Logo Recognition with the Use of Deep Convolutional Neural Networks

Ahmed Alsheikhy
Electrical Engineering Department
College of Engineering
Northern Border University
Arar, Saudi Arabia
aalsheikhy@nbu.edu.sa

Yahia Said
Electrical Engineering Department, College of Engineering,
Northern Border University, Arar, Saudi Arabia and
Laboratory of Electronics and Microelectronics, Faculty of
Sciences of Monastir, University of Monastir, Tunisia
said.yahia1@gmail.com

Mohammad Barr
Electrical Engineering Department
College of Engineering
Northern Border University
Arar, Saudi Arabia
m.abarr@nbu.edu.sa

Abstract—**Automatic logo recognition is gaining importance due to the increasing number of its applications. Unlike other object recognition tasks, logo recognition is more challenging because of the limited amount of the available original data. In this paper, the transfer leaning technique was applied to a Deep Convolutional Neural Network model to guarantee logo recognition using a small computational overhead. The proposed method was based on the Densely Connected Convolutional Networks (DenseNet). The experimental results show that for the FlickrLogos-32 logo recognition dataset, our proposed method performs comparably with state-of-the-art methods while using fewer parameters.**

Keywords-logo recognition; deep learning; Convolutional Neural Networks (CNNs); DenseNet; artificial intelligence

I. Introduction

Logos are symbols that are generally used by firms to identify themselves and their products. They normally contain colors, shapes, textures, and/or text [1]. Logo recognition is a key problem for many applications such as copyright infringement detection, online brand management, vehicle recognition, contextual advertisement placement, etc. [2, 3]. Although companies do not change their logos often, and it is only the context in which the logo appears that changes for each product of the same company, logo recognition is still a challenging task. Some of the challenges for accurate logo recognition are perspective deformations, varying background, occlusions, warping, varying size, varying colors, etc. [1]. Moreover, the growing number of brands having personalized logos makes the logo recognition task even more challenging. The logo recognition systems require high computational powers to support multi-class classification efficiently.

Traditionally, Artificial Intelligence (AI) techniques have been used to solve object recognition problems. In particular, Convolutional Neural Networks (CNNs) with deep structure and many hidden layers is a very popular model that is commonly used for solving object recognition problems [18, 19]. The approach followed by these techniques is based on two important tasks: feature extraction and feature classification. These tasks are commonly achieved using the convolutional layers as feature extraction modules and the Fully Connected (FC) layers for classification [17]. Many techniques derived from CNNs have been used for solving the problem of logo recognition. For example, authors in [4, 5] used pre-trained CNNs for logo recognition. However, such techniques have high computational overhead associated with them. This limits the context in which such computational intensive solutions can be used. Hence, the problem of accurate logo recognition using low computational effort remains unresolved.

In this paper, the transfer learning technique was applied to a Deep Convolutional Neural Network (DCNN) model to ensure logo recognition without using huge computation resources and the results of the accuracy comparison with the state-of-the-art methods show that the proposed method performs similarly.

II. Related works

Many solutions have been proposed for the problem of accurate logo recognition. Earlier works on logo recognition were mainly based on keypoint detectors and descriptors. A feature bundling was proposed by Romberg and Lienhard [6] for scalable logo recognition. Their method combined both local features and those from the spatial neighbourhood into a Bag of Words (BoW). Similarly, Romberg et al. [7] proposed another logo recognition system based on the relative spatial

Corresponding author: Yahia Said

layout of local features encoding and indexing (e.g. edges and triangles). The local features and the spatial layout helped them to quantize the regions in the logos. Francesconi [8] presented a Recursive Neural Network-based technique for the classification of black and white logos. The method also used contour trees to hold the topological structural information. Although Francesconi's method was efficient, its main limitation is that its performance for more complex colored logos is not known. Moreover, this approach assumes that the maximum number of children for each node are known in advance. Duffner and Garcia [9] also proposed a CNN-based solution for recognizing logos in television programs. In this technique, pixel values were directly fed into a CNN with two convolution layers to detect watermarks on television. The main limitations of this technique are its low detection rates and its limited applicability (i.e. only television logos can be detected). Zhu and Doermann [10] used Fisher classifiers for recognizing logos in documents. A problem with their technique is that it cannot handle large variations [11]. Authors in [3-5, 12] proposed Deep Learning-based solutions for the problem of accurate logo recognition. Authors in [4, 5] relied on pre-trained CNNs and synthetically generated data for logo detection. Similarly, authors in [3] proposed and evaluated several network architectures while authors in [12] used pre-trained CNN models along with Fast Region-Based Convolutional Networks. The main limitations of these techniques are that they rely on pre-trained CNNs. A major limitation is that the training data available for logo recognition was limited. Authors in [2] proposed a Deep Learning-based logo recognition. The proposed recognition pipeline includes a logo region proposal module followed by a CNN module that is trained for logo identification. This method can also handle logos that are not localized. However, its accuracy is still limited.

III. Proposed Architecture for Logo Recognition

CNNs are one of the most powerful Deep Learning models. A CNN is composed of multiple layers of Convolution, Pooling, ReLU correction and Fully-Connected (FC) stocked in a robust manner. CNN models have demonstrated an impressive ability to generalize large datasets with millions of images. The input image passes through multiple hidden layers to be filtered, corrected, and compressed many times, to finally form a vector. For the classification task, the output vector presents the probabilities of class membership. All CNNs must start with a convolutional layer and end with a fully-connected layer. The intermediate layers can be stacked in different ways, provided that the output of one layer has the same structure as the input of the next. In general, a CNN stacks multiple Convolution and ReLU correction layers then adds a Pooling layer (optional), and repeats this pattern multiple times. Then, it stacks the FC layers. The more layers there are, the more "deep" the neural network is.

To get more accurate CNNs and achieve better results, we must use a model that can learn more competitive representations without a dramatic increase in network parameters. As we tackle recognition with a limited amount of original data, we are interested in efficient representations that can be achieved with a small number of parameters. In effect, exploring the Densely Connected Convolutional Networks (DenseNet) [14] can lead to efficient results. In DenseNet, the original CNN layers are replaced by dense blocks and transition layers except for the first convolutional layer. It outperforms the state-of-the-art CNN models in the classification task on many datasets using a lower complexity network. A DenseNet architecture with three dense blocks and two transition layers is illustrated in Figure 1.

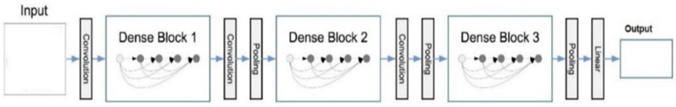

Fig. 1. DenseNet architecture.

DenseNet is a revolutionary Deep CNN with a stack of dense blocks and transition layers. A dense block is a group of convolution layers in which the layers are densely connected. Each layer receives as input all previous layers' outputs. A single dense block packs Convolution layers followed by a ReLU activations layer and a Batch Normalization layer. To reduce the size of the feature maps, DenseNet uses transition layers which are composed of a Batch Normalization layer, followed by a 1×1 convolution and a 2×2 average pooling. The transition layer reduces the height and the width dimensions but leaves feature dimensions the same. The DenseNet stacks hundreds of layers without any optimization complexity. Thus, DenseNet is one of the best deep CNN models for image classification and object recognition tasks. In order to classify logos, our custom-made model consists of four dense blocks and three transition layers. Figure 2 presents the proposed CNN model. Before the first dense block, a 7×7 convolution layer and a 3×3 Max Pooling layer are performed on the input images to extract more important features in order to detect small variations in the image. Between each consecutive dense blocks, a transition layer composed of a 1×1 convolution followed by 2×2 average pooling layers was used. A 7×7 global average pooling layer with stride of 2 is placed after the fourth dense block to fix the size of the feature maps to be connected to the fully connected layer. Finally, the transfer learning technique was used to configure the output layer for the logos classes instead of the original ImageNet dataset classes [15].

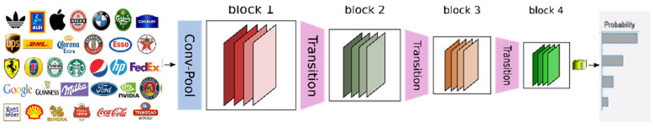

Fig. 2. The architecture of the proposed CNN model.

DenseNet was trained to classify natural images according to the ImageNet with 1000 classes. The Transfer Learning technique was applied which allows the reuse of existing Deep CNN models without the need of a long time computing. The main idea of the Transfer Learning technique is to reuse the existing CNN models designed for a special application for a new application. In addition to accelerating network training, Transfer Learning helps prevent overfitting. Indeed, when the collection of input images is limited, it is hard to train the CNN from scratch with random weight initialization. In effect, the number of parameters to learn is much higher than the number of images and this may lead to the risk of overfitting. So, implementing this technique is very effective and it is widely used in practice. It requires having a CNN already trained, preferably on a problem close to the one we want to solve.

IV. Experiments and Results

In order to evaluate the proposed model, the FlickrLogos-32 logo recognition dataset [7] was used for training and testing. FlickrLogos-32 is a publicly-available dataset with a collection of 8240 real-world images of 32 different brand logos from Flickr and it was built for the evaluation of logo retrieval and multi-class logo detection and recognition systems. The dataset was divided into three separated subsets: P1, P2, and P3. Each subset contains images of all the 32 classes. The training set P1 contains 320 images, 10 images per class, while both the validation set P2 and the test set P3 consist of 3960 images each containing both 30 positive examples per class and 3000 negative examples containing no logos. Figure 3 shows sample images of 32 logo classes of the FlickrLogos-32 dataset. All experiments were performed using the CAFFE (Convolutional Architecture for Fast Feature Embedding) framework [16] on an NVIDIA Tesla K40c GPU with 12 GB RAM. CAFFE is a Deep Learning framework written in C++, with a Python interface. It supports many different types of Deep Learning architectures geared towards image classification and image segmentation.

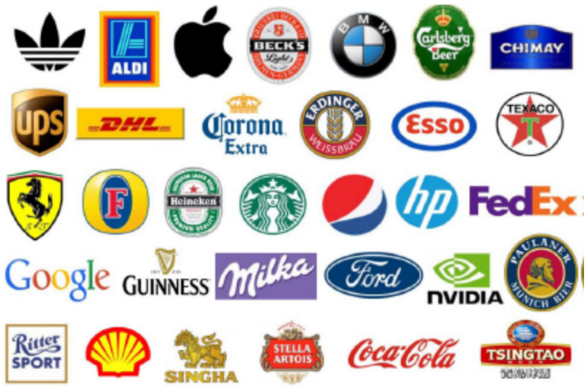

Fig. 3. The 32 classes of the FlickrLogos-32 dataset.

The weights of our custom-made network were initialized with a pre-trained DenseNet model on the ImageNet Dataset. Then, the network was fine-tuned on the FlickerLogo32 training set. In effect, the pre-trained weights were loaded to the first convolution layer, the dense blocks, and transition layers while the weights of the output layer were updated until optimizing the loss function. Then, the entire network including the first convolution layer was fine-tuned using the validation set. After the training of our model, the test set P3 was used for testing. Table I shows the achieved accuracy of the custom-made network compared to existing models. The proposed model achieves state-of-the-art performance with high classification accuracy. We can see that 92.8 % of average accuracy was achieved compared to 91.7 % achieved in [2].

TABLE I. CLASSIFICATION ACCURACIES ON FLICKRLOGOS-32 DATASET

Method	Dropout	Pretrain	Finetune	Accuracy
[3]	0.5	ImageNet-2012	FlickrLogos-32	70.1%
[3] (Full-inception)	0.7	ImageNet-2012	FlickrLogos-32	77.1%
[3] (GoogLeNet)	0.7	ImageNet-2012	FlickrLogos-32	87.6%
	0.8	ImageNet-2012	FlickrLogos-32	88.7%
[3] (GoogLeNet-GP)	0.8	ImageNet-2012	FlickrLogos-32	89.1%
	0.9	ImageNet-2012	FlickrLogos-32	89.6%
[2]	-	-	FlickrLogos-32	91.7%
[4]	-	-	FlickrLogos-32	84.6%
[1]	-	-	FlickrLogos-32	88.9%
Proposed	0.9	ImageNet-2012	FlickrLogos-32	92.8%

V. Conclusion

Logo recognition is considered an important task for many applications. In this paper, a Deep CNN model was designed for logo recognition based on the DenseNet model. The proposed model was trained and tested on the FlickerLogo32 dataset. The obtained results were very encouraging compared to the state-of-the-art works. The proposed logo recognition and classification model can be used in many applications like online brand management and contextual advertisement placement. As potential future works, the design of a real-time logo recognition system for a mobile application may be considered.

Acknowledgment

The authors wish to acknowledge the approval and the support of this research study by the grant N°ENG-2019-1-10-F-8140 from the Deanship of the Scientific Research in Northern Border University, Arar, KSA.

References

[1] R. Boia, A. Bandrabur, and C. Florea, “Local description using multi-scale complete rank transform for improved logo recognition,” in *2014 10th International Conference on Communications (COMM)*, May 2014, pp. 1–4, doi: 10.1109/ICComm.2014.6866723.

[2] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, “Deep Learning for Logo Recognition,” *Neurocomputing*, Jan. 2017, doi: 10.1016/j.neucom.2017.03.051.

[3] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, “DeepLogo: Hitting Logo Recognition with the Deep Neural Network Hammer,” *arXiv:1510.02131 [cs]*, Oct. 2015, Accessed: Aug. 12, 2020. [Online]. Available: http://arxiv.org/abs/1510.02131.

[4] C. Eggert, A. Winschel, and R. Lienhart, “On the Benefit of Synthetic Data for Company Logo Detection,” in *Proceedings of the 23rd ACM international conference on Multimedia*, Oct. 2015, pp. 1283–1286, doi: 10.1145/2733373.2806407.

[5] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, “Logo Recognition Using CNN Features,” in *Image Analysis and Processing — ICIAP 2015*, Cham, 2015, pp. 438–448, doi: 10.1007/978-3-319-23234-8_41.

[6] S. Romberg and R. Lienhart, “Bundle min-hashing for logo recognition,” in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, Apr. 2013, pp. 113–120, doi: 10.1145/2461466.2461486.

[7] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol, “Scalable logo recognition in real-world images,” in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, Apr. 2011, pp. 1–8, doi: 10.1145/1991996.1992021.

[8] E. Francesconi *et al.*, “Logo recognition by recursive neural networks,” in *Graphics Recognition Algorithms and Systems*, 1998, pp. 104–117, doi: 10.1007/3-540-64381-8_43.

[9] S. Duffner and C. Garcia, “A Neural Scheme for Robust Detection of Transparent Logos in TV Programs,” in *Artificial Neural Networks – ICANN 2006*, 2006, pp. 14–23, doi: 10.1007/11840930_2.

[10] G. Zhu and D. Doermann, “Automatic Document Logo Detection,” in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Sep. 2007, vol. 2, pp. 864–868, doi: 10.1109/ICDAR.2007.4377038.

[11] G. Zhu and D. Doermann, “Logo Matching for Document Image Retrieval,” in *2009 10th International Conference on Document Analysis and Recognition*, Jul. 2009, pp. 606–610, doi: 10.1109/ICDAR.2009.60.

[12] G. Oliveira, X. Frazão, A. Pimentel, and B. Ribeiro, “Automatic graphic logo detection via Fast Region-based Convolutional Networks,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 985–991, doi: 10.1109/IJCNN.2016.7727305.

[13] T. Williams and R. Li, “An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification,” *Journal of Software Engineering and Applications*, vol. 11, no. 2, pp. 69–88, Feb. 2018, doi: 10.4236/jsea.2018.112004.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[15] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[16] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, Nov. 2014, pp. 675–678, doi: 10.1145/2647868.2654889.

[17] U. Khan, K. Khan, F. Hassan, A. Siddiqui, and M. Afaq, “Towards Achieving Machine Comprehension Using Deep Learning on Non-GPU Machines,” *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4423–4427, Aug. 2019.

[18] Y. Said, M. Barr, and H. E. Ahmed, “Design of a Face Recognition System based on Convolutional Neural Network (CNN),” *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5608–5612, Jun. 2020.

[19] Y. F. Said and M. Barr, “Pedestrian Detection for Advanced Driver Assistance Systems using Deep Learning Algorithms,” *International Journal of Computer Science and Network Security*, vol. 19, no. 9, pp. 9–14, Sep. 2019.