Street View House Number Identification Based on Deep Learning

Yang Haoqi
School of computer science and engineering
Xi'an Technological University
Xi'an, China
e-mail: curioyhq@gmail.com

Yao Hongge
School of computer science and engineering
Xi'an Technological University
Xi'an, China
e-mail: 835092445@qq.com

Abstract—**In this paper, the difficult problem of character recognition in natural scenes caused by many factors such as variability of light in the natural scene, background clutter and inaccurate viewing angle, and inconsistent resolution. Based on the deep learning framework PyTorch, a convolutional neural network is implemented. Based on the classic LeNet-5 network, the network optimizes the input layer to accept three-channel images, changes the pooling method to maximum pooling to simplify parameters, and the activation function is replaced by Rectified Linear Unit with faster convergence. The cross-entropy loss is used instead of the minimum mean square error to mitigate the slow learning. Furthermore, we also enroll the gradient descent optimization algorithm RMSprop and L2 regularization to improve the accuracy, speed up the convergence and suppress the over-fitting. The experiment results show that our model achieved an accuracy of 92.32% after training for 7h24min on the street view house number(SVHN) dataset, effectively improving the performance of LeNet-5.**

Keywords-House Number Identification; Convolutional Neural Network; Lenet-5

I. INTRODUCTION

The traditional method of classifying house numbers from natural scene images is usually to use manual feature extraction[1-2] and template matching[3-4]. In order to identify the house number of the corridor environment, Zhang Shuai et al. used the combination of Robert edge detection and morphological operation to locate the position of the house number image, and then divide the house number by horizontal and vertical projection method, tilt correction, etc., and finally use pattern recognition to identify the house number [5]. Ma Liling et al. used the linear discriminant linear local tangent space alignment algorithm (ODLLTSA) and the support vector machine (SVM) method to identify the house number, use the extracted features to train the SVM classifier, and use the SVM classifier to the new house number classification [6].

For these traditional methods, the key to determining their performance is to have a good classifier, and the features in the classifier are mostly designed manually (such as SIFT, SURF, HOG, etc.), and the features of the artificial design are well interpreted. However, in the face of complex backgrounds, changing fonts and various deformations, it is rather troublesome and difficult to extract more general features[7].

The Convolutional Neural Network (CNN) is a multi-layered supervised learning neural network. Although the training process requires a large amount of data compared with the traditional method, the convolutional neural network can automatically summarize the target feature from these data. Features do not require human intervention. Overcome the shortcomings of manual design features that are time-consuming and labor-intensive, have poor general use and require high experience in the designer field. It is precise because of these advantages of convolutional neural networks that a large number of researchers have begun to apply it to solve character recognition problems.

In response to this situation, we implemented a LeNet-5-based neural network based on the deep learning framework PyTorch and achieved an accuracy of 92.32% on the SVHN dataset at a time of 6 hours and 17 minutes.

II. RELATED WORK

A. Network structure

The network used in this experiment is modified by LeNet-5 as shown in Figure 1. LeNet-5 appeared to solve the problem of recognition of handwritten characters. The data set used in the training process is the MNIST. The samples in the data set are single-

DOI: 10.21307/ijanmc-2019-058

channel grayscale images, and the street view dataset is The three-channel color picture, to improve the robustness of the model, minimize the intervention on the original data set, we do not pre-process it, such as grayscale, but choose to adjust the input layer of LeNet-5 to three channels.

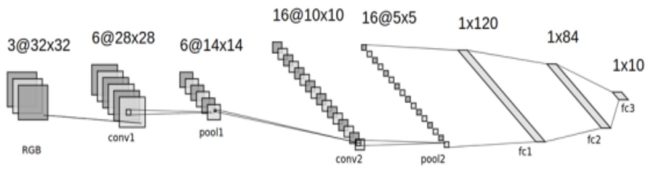

Figure 1. Network structure

The pooling layer in the original LeNet-5 network is very different from the currently recognized pooling layer operation, so we replace it directly with the max-pooling layer, which on the other hand reduces the number of trainable parameters of the network. It is conducive to controlling the scale of the network and speeding up the training. In terms of the activation function, the activation function in the original LeNet-5 is Sigmoid or TanH. Here we use a Rectified Linear Unit (ReLU) with faster convergence speed and no significant impact on the generalization accuracy of the model. LeNet-5's loss function is Minimum Mean Squared Error:

$$MSE = \frac{1}{n}\sum_{i}^{n}(\hat{y}_i - y_i)^2 \# \qquad (1)$$

Where n is the number of samples， $\hat{y}_i$ represents the predicted value of the ith sample, and y_i is the labelof the ith sample. In the case of back-propagation by the gradient descent method, the minimum mean square error is easy to occur when the neuron output is close to ‘1’ and the gradient is too small to learn slow. We use the cross-entropy loss function here:

$$L = -\sum_{i=1}^{n} y_i \, log(\hat{y}_i) \# \qquad (2)$$

In addition to the above improvements, we will introduce four optimization algorithms, SGD (with momentum), Adam, Adamax, and RMSprop.

B. Comparison effects of different optimizers

The package ‘torch.optim’ in PyTorch encapsulates a large number of optimization algorithms, which are often referred to as optimizers. In Figure 2, we take the more common SGD, Adam, Adamax and RMSprop optimizers according to the parameters listed in Table 1.After 90 epochs training, compare their optimization effects on the SVHN dataset used in the improved LeNet-5 network proposed in this paper. It can be seen from Figure 2 that the network using the SGD optimizer has almost no improvement in the test set accuracy in the first 14 epoch, and the 14th epoch only starts to rise significantly; the network using the other three optimizers is in the toptenepochs, a good test set accuracy rate is obtained, and the test set accuracy of the network using the RMSprop optimizer is the fastest. So in the next experiment, we will use RMSprop as the default optimizer.

TABLE I. OPTIMIZER PARAMETER SETTING

Optimizer	**parameter**
SGD	lr=0.001,
Adam	lr=0.001,
Adamax	lr=0.002,
RMSprop	lr=0.001,

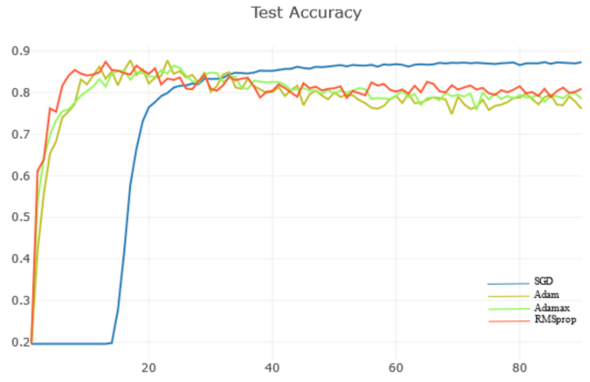

Figure 2. Optimizer effect

It can be observed from Figure 2 that the accuracy of the test set except the SGD optimizer is not improved much in the later stage of training, and the accuracy of the test set of the other three networks even shows a small downward trend. Table 2 shows the statistics in Figure 2. The data of test sets with high accuracy rate, the difference ofhighest accuracy rate on the SVHN test set is only 1.793816%, which can be considered as the optimization effect of the four optimizers on the accuracy rate; From the position of the highest test set accuracy, only the 7th epoch appears in the network using the SGD optimizer. The highest test set accuracy of the network using the other three optimizers is relatively high, indicating that theperformanceofthese three optimizers in the latter part of the trainingdecreased. It may be that the network has been over-fitting, and it is necessary to introduce evaluation and avoid over-fitting.

TABLE II. OPTIMIZERS TRAINING RESULTS

optimizer	Top Accuracy/%
SGD	87.350184
Adam	89.090000
Adamax	88.955900
RMSprop	88.676000

C. Comparison of the application of L2 regularization with the appropriate weight attenuation coefficient

In the face of possible over-fitting, one possible inhibitory measure is the introduction of regularization. We first use the L2 regularization and introduce the training set accuracy rate, training set loss, test set loss three indicators to enrich the evaluation results of the experimental results. The experimental design is shown in Table 3. The default optimizer is RMSprop (lr=0.001, alpha=0.9), the maximum iteration number is still set to 90epoch, and the L2 pooling corresponding weight attenuation coefficient (weight_decay) is the best and the best. The resulting position is shown in Table 3, and the corresponding accuracy and loss curves are shown in Figure 2-6.

It can be seen from Table 3 and Figure 4 that the training set loss curves show a smooth downward trend under the four values of the weight attenuation coefficient. Comparing the training set accuracy rate of Figure 3 with the test set accuracy rate of Figure 5, it can be seen that the accuracy rate under the corresponding weight attenuation coefficient is about 5% up and down, within an acceptable range, but the weight attenuation coefficient in Figure 6 is The loss curve of the test set at 0.001 is firstly decreased and then increased. This indicates that the over-fitting phenomenon appears under this parameter, which indicates that the improved LeNet-5 network proposed in this paper is attenuated by the weight of 0.001 when training on the SVHN dataset. The coefficient can not suppress the over-fitting, we should choose a higher weight attenuation coefficient; from Table 3, we can see that the data with the weight attenuation coefficient of 0.0025 is better than the weight attenuation coefficient of 0.005 and 0.01. On the accuracy curve of the test set in Figure 5, the curve with the weight attenuation coefficient of 0.0025 is higher than the curve with the weight attenuation coefficient of 0.005 and 0.01, and the weight attenuation coefficient is 0.0025 in the later stage of the training process. More obvious and less shocking. The above analysis shows that among the selected four sets of weight attenuation coefficients, the weight attenuation coefficient of 0.0025 can avoid over-fitting and achieve better training results.

TABLE III. RESULT OF DIFFERENT WEIGHT DECAY

weight_decay	Train Acc%(e)	Test Acc%(e)
0.01	89.01538(85)	87.14659(85)
0.005	91.59397(57)	88.95590(57)
0.0025	94.51470(88)	90.01229(88)
0.001	97.21119(90)	89.70498(24)

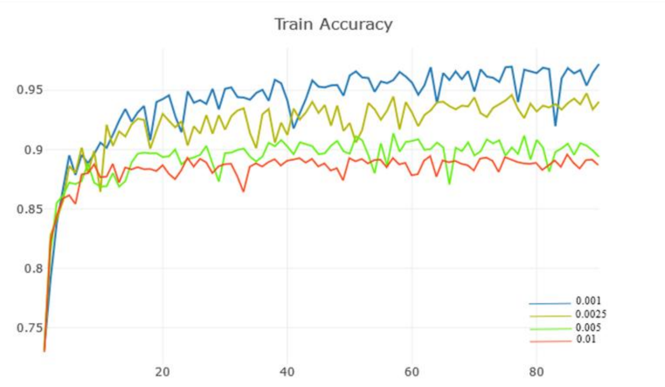

Figure 3. Training Accuracy of different regularization

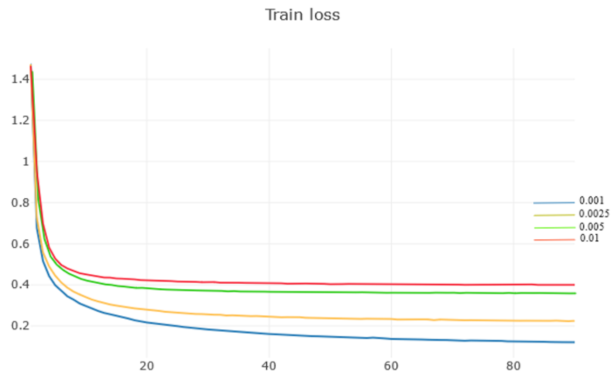

Figure 4. Training Loss of different regularization

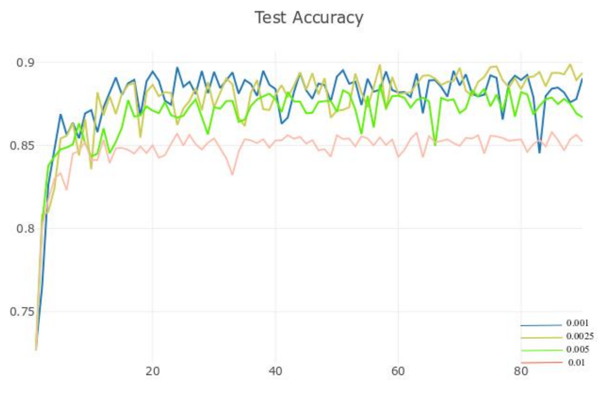

Figure 5. Test Accuracy of different regularization

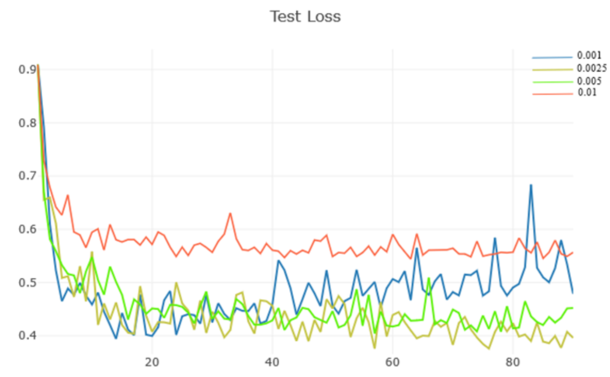

Figure 6. Tetst Loss of different regularization

III. EXPERIMENT AND ANALYSIS

A. *SVHN(The Street View House Numbers)*

Currently, for the identification task of the street view number, the better public data set is the SVHN data set. The SVHN dataset is a real-world image dataset focused on the development of machine learning and target detection algorithms with minimal need for data preprocessing and format conversion. There are ten types of labels in the dataset. Each class represents 1 number. For example, the category label of the number "1" is 1, and so on. The label of "9" is 9, and the label of "0" is 10. In general, the SVHN dataset contains three subsets: training set, test set, and extended set; the data set is divided into two formats based on the difficulty of recognition: a character-level bounding box containing the entire house number and a small number of wall backgrounds. The full resolution image (Figure 7); a 32x32 pixel centered on a single number similar to the MNIST dataset format image (Figure 8). The latter style is highly similar to the classic MNIST dataset, but is larger and more difficult to identify: the training set consists of 73,257 hard-to-recognize digital images, and the test set consists of 26,032 digital images, with an additional set of 531131. A simpler digital image that can be used to extend the test set. Unless otherwise stated, the SVHNdataset mentioned later in this article refers to the dataset in the format after cropping.

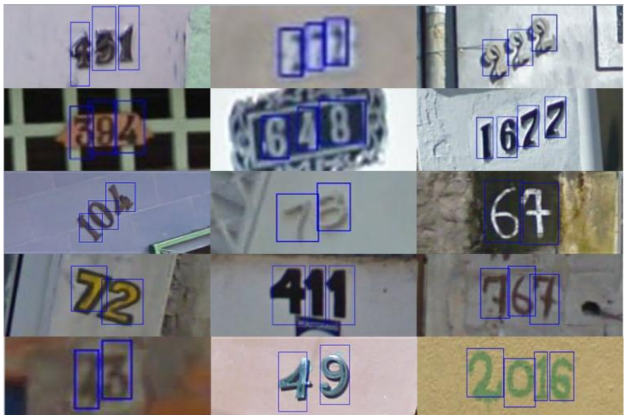

Figure 7. SVHN-Complete house number

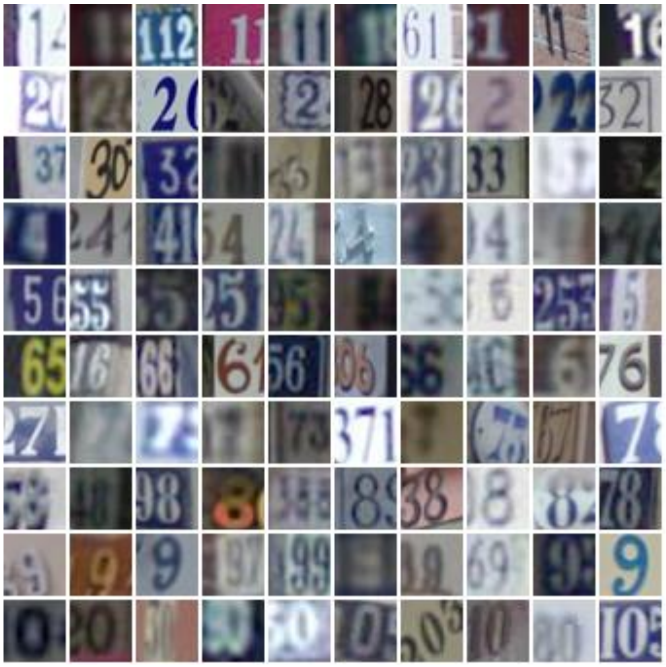

Figure 8. SVHN-Part number

B. *Data augmentation*

Augmenting the data set is also an effective means to improve the accuracy of the model. The size of each subset of the SVHN dataset is shown in Table 4, where the extension set official mentions that it can be used to extend the training data. Figure 9-11 shows a small number of samples and their labels in each subset. It can be seen that the resolution and brightness of the extended set are high and the background interference is small. The human eye recognition is indeed higher than the training set and test set, that is, It is said that the recognition of the extended set is relatively difficult, but the addition of the extended set can make the training set expand to 8.25 times, which is still expected to improve the accuracy of the model. Figure 12 shows the distribution of the original training set, extension set and test set ("1" for the number 1, "2" for the number 2, ..., "10" for the number 0), which can be seen in the three sub-categories The proportional distribution of each category is approximated, so the distribution of the new training set incorporating the extended set is similar to the distribution of the original training set. This correlation helps to suppress the disadvantages brought by the introduction of the extended set to the model training. influences.

TABLE IV. AUGMENTATION RESULT

Subset category	Number of samples
Training set	73257
Extra set	531131
Test set	26032

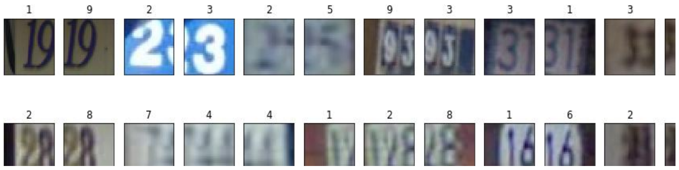

Figure 9. Example of train set

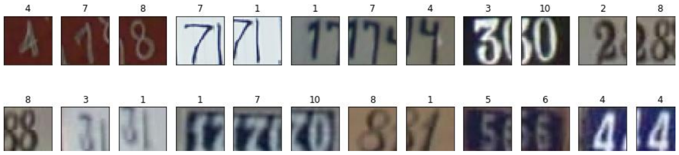

Figure 10. Example of extra set

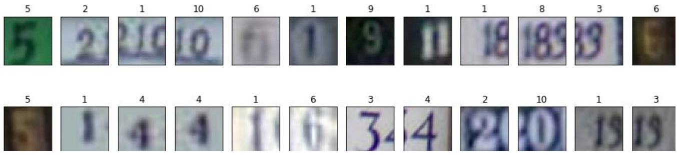

Figure 11. Example of test set

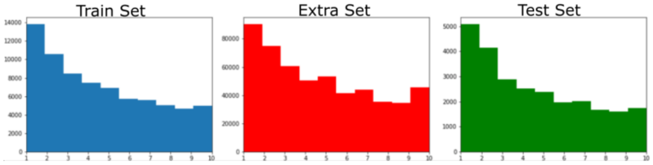

Figure 12. Category distribution of SVHN

The effect of 90 epoch training before and after the introduction of the extended set is shown in Table 5 and Figure 13. One point that needs to be specially stated is that the visualization tool Visdom we use has an automatic zoom function when drawing, which automatically hides the blank area of auto-hide the chart. Therefore, the different training sets in Figure 13 correspond to the vertical axis starting position and scaling of the chart. It's the same. The accuracy of the training set is lower than 95% and the loss curve of the test set shows a downward trend, indicating that there is no over-fitting phenomenon after expanding the data set; the accuracy of the test set is gradually reduced in the later period. The small description model tends to converge, and it can be seen that the model after the extended training set is higher than the extended training set. From Table 5, it can be seen that the training time after the expansion of the training set is 4 hours and 53 minutes, and the best accuracy rate is increased by 2.31254% compared with the expansion.

TABLE V. RESULT AFTER DATA AUGMENTATION

Train sample number	test sample number	Best test accuracy	time
73257	26032	90.01229	1h24min
604388	26032	92.32483	6h17min

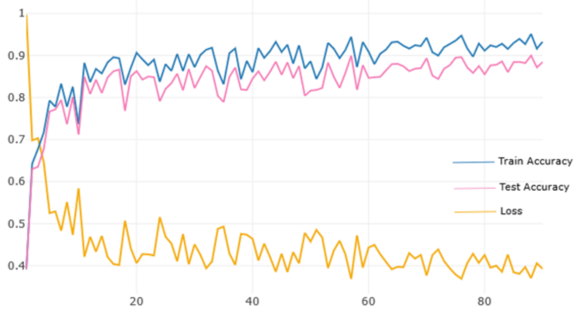

Figure 13. Training of the model after adding data augmentation

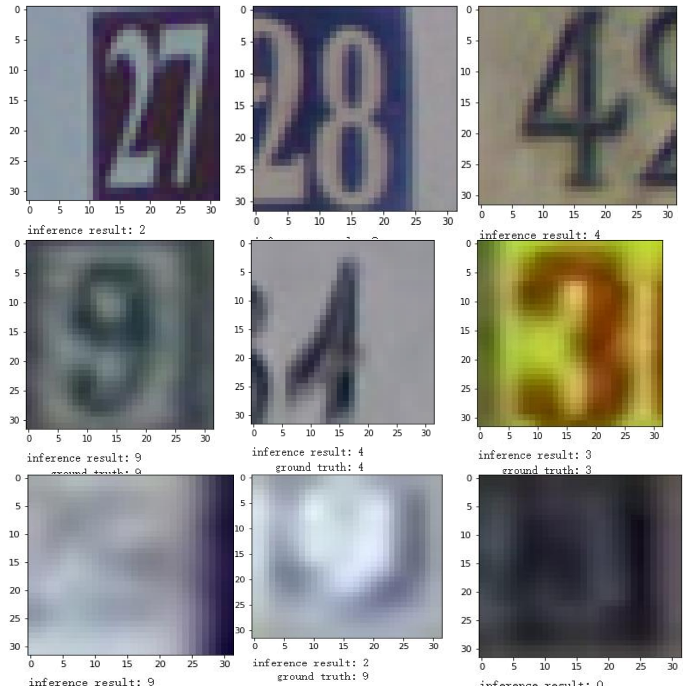

Figure 14. Figure 1 Test result

IV. CONCLUSION

The convolutional neural network applied in the SVHN dataset to improve the classic LeNet-5 network is: (1) modify the input layer to accept three-channel images; (2) switch to the more commonly used maximum pooling and Activation function, loss function; (3) introduction of gradient descent optimization algorithm RMSprop; (4) use L2 regularization. The seven-layer convolutional neural network implemented in this paper achieves direct processing of color pictures without complicated preprocessing, which improves the versatility of the model, speeds up the training and effectively avoids over-fitting. In the end, both the training speed and the prediction accuracy are better than the domestic Ma Miao and others based on the experimental results of the improved LeNet-5. After expanding the dataset, I tried to run a maximum of 170 epoch, and there was no obvious improvement in the test accuracy. Therefore, the future improvement direction should still be based on the principle. We can consider deepening the network level to obtain more abundant features.

REFERENCES

[1] Mori S, Suen C Y, Yamamoto K. Historical review of OCR research and development. Proceedings of the IEEE, 1992, 80(7): 1029-1058.

[2] Plamondon R, Srihari S N. Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(1): 63-84.

[3] De Campos T E, Babu B R, Varma M. Character recognition in natural images. VISAPP (2), 2009, 7.

[4] Yamaguchi T, Nakano Y, Maruyama M, et al. Digit classification on signboards for telephone number recognitio. IEEE, 2003: 359.

[5] ZHANG Shuai, SU Shi-tao. Doorplate Identification for Mobile Robot in Hallway Based on Morphological. Modern Electronics Technique, 2011, 34(14): 7-9+12.

[6] MA Li-ling. An algorithm based on ODLLTSA and SVM classier for door plate number recognition. Journal of Central South University (Science and Technology), 2011, 42: 789.

[7] ZHOU Cheng-wei. Recognition of Numbers in Natural Scene with Convolutional Neural Network.Computer Technology and Development, 2017.