IEICE TRANS. INF. & SYST., VOL.E103–D, NO.3 MARCH 2020

LETTER

Edge-SiamNet and Edge-TripleNet: New Deep Learning Models for Handwritten Numeral Recognition

Weiwei JIANG[†a)], ***Member and*** **Le ZHANG**[††b)], ***Nonmember***

SUMMARY Handwritten numeral recognition is a classical and important task in the computer vision area. We propose two novel deep learning models for this task, which combine the edge extraction method and Siamese/Triple network structures. We evaluate the models on seven handwritten numeral datasets and the results demonstrate both the simplicity and effectiveness of our models, comparing to baseline methods.

key words: *deep learning, handwritten numeral recognition, convolutional neural network*

1. Introduction

Handwritten numeral recognition has been a classical and important problem in computer vision, which has a wide application in different areas, e.g., recognition of the handwritten amount on bank checks and postal codes on postcards. Before the popularity of deep learning [1], "shadow" machine learning methods including random forest and support vector machine have been applied to this problem and have achieved some progresses [2], [3].

In 2012, a deep convolutional neural net [4] called AlexNet [5] achieved 16% error rate in the ImageNet Challenge [6]. Since then, deep learning methods have achieved a great success in different tasks, e.g., futures price trends forecasting [7], traffic forecasting [8], food recognition [9]. With deep learning models, the recognition accuracy of handwritten numeral has been greatly improved in the fast few years. Even so, this problem has not been fully solved, and the progress is achieved on some very limited datasets.

Some of the previous studies are trying to extend the evaluations of handwritten numeral recognition tasks from the Arabic language to other languages by building new datasets, e.g., Kannada-MNIST [10]. With these up-to-date datasets, we can propose and evaluate new models, as we do here.

In this study, we propose two novel deep learning models, inspired by the idea of extracting and incorporating edge information [11] and the structure of Siamese Network [12], for the problem of handwritten numeral recognition. For an overall evaluation, we test and compare our models on seven datasets with the same or similar data format of MNIST and release our code and trained models for reproducibility and further research*. The experimental results demonstrate both the simplicity and effectiveness of our models.

Our contributions are summarized below:

- We propose two simple but effective deep learning models for handwritten numeral recognition, with the replicable code.
- We introduce the idea of using Siamese/Triple network structures into the handwritten numeral recognition task.
- We evaluate our models on seven handwritten numeral datasets and achieve the state-of-the-art performance.

2. Related Work

Deep learning models, especially those based on convolutional neural networks, have been applied in the field of handwritten numeral recognition. Bhattacharya and Chaudhuri [13] investigated the use of wavelets and multilayer perceptron, in the identification of mixed Bangla, Devanagari and English numbers. Maitra et al. [14] used the classical LeNet-5 CNN model in the identification of five kinds of handwritten numbers, namely, Bangla, Devanagari, English, Oriya, and Telugu. For Bangla handwritten digit recognition only, auto-encoder and deep convolutional neural network are both explored [15].

3. Dataset

To fully evaluate the effectiveness of our models, we use seven handwritten datasets in different languages. They have the same or similar image formats with MNIST, which makes it easier for us to implement and compare the models. Unless otherwise stated, each input image has a resolution of 28×28, in gray-scale format, and corresponds to a number between 0 and 9. Specifically, the datasets used in this study include:

1. MNIST [4]: It is the classical handwritten numeral dataset, which consists of 60,000 training samples and 10,000 testing samples. It has been used extensively in the research area of handwritten numeral recognition as an widely-accepted dataset for comparison.
2. Tibetan-MNIST**: It is the first open dataset of Tibetan

Manuscript received November 7, 2019.
Manuscript publicized December 9, 2019.

[†]The author is with Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
[††]The author is with School of Computer Science and Information Engineering, Hubei University, Wuhan 430000, China.
a) E-mail: jiangweiwei@mail.tsinghua.edu.cn
b) E-mail: 201622111920039@student.hubu.edu.cn

DOI: 10.1587/transinf.2019EDL8199

*https://github.com/jwwthu/Edge-SiamNet
**https://www.kesci.com/home/dataset/5bfe734a954d6e0010683839

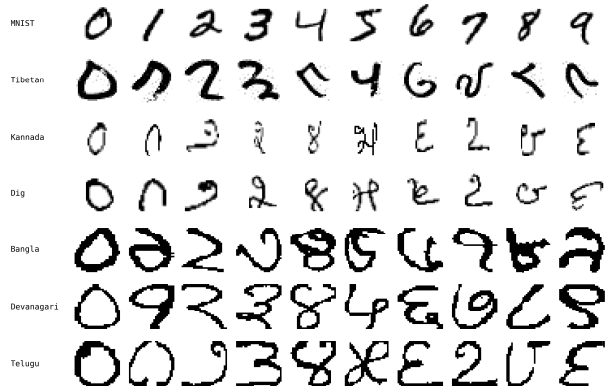

Fig. 1 The numbers in different languages.

handwritten numerals and has not been used in previous studies. We transform the RGB format to the grayscale format and resize the original data from 32×32 to 28×28, to keep consistent with the other datasets. We manually divide the dataset into a training set with 14,214 samples and a testing set with 3,554 samples, while preserving the percentage of samples for each class during the splitting process.

3. Kannada-MNIST and Dig-MNIST [10]: Kannada-MNIST is a handwritten digits dataset for the Kannada language, which consists of 60,000 training samples and 10,000 testing samples. Dig-MNIST is an out-of-domain test dataset for Kannada-MNIST, which contains 10,240 testing samples and acts as an excellent test set for the generality of recognition models.
4. Bangla, Devanagari, and Telugu [16]: These datasets come from the CMATERdb database, which is a pattern recognition database created by the Cmater Research Laboratory for Training Education and Research in Jadavpur University, India. They are all languages used in some areas of India. We would use 6,000 Bangla numerals, 3,000 Devanagari numerals, and 3,000 Telugu numerals in this study and divide the datasets into training and testing sets with a ratio of 80% : 20%.

For a better illustration of these datasets, we show the samples of the numbers in different languages in Fig. 1. All the other datasets except Tibetan-MNIST are balanced, which means each digit has the same number of samples.

4. Proposed Models

In this study, we propose two novel models, namely, Edge-SiamNet and Edge-TripleNet, for handwritten numeral recognition. As indicated in their names, we firstly extract the edge images by the canny edge extraction method. An example of the input images and extracted edge images from 0 to 9 is shown in Fig. 2. Instead of using the edge features as residual connection as in [11], we propose to use a Siamese/Triple network structure to extract features from both the original input image and the edge image. Without

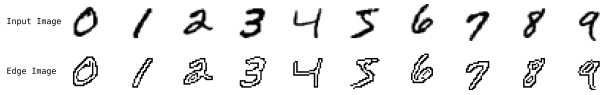

Fig. 2 The input images and the extracted edge images.

introducing more parameters than EdgeNet, our approach improves the network's performance of recognizing different numbers.

The network structures of Edge-SiamNet and Edge-TripleNet are shown in Fig. 3. In both Edge-SiamNet and Edge-TripleNet, we use the similar convolutional unit, which consists of a kernel size of 3×3, stride size of 1, activated with a ReLu function and followed by a dropout ratio of 25%. In Edge-SiamNet, the weights of three convolutional layers, namely, Conv1, Conv2, and Conv3 in Fig. 3, are shared, as shown by the dashed line with arrows. This Siamese network structure helps us to learn features from both the input and edge images with the same mechanism. In Edge-TripleNet, we further add the outputs of the first convolutional layers (iConv and eConv) and share the weights of Conv1, Conv2, and Conv3 by three times. Both Conv2 and Conv3 have a dilation rate of 2 and are used to extract the high-level features. After the Siamese and Triple network structures, the outputs are added and further processed by Conv4. Then the pooling and flatten operations are conducted. The softmax classifier consists of two fully connected layers and the dropout rate is also 25% in FC1. For the final layer F2, it has a output dimension of 10, because we have 10 classes in numeral recognition.

5. Experiment

5.1 Model Implementation

We use the Canny function provided by OpenCV[†] to extract edge images. The parameter setting used in the edge extraction process is consistent with the previous studies [11]. Specifically, we use 100 as the first threshold for the hysteresis procedure, 200 as second threshold for the hysteresis procedure, and 3 as the aperture size for the Sobel operator.

To build the neural networks, we use the Python packages Keras[††] and Tensorflow[†††]. To compare with the models we propose, we also implement many standard classification models as baselines, including EdgeNet [11], a convolutional neural network (CNN)[††††], LeNet [4], MobileNet [17], VGG16 [18], AlexNet [5], and ResNet50 [19]. Among all the networks, EdgeNet, Edge-SiamNet, and Edge-TripleNet have the smallest number of parameters as 836,938. The remaining parameter numbers are 1,199,882 for CNN, 1,256,080 for LeNet, 3,238,538 for MobileNet, 14,718,666 for VGG16, 21,598,922 for AlexNet,

[†]https://github.com/skvark/opencv-python
[††]https://keras.io/
[†††]https://www.tensorflow.org/
[††††]https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py

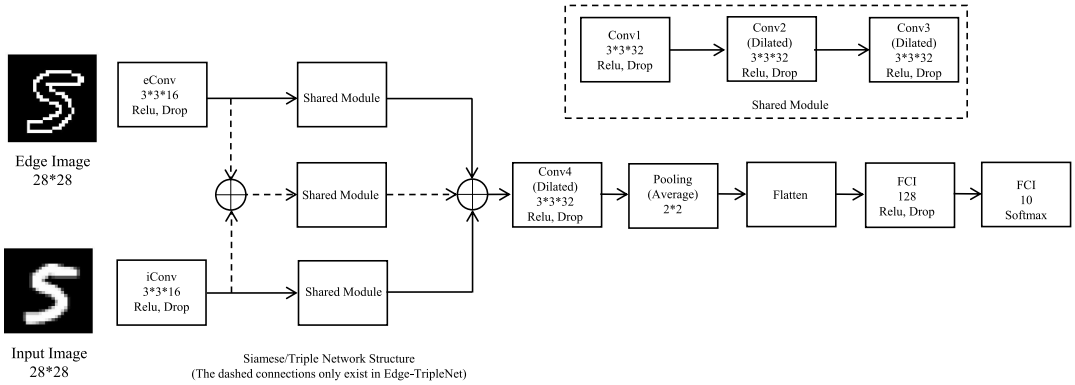

Fig. 3 The network structures of Edge-SiamNet and Edge-TripleNet.

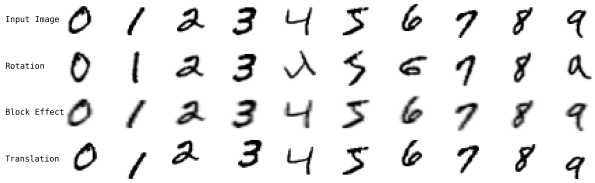

Fig. 4 The illustration of the data augmentation techniques with numbers of MNIST.

23,601,930 for ResNet50 in our implementations.

The model is trained and tested on a computer with Windows 10 OS, which has a hardware configuration of 16GB random-access memory (RAM) and Intel core i5-9600K central processing unit (CPU). We use a graphical accelerated processing (GPU) of GeForce RTX 2070 with 8GB RAM to accelerate the convolution operations used in the models.

5.2 Data Augmentation

To further improve the models' performance, we use the same data augmentation techniques as in [11]. The benefits of data augmentation include adding data diversity and avoiding overfitting. With the three data augmentation techniques being applied, the training set is multiplied by three times in size.

- Rotation: The original image is randomly rotated between −45 to 45 degrees.
- Block Effect: The original image is resized into 14×14 dimension to loss some information and then is resized back into 28×28.
- Translation: A translation matrix, which is randomly selected between −5 to +5 pixels, is applied to the original image.

The result of applying these data augmentation techniques for numbers in MNIST is shown in Fig. 4. The data augmentation techniques are also implemented with OpenCV. The implementation details may refer to our public code.

5.3 Evaluation

To choose the best model, we further divide the training set into training and validation sets with a ratio of 80% : 20%. During the training process, we use a batch size of 128 and run each model for 100 epochs. The Adadelta [20] optimizer with a learning rate of 0.001 is chosen to minimize the categorical cross entry losses. Then the model weight with the best validation accuracy is saved and evaluated in the testing set. As a specific case, for the evaluation of Dig-MNIST, we use the best model trained and validated from Kannada-MNIST directly without any further revision.

We use accuracy as our main evaluation metric. For Tibetan-MNIST, we use a balanced accuracy, so that all classes have equal weights. The result is shown in Table 1, with a highlight of the best results in both cases of with and without data augmentation.

By comparing the performances between different models, Edge-SiamNet and Edge-TripleNet achieve a comparable performance with some of the state-of-the-art models, by using a much less parameters (*e.g.*, 836,938 vs VGG16's 14,718,666 and ResNet50's 23,601,930), even though our models are not the best in every single dataset. The simple structure of our models make them still attractive, especially on mobile devices with limited resources.

We also observe that data augmentation techniques help to improve the out-of-sample accuracy for all models. The data augmentation techniques also narrow down the gap between our models and other models, *e.g.*, VGG16 and ResNet50. With data augmentation, Edge-TripleNet and VGG16 both achieve the best accuracy on 4 out of 7 datasets.

The result of 99.25% on Tibetan-MNIST gives a competitive baseline for this dataset, which has not been covered in previous studies. The result of 87.45% on Dig-MNIST also achieves the best result for this dataset in the literature and significantly increases the result of 76.1% in [10].

6. Conclusion

In this study, we propose two novel deep learning models

Table 1 Comparison between different models.

Method \ Data		MNIST	Tibetan-MNIST	Kannada-MNIST	Dig-MNIST	Bangla	Devanagari	Telugu
Without data augmentation	Edge-SiamNet	99.36	**98.86**	97.37	77.14	97.42	96.17	**98.33**
	Edge-TripleNet	99.26	98.60	97.21	77.10	97.58	**96.33**	**98.33**
	EdgeNet	98.96	98.18	97.34	76.59	95.58	93.00	97.33
	CNN	99.16	98.79	97.42	78.88	97.00	95.50	97.67
	LeNet	99.26	**98.86**	97.36	75.28	97.25	95.33	**98.33**
	MobileNet	99.21	98.66	97.22	79.96	96.50	93.67	96.33
	VGG16	**99.47**	98.82	97.76	**81.20**	**97.83**	96.17	97.50
	AlexNet	99.25	98.66	97.34	75.63	96.33	95.67	97.00
	ResNet50	99.44	**98.86**	**97.83**	78.85	96.25	94.83	**98.33**
With data augmentation	Edge-SiamNet	**99.51**	99.20	**98.24**	85.50	98.17	96.83	**99.33**
	Edge-TripleNet	99.46	**99.25**	97.97	85.97	**98.58**	**97.33**	**99.33**
	EdgeNet	99.42	98.89	98.13	85.95	97.50	96.17	98.33
	CNN	99.41	99.08	97.44	82.96	98.00	96.67	99.17
	LeNet	99.40	99.18	97.35	80.83	98.17	96.83	98.83
	MobileNet	99.40	99.17	97.80	85.09	97.50	96.67	98.83
	VGG16	99.50	99.09	98.17	**87.45**	**98.58**	**97.33**	**99.33**
	AlexNet	99.37	99.07	97.48	84.28	97.25	97.00	98.50
	ResNet50	99.50	99.01	98.01	87.32	98.50	96.83	99.17

for handwritten numeral recognition. We validate the models with seven handwritten numeral datasets, including some of the latest ones. The experimental results show both the simplicity and effectiveness of our models. And our models achieve the best accuracy on the latest dataset. The release of our code is convenient for deployment and comparison with our models in the further research.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol.521, no.7553, pp.436–444, 2015.

[2] T. Hassan and H.A. Khan, "Handwritten Bangla numeral recognition using local binary pattern," 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), pp.1–4, May 2015.

[3] R. Sarkhel, N. Das, A.K. Saha, and M. Nasipuri, "A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition," Pattern Recognition, vol.58, pp.172–189, 2016.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol.86, no.11, pp.2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25, ed. F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, pp.1097–1105, Curran Associates, Inc., 2012.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp.248–255, 2009.

[7] W. Lu, C. Geng, and D. Yu, "A new method for futures price trends forecasting based on BPNN and structuring data," IEICE Trans. Inf. & Syst,, vol.E102-D, no.9, pp.1882–1886, Sept. 2019.

[8] W. Jiang and L. Zhang, "Geospatial data to images: A deep-learning framework for traffic forecasting," Tsinghua Science and Technology, vol.24, no.1, pp.52–64, 2019.

[9] M. Anzawa, S. Amano, Y. Yamakata, K. Motonaga, A. Kamei, and K. Aizawa, "Recognition of multiple food items in a single photo for use in a buffet-style restaurant," IEICE Trans. Inf. & Syst., vol.E102-D, no.2, pp.410–414, Feb. 2019.

[10] V.U. Prabhu, "Kannada-MNIST: A new handwritten digits dataset for the Kannada language," arXiv preprint arXiv:1908.01242, 2019.

[11] S. Sharif, G. Mujtaba, and S. Uddin, "Edgenet: A novel approach for arabic numeral classification," arXiv preprint arXiv:1908.02254, 2019.

[12] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," Advances in Neural Information Processing Systems, pp.737–744, 1994.

[13] U. Bhattacharya and B.B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.3, pp.444–457, 2009.

[14] D.S. Maitra, U. Bhattacharya, and S.K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts," 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp.1021–1025, 2015.

[15] M. Shopon, N. Mohammed, and M.A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," 2016 International Workshop on Computational Intelligence (IWCI), pp.64–68, 2016.

[16] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "A benchmark image database of isolated Bangla handwritten compound characters," International Journal on Document Analysis and Recognition (IJDAR), vol.17, no.4, pp.413–431, 2014.

[17] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Computer Science, 2014.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, June 2016.

[20] M.D. Zeiler, "Adadelta: An adaptive learning rate method," arXiv preprint arXiv:1212.5701, 2012.