 25, 28, 53, 63, 86, 116, 118, 136, 143, and 155.

L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell (2016). Generating visual explanations, in *ECCVVision 2016*. Cited on page 107.

G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger (2019). Convolutional Networks with Dense Connectivity, *IEEE transactions on pattern analysis and machine intelligence*. Cited on page 15.

S. Hussain and B. Triggs (2010). Feature Sets and Dimensionality Reduction for Visual Object Detection, in *BMVC 2010*. Cited on pages 29 and 31.

L. Jain, W. Scheirer, and T. Boult (2014). Multi-Class Open Set Recognition Using Probability of Inclusion, in *ECCV 2014*. Cited on page 54.

D. Jayaraman and K. Grauman (2014). Zero-shot recognition with unreliable attributes, in *NIPS 2014*. Cited on page 18.

J. Ji, S. Buch, A. Soto, and J. C. Niebles (2018a). End-to-End Joint Semantic Segmentation of Actors and Actions in Video, in *ECCV 2018*. Cited on page 111.

Z. Ji, Y. Fu, J. Guo, Y. Pang, Z. M. Zhang, *et al.* (2018b). Stacked semantics-guided attention model for fine-grained zero-shot learning, in *Advances in Neural Information Processing Systems 2018*. Cited on page 18.

T. Joachims (2002). Optimizing search engines using clickthrough data, in *ACM SIGKDD 2002*. Cited on page 55.

A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov (2016a). Fasttext. zip: Compressing text classification models, *arXiv preprint arXiv:1612.03651*. Cited on pages 16, 112, 116, and 134.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov (2017). Bag of Tricks for Efficient Text Classification, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers 2017*. Cited on page 131.

A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache (2016b). Learning visual features from large weakly supervised data, in *European Conference on Computer Vision 2016*. Cited on page 128.

Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio (2017). Learning to remember rare events, *arXiv preprint arXiv:1703.03129*. Cited on page 135.

M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing (2019). Rethinking knowledge graph propagation for zero-shot learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. Cited on page 19.

B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell (2019). Few-shot object detection via feature reweighting, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. Cited on page 24.

P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa (2012). Online incremental attribute-based zero-shot learning, in *CVPR 2012*. Cited on pages 28 and 35.

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei (2014). Large-scale video classification with convolutional neural networks, in *CVPR 2014*. Cited on pages 24, 128, 129, and 134.

W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.* (2017). The kinetics human action video dataset, *arXiv preprint arXiv:1705.06950*. Cited on pages 129 and 132.

C. C. Kemp, A. Edsinger, and E. Torres-Jara (2007). Challenges for robot manipulation in human environments [grand challenges of robotics], *IEEE Robotics & Automation Magazine*, vol. 14(1), pp. 20–29. Cited on page 25.

A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele (2017). Simple Does It: Weakly Supervised Instance and Semantic Segmentation., in *CVPR 2017*. Cited on pages 24, 110, and 111.

A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei (). *Stanford Dogs Dataset*, *http://vision.stanford.edu/aditya86/ImageNetDogs/*. Cited on pages 27, 30, and 35.

D. P. Kingma and J. Ba (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*. Cited on page 116.

D. P. Kingma and M. Welling (2014). Auto-encoding variational bayes, in *ICLR 2014*. Cited on pages 20, 96, 97, 98, and 99.

T. N. Kipf and M. Welling (2017). Semi-supervised classification with graph convolutional networks, in *ICLR 2017*. Cited on page 149.

E. Kodirov, T. Xiang, Z. Fu, and S. Gong (2015). Unsupervised domain adaptation for zero-shot learning, in *ICCV 2015*. Cited on pages 19, 20, and 28.

E. Kodirov, T. Xiang, and S. Gong (2017). Semantic Autoencoder for Zero-Shot Learning, in *CVPR 2017*. Cited on pages 18, 55, 56, 66, 67, 68, 69, 70, 71, 72, 75, 77, and 146.

B. Korbar, D. Tran, and L. Torresani (2018). Cooperative learning of audio and video models from self-supervised synchronization, in *NeurIPS 2018*. Cited on pages 129, 138, 139, and 163.

A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks, in *NIPS 2012*. Cited on pages 15, 28, and 30.

V. Kumar Verma, G. Arora, A. Mishra, and P. Rai (2018a). Generalized zero-shot learning via synthesized examples, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2018*. Cited on pages 19 and 20.

V. Kumar Verma, G. Arora, A. Mishra, and P. Rai (2018b). Generalized Zero-Shot Learning via Synthesized Examples, in *CVPR 2018*. Cited on pages 95, 96, 98, and 102.

C. Lampert, H. Nickisch, and S. Harmeling (2013). Attribute-based classification for zero-shot visual object categorization, in *TPAMI 2013*. Cited on pages 7, 17, 18, 19, 27, 28, 30, 35, 36, 37, 38, 42, 52, 57, 58, 60, 61, 62, 63, 64, 66, 67, 68, 71, 75, 77, 80, 85, 86, 98, 110, 112, 121, 146, 156, and 161.

I. Laptev (2005). On space-time interest points, *International journal of computer vision*, vol. 64(2-3), pp. 107–123. Cited on page 128.

H. Larochelle, D. Erhan, and Y. Bengio (2008). Zero-data Learning of new tasks, in *AAAI 2008*. Cited on pages 28 and 52.

A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther (2016). Autoencoding beyond pixels using a learned similarity metric, in *ICML 2016*. Cited on page 99.

Y. LeCun, Y. Bengio, and G. Hinton (2015). Deep learning, *nature*, vol. 521(7553), pp. 436–444. Cited on pages 15 and 149.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (1989). Backpropagation applied to handwritten zip code recognition, *Neural computation*, vol. 1(4), pp. 541–551. Cited on page 15.

C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.* (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network., in *CVPR 2017*. Cited on page 97.

J. Lei Ba, K. Swersky, S. Fidler, *et al.* (2015). Predicting deep zero-shot convolutional neural networks using textual descriptions, in *Proceedings of the IEEE International Conference on Computer Vision 2015*. Cited on page 19.

Y. Li, K. Swersky, and R. Zemel (2015). Generative moment matching networks, in *ICML 2015*. Cited on pages 20, 82, and 98.

D. Lin, J. Dai, J. Jia, K. He, and J. Sun (2016). Scribblesup: Scribble-supervised convolutional networks for semantic segmentation, in *CVPR 2016*. Cited on pages 110 and 111.

S. Liu, M. Long, J. Wang, and M. I. Jordan (2018). Generalized zero-shot learning with deep calibration network, in *Advances in Neural Information Processing Systems 2018*. Cited on page 19.

W. Liu, A. Rabinovich, and A. C. Berg (2016). Parsenet: Looking wider to see better, *ICLR workshop*. Cited on page 111.

J. Long, E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation, in *CVPR 2015*. Cited on pages 2, 24, 111, 112, and 155.

D. G. Lowe (1999). Object recognition from local scale-invariant features, in *Proceedings of the seventh IEEE international conference on computer vision 1999*. Cited on page 14.

D. G. Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, vol. 60, pp. 91–110. Cited on pages 52 and 149.

D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten (2018). Exploring the limits of weakly supervised pretraining, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. Cited on page 129.

A. Mahendran and A. Vedaldi (2015). Understanding deep image representations by inverting them, in *CVPR 2015*. Cited on page 106.

T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka (2012). Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost, in *ECCV 2012*. Cited on page 127.

T. E. J. Mensink, E. Gavves, and C. G. M. Snoek (2014). COSTA: Co-Occurrence Statistics for Zero-Shot Classification, in *CVPR 2014*. Cited on page 28.

K. Mikolajczyk and C. Schmid (2004). Scale & affine invariant interest point detectors, *International journal of computer vision*, vol. 60(1), pp. 63–86. Cited on page 14.

K. Mikolajczyk and C. Schmid (2005). A performance evaluation of local descriptors, *IEEE transactions on pattern analysis and machine intelligence*, vol. 27(10), pp. 1615–1630. Cited on page 14.

T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin (2018). Advances in Pre-Training Distributed Word Representations, in *LREC 2018*. Cited on page 116.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013a). Distributed representations of words and phrases and their compositionality, in *Advances in neural information processing systems 2013*. Cited on page 16.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013b). Distributed Representations of Words and Phrases and their Compositionality, in *NIPS 2013*. Cited on pages 28, 30, 36, 63, 92, 96, 105, 112, 114, 116, and 149.

G. A. Miller (1995). WordNet: a lexical database for English, *CACM*, vol. 38, pp. 39–41. Cited on pages 30, 36, 62, and 112.

M. Mirza and S. Osindero (2014). Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784*. Cited on pages 81, 82, and 97.

A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murthy (2018). A generative model for zero shot learning using conditional variational autoencoders, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2018*. Cited on pages 19 and 20.

T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida (2018). Spectral normalization for generative adversarial networks, in *ICLR 2018*. Cited on page 97.

Q. Nguyen, M. C. Mukkamala, and M. Hein (2019). On the loss landscape of a class of deep neural networks with no bad local valleys. Cited on page 15.

M.-E. Nilsback and A. Zisserman (2008). Automated Flower Classification over a Large Number of Classes, in *ICCVGI 2008*. Cited on pages 17, 85, 86, and 101.

M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean (2014). Zero-Shot Learning by Convex Combination of Semantic Embeddings, in *ICLR 2014*. Cited on pages 18, 52, 58, 67, 69, 71, 72, 75, 77, and 146.

S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, B. Schiele, *et al.* (2017). Exploiting saliency for object segmentation from image level labels, in *CVPR 2017*. Cited on pages 110 and 111.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks, in *CVPR 2015*. Cited on page 4.

A. Owens and A. A. Efros (2018). Audio-visual scene analysis with self-supervised multisensory features, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. Cited on page 129.

M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell (2009). Zero-shot learning with semantic output codes, in *NIPS 2009*. Cited on pages 28 and 31.

D. Papadopoulos, A. Clarke, F. Keller, and V. Ferrari (2014). Training Object Class Detectors from Eye Tracking Data, in *ECCV 2014*. Cited on page 28.

G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille (2015). Weakly-and semi-supervised learning of a dcnn for semantic image segmentation, in *ICCV 2015*. Cited on pages 110 and 111.

D. Parikh and K. Grauman (2011). Relative attributes, in *ICCV 2011*. Cited on page 28.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer (2017). Automatic differentiation in PyTorch, in *NIPS-W 2017*. Cited on page 116.

D. Pathak, E. Shelhamer, J. Long, and T. Darrell (2015). Fully convolutional multi-class multiple instance learning, in *ICLR workshop 2015*. Cited on pages 110 and 111.

G. Patterson and J. Hays (2012). SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes, in *CVPR 2012*. Cited on pages 7, 52, 60, 61, 63, 85, 86, 101, 121, and 161.

J. Pennington, R. Socher, and C. D. Manning (2014). GloVe: Global Vectors for Word Representation, in *EMNLP 2014*. Cited on pages 16, 28, 30, 36, and 149.

S. Purushwalkam, M. Nickel, A. Gupta, and M. Ranzato (2019). Task-Driven Modular Networks for Zero-Shot Compositional Learning, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. Cited on page 149.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2017*. Cited on page 24.

H. Qi, M. Brown, and D. G. Lowe (2018). Low-Shot Learning With Imprinted Weights, in *CVPR 2018*. Cited on pages 21, 22, 103, 104, 126, and 128.

R. Qiao, L. Liu, C. Shen, and A. van den Hengel (2016). Less Is More: Zero-Shot Learning From Online Textual Documents With Noise Suppression, in *CVPR 2016*. Cited on pages 28 and 112.

S. Qiao, C. Liu, W. Shen, and A. L. Yuille (2018). Few-shot image recognition by predicting parameters from activations, in *CVPR 2018*. Cited on page 22.

A. Radford, L. Metz, and S. Chintala (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks., in *ICLR 2016*. Cited on pages 81 and 97.

P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.* (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning, *arXiv preprint arXiv:1711.05225*. Cited on page 151.

K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine (2018). Conditional Networks for Few-Shot Semantic Segmentation. Cited on pages 24 and 112.

S. Ravi and H. Larochelle (2016). Optimization as a model for few-shot learning, in *ICLR 2016*. Cited on pages 23, 126, 127, and 128.

A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson (2014). CNN features off-the-shelf: an astounding baseline for recognition, in *CVPR Workshops 2014*. Cited on page 28.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi (2016). You only look once: Unified, real-time object detection, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2016*. Cited on page 24.

S. Reed, Z. Akata, H. Lee, and B. Schiele (2016a). Learning Deep Representations of Fine-Grained Visual Descriptions, in *CVPR 2016*. Cited on pages 85, 86, 92, 101, 108, and 112.

S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee (2016b). Learning What and Where to Draw, in *NIPS 2016*. Cited on page 81.

S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee (2016c). Generative Adversarial Text to Image Synthesis, in *ICML 2016*. Cited on pages 80, 81, and 97.

G. Riegler, A. Osman Ulusoy, and A. Geiger (2017). Octnet: Learning deep 3d representations at high resolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. Cited on page 24.

M. Rohrbach, S. Ebert, and B. Schiele (2013). Transfer Learning in a Transductive Setting, in *NIPS 2013*. Cited on pages 45 and 98.

M. Rohrbach, M. Stark, and B.Schiele (2011). Evaluating knowledge transfer and zero-shot learning in a large-scale setting, in *CVPR 2011*. Cited on pages 35, 45, 52, 54, and 62.

M. Rohrbach, M. Stark, G. Szarvas, and B. Schiele (2012). Combining Language Sources and Robust Semantic Relatedness for Attribute-Based Knowledge Transfer, *Trends and Topics in Computer Vision*. Cited on page 17.

B. Romera-Paredes, E. OX, and P. H. Torr (2015). An embarrassingly simple approach to zero-shot learning, in *ICML 2015*. Cited on pages 18, 27, 28, 29, 31, 37, 38, 52, 55, 56, 66, 67, 68, 71, 72, 75, 77, 146, and 161.

O. Ronneberger, P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation, in *International Conference on Medical image computing and computer-assisted intervention 2015*. Cited on page 24.

S. Sadanand and J. J. Corso (2012). Action bank: A high-level representation of activity in video, in *2012 IEEE Conference on Computer Vision and Pattern Recognition 2012*. Cited on page 128.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen (2016). Improved techniques for training gans, in *NIPS 2016*. Cited on page 80.

J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek (2013). Image classification with the fisher vector: Theory and practice, *International journal of computer vision*, vol. 105(3), pp. 222–245. Cited on page 15.

W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult (2013). Towards Open Set Recognition, *TPAMI*. Cited on page 54.

E. Schoenfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata (2019). Generalized Zero- and Few-Shot Learning via Aligned Variational Autoencoders, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. Cited on page 128.

E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata (2019). Generalized zero-and few-shot learning via aligned variational autoencoders, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. Cited on pages 19 and 20.

A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots (2017). One-shot learning for semantic segmentation, in *BMVC 2017*. Cited on pages 24, 25, 110, and 112.

K. Simonyan and A. Zisserman (2014a). Two-stream convolutional networks for action recognition in videos, in *Advances in neural information processing systems 2014*. Cited on pages 128 and 129.

K. Simonyan and A. Zisserman (2014b). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*. Cited on pages 15, 52, 116, and 118.

J. Snell, K. Swersky, and R. Zemel (2017). Prototypical networks for few-shot learning, in *NIPS 2017*. Cited on pages 22, 23, 25, 105, and 112.

R. Socher, M. Ganjoo, C. D. Manning, and A. Ng (2013). Zero-Shot Learning Through Cross-Modal Transfer, in *NIPS 2013*. Cited on pages 17, 27, 28, 31, 37, 38, 45, 52, 54, 57, 66, 67, 69, 71, 72, 75, 77, 146, and 161.

J. Song, C. Shen, Y. Yang, Y. Liu, and M. Song (2018). Transductive Unbiased Embedding for Zero-Shot Learning, in *CVPR 2018*. Cited on pages 19 and 102.

K. Soomro, A. R. Zamir, and M. Shah (2012). UCF101: A dataset of 101 human actions classes from videos in the wild, *arXiv preprint arXiv:1212.0402*. Cited on page 132.

F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales (2018). Learning to compare: Relation network for few-shot learning, in *CVPR 2018*. Cited on page 23.

T. Sylvain, L. Petrini, and D. Hjelm (2019). Locality and compositionality in zero-shot learning, *arXiv preprint arXiv:1912.12179*. Cited on page 148.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going deeper with convolutions, in *CVPR 2015*. Cited on pages 15, 28, 36, 63, and 67.

B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li (2015). YFCC100M: The new data in multimedia research, *arXiv preprint arXiv:1503.01817*. Cited on pages 127, 128, 130, 131, 133, 138, 159, and 163.

A. Torralba, A. A. Efros, *et al.* (2011). Unbiased look at dataset bias., in *CVPR 2011*. Cited on page 128.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri (2015). Learning spatiotemporal features with 3d convolutional networks, in *CVPR 2015*. Cited on pages 128, 129, and 135.

D. Tran, H. Wang, L. Torresani, and M. Feiszli (2019). Video Classification with Channel-Separated Convolutional Networks, *arXiv preprint arXiv:1904.02811*. Cited on page 129.

D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri (2018). A closer look at spatiotemporal convolutions for action recognition, in *CVPR 2018*. Cited on pages 2, 129, 134, 135, and 155.

E. Triantafillou, R. Zemel, and R. Urtasun (2017). Few-shot learning through an information retrieval lens, in *NeurIPS 2017*. Cited on page 23.

E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, *et al.* (2019). Meta-dataset: A dataset of datasets for learning to learn from few examples, *arXiv preprint arXiv:1903.03096*. Cited on pages 22 and 151.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun (2005). Large Margin Methods for Structured and Interdependent Output Variables, *JMLR*. Cited on page 56.

N. Usunier, D. Buffoni, and P. Gallinari (2009). Ranking with Ordered Weighted Pairwise Classification, in *ICML 2009*. Cited on page 56.

L. van der Maaten and G. Hinton (2008). Visualizing High-Dimensional Data Using t-SNE, *JMLR*. Cited on page 46.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need, in *Advances in neural information processing systems 2017*. Cited on page 151.

V. K. Verm and P. Rai (2017). A Simple Exponential Family Framework for Zero-Shot Learning, in *ECML 2017*. Cited on pages 59, 66, 69, 72, 74, 75, 77, and 146.

V. K. Verma and P. Rai (2017). A simple exponential family framework for zero-shot learning, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2017*. Cited on pages 19 and 121.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.* (2016). Matching networks for one shot learning, in *NIPS 2016*. Cited on pages 22, 23, 25, 112, 126, and 128.

B. Wallace and B. Hariharan (2019). Few-Shot Generalization for Single-Image 3D Reconstruction via Priors, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. Cited on page 24.

H. Wang and C. Schmid (2013). Action recognition with improved trajectories, in *Proceedings of the IEEE international conference on computer vision 2013*. Cited on page 128.

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool (2016). Temporal segment networks: Towards good practices for deep action recognition, in *European conference on computer vision 2016*. Cited on page 128.

X. Wang and A. Gupta (2016). Generative image modeling using style and structure adversarial networks, in *ECCV 2016*. Cited on page 81.

X. Wang, Y. Ye, and A. Gupta (2018a). Zero-shot recognition via semantic embeddings and knowledge graphs, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2018*. Cited on pages 19 and 116.

X. Wang, Y. Ye, and A. Gupta (2018b). Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs, in *CVPR 2018*. Cited on page 149.

X. Wang, F. Yu, R. Wang, T. Darrell, and J. E. Gonzalez (2019a). Tafe-net: Task-aware feature embeddings for low shot learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. Cited on page 22.

Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten (2019b). SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning, *arXiv preprint arXiv:1911.04623*. Cited on page 128.

Y. Wang, R. Girshick, M. Hebert, and B. Hariharan (2018c). Low-Shot Learning from Imaginary Data, in *CVPR 2018*. Cited on pages 23, 105, and 128.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona (2010). Caltech-UCSD Birds 200, Technical report CNS-TR-2010-001, Caltech. Cited on pages 7, 17, 27, 30, 35, 52, 60, 61, 63, 85, 86, 101, 121, and 161.

J. Weston, S. Bengio, and N. Usunier (2011). WSABIE: Scaling Up to Large Vocabulary Image Annotation, in *IJCAI 2011*. Cited on pages 30, 31, 32, 35, and 56.

Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele (2016). Latent Embedding for Zero-shot Recognition, in *CVPR 2016*. Cited on pages 9, 28, 52, 57, 66, 67, 68, 71, 72, 75, 77, 85, 115, and 146.

Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata (2019a). SPNet: Semantic Projection Network for Zero- and Few-Label Semantic Segmentation, in *CVPR 2019*. Cited on page 10.

Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata (2019b). Zero-shot learning-A comprehensive evaluation of the good, the bad and the ugly, *TPAMI*. Cited on pages 10, 101, 103, 116, 117, 120, 121, and 157.

Y. Xian, T. Lorenz, B. Schiele, and Z. Akata (2018). Feature Generating Networks for Zero-Shot Learning, in *CVPR 2018*. Cited on pages 10, 95, 96, 98, 99, 102, 103, and 157.

Y. Xian, B. Schiele, and Z. Akata (2017). Zero-Shot Learning - The Good, the Bad and the Ugly, in *CVPR 2017*. Cited on pages 10, 80, 82, 86, 87, 92, and 157.

Y. Xian, S. Sharma, B. Schiele, and Z. Akata (2019c). F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019*. Cited on pages 10 and 128.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba (2010). SUN database: Large-scale scene recognition from abbey to zoo, in *CVPR 2010*. Cited on page 18.

G.-S. Xie, L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao, and L. Shao (2019). Attentive region embedding network for zero-shot learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. Cited on page 18.

X. Xu, T. Hospedales, and S. Gong (2015). Semantic embedding space for zero-shot action recognition, in *2015 IEEE International Conference on Image Processing (ICIP) 2015*. Cited on page 25.

X. Xu, Y. Yang, D. Zhang, H. T. Shen, and J. Song (2017). Matrix Tri-Factorization with Manifold Regularizations for Zero-shot Learning, in *CVPR 2017*. Cited on page 52.

I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan (2019). Billion-scale semi-supervised learning for image classification, *arXiv preprint arXiv:1905.00546*. Cited on page 129.

J. Yang, K. Yu, Y. Gong, and T. Huang (2009). Linear spatial pyramid matching using sparse coding for image classification, in *2009 IEEE Conference on computer vision and pattern recognition 2009*. Cited on page 15.

Y. Yang and D. Ramanan (2011). Articulated pose estimation with flexible mixtures-of-parts, in *CVPR 2011*. Cited on pages 29 and 31.

M. Ye and Y. Guo (2017). Zero-shot classification with discriminative semantic representation learning, in *CVPR 2017*. Cited on pages 59, 60, 66, 77, and 146.

J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn (2018). Bayesian model-agnostic meta-learning, in *Advances in Neural Information Processing Systems 2018*. Cited on page 151.

X. Yu and Y. Aloimonos (2010). Attribute-Based Transfer Learning for Object Categorization with Zero or One Training Example, in *ECCV 2010*. Cited on pages 28, 35, and 52.

J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici (2015). Beyond short snippets: Deep networks for video classification, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2015*. Cited on page 129.

R. Zabih and J. Woodfill (1994). Non-parametric local transforms for computing visual correspondence, in *European conference on computer vision 1994*. Cited on page 14.

H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal (2018a). Context encoding for semantic segmentation, in *CVPR 2018*. Cited on pages 24 and 111.

H. Zhang, X. Shang, W. Yang, H. Xu, H. Luan, and T.-S. Chua (2016a). Online Collaborative Learning for Open-Vocabulary Visual Classifiers, in *CVPR 2016*. Cited on page 54.

H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas (2016b). Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition, in *CVPR 2016*. Cited on page 148.

H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas (2017a). Stack-GAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, in *ICCV 2017*. Cited on pages 80, 81, and 93.

L. Zhang, T. Xiang, and S. Gong (2017b). Learning a deep embedding model for zero-shot learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. Cited on page 18.

N. Zhang, J. Donahue, R. Girshick, and T. Darrell (2014). Part-based R-CNNs for fine-grained category detection, in *ECCV 2014*. Cited on page 148.

X. Zhang, Y. Wei, Y. Yang, and T. Huang (2018b). SG-One: Similarity Guidance Network for One-Shot Semantic Segmentation, *arXiv preprint arXiv:1810.09091*. Cited on page 112.

Z. Zhang and V. Saligrama (2015). Zero-Shot Learning via Semantic Similarity Embedding, in *ICCV 2015*. Cited on pages 27, 37, 38, 52, 58, 66, 67, 68, 71, 72, 75, 77, 146, and 161.

Z. Zhang and V. Saligrama (2016). Zero-Shot Learning via Joint Semantic Similarity Embedding, in *CVPR 2016*. Cited on pages 27, 37, 38, 110, 115, and 161.

H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba (2017a). Open Vocabulary Scene Parsing, in *ICCV 2017*. Cited on pages 25, 111, and 112.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia (2017b). Pyramid Scene Parsing Network, in *CVPR 2017*. Cited on page 111.

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr (2015). Conditional random fields as recurrent neural networks, in *ICCV 2015*. Cited on page 111.

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva (2014). Learning deep features for scene recognition using places database, in *NIPS 2014*. Cited on page 67.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf (2004). Learning with local and global consistency, in *NIPS 2004*. Cited on page 59.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, in *ICCV 2017*. Cited on page 97.

L. Zhu and Y. Yang (2018). Compound memory networks for few-shot video classification, in *ECCV 2018*. Cited on pages 25, 126, 127, 128, 129, 133, 134, 135, 136, and 163.

X. Zhu and D. Ramanan (2012). Face detection, pose estimation, and landmark localization in the wild, in *CVPR 2012*. Cited on pages 29 and 31.

Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal (2018a). A generative adversarial approach for zero-shot learning from noisy texts, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2018*. Cited on pages 19 and 20.

Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal (2018b). A Generative Adversarial Approach for Zero-Shot Learning From Noisy Texts, in *CVPR 2018*. Cited on page 98.

B. Zoph and Q. V. Le (2016). Neural architecture search with reinforcement learning, *arXiv preprint arXiv:1611.01578*. Cited on page 15.