Evaluation of deep learning models for Urdu handwritten characters recognition

Weiwei Jiang1*

1 Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
*Corresponding author’s e-mail: jiangweiwei@mail.tsinghua.edu.cn

Abstract. As a classical and significant problem, handwritten character recognition has been widely used in our daily lives. With recent deep learning methods, previous studies have achieved a great improvement for this problem in the past few years. However, the handwritten character recognition for Urdu, which is one of the largest languages of the world, is less studied in the existing literature. In this paper, we fill in this gap and evaluate different deep learning models on the problem of Urdu handwritten characters recognition based on a newly released dataset. Combined with data augmentation and transfer learning techniques, we achieve the state-of-the-art results by recognizing digits and characters with an accuracy of 98.94% and 99.08%, respectively, which greatly improves the baselines of 97% and 86.5%.

1. Introduction

With the wide application of computers and mobile devices, character recognition systems have been universally used in our daily lives. The functionality of handwritten character recognition is intuitive, with a digitized image as the input and the desired character as the output. The usage of automatic character recognition systems would improve the efficiency compared with human workers, e.g., recognizing the numbers in the bank check and postcards. However, it is not as simple as today to achieve a reliable recognition performance.

To build and test different recognition models fairly, a standard handwritten digit dataset named MNIST [1] was proposed back in 1989. MNIST contains 60,000 and 10,000 images for training and testing. Each image is centered in a 28×28 grey-scale data format. It also becomes the standard data format for many following character datasets. MNIST soon became the benchmark dataset for handwritten digit recognition. Afterwards, more handwritten character datasets appear for research purposes, with different languages and more characters.

Based on these datasets, different methods have been explored for handwritten character recognition tasks. Various machine learning models have been applied, e.g., k-nearest neighbours (kNN) [2] and support vector machines (SVM) [3]. In 2012, a deep convolutional neural net named AlexNet [4] managed to achieve a 16% error rate in the ImageNet Challenge [5]. Afterwards, deep learning models have become the dominant approach in different areas, e.g., traffic forecasting [6], time series classification [7], stock prediction [8], etc. The application of deep learning models greatly improves the performance of handwritten character recognition and make it reliable in practice.

Urdu is one of the largest languages of the world and the national language of Pakistan, being used by more than 60 million people [9]. However, compared to the massive studies for recognition of handwritten English characters and other languages, Urdu handwritten characters recognition is less discussed in current literature. Urdu text recognition is also more challenging due to the presence of diacritics. With a newly released pioneer dataset for handwritten digits and characters of Urdu which
contains samples from more than 900 individuals, we have the chance to evaluate the performances of different deep learning models, as well as the data augmentation and transfer learning techniques. We manage to improve the recognition accuracy of digits to 98.94% and the recognition accuracy of characters to 99.08%, which is a great improvement from the baselines of 97% and 86.5%, respectively. We present our exploration process and findings in this paper.

Our contributions in this paper are summarized as follows:
1. We give a systematic evaluation of 12 different deep learning models for Urdu handwritten characters recognition.
2. We find that sophisticated models, which are prone to overfitting, may not outperform simpler models.
3. We demonstrate that data augmentation and transfer learning techniques can improve the models’ performance for Urdu handwritten characters recognition.
4. We manage to recognize digits and characters with an accuracy of 98.94% and 99.08%, which greatly improves the baselines of 97% and 86.5% in [9].

The rest of this paper is organized as follows. Section 2 is the related work. Section 3 describes the dataset we use in this study. Section 4 presents the deep learning models we use in this study. Section 5 presents the experiments and the associated result analysis. We conclude this paper in Section 6.

2. Related Work

In this part, we give a short review of the related work, covering the handwritten character recognition models both in the machine-learning and deep-learning approaches.

2.1. Machine Learning Approach

To differ from deep neural networks, machine learning models are "shallow" models that try to learn patterns from data, instead of handcrafted rules. Different families of machine learning models, e.g., kNN and SVM, have been used for handwritten character recognition in previous studies. A k-nearest neighbor classifier that uses a feature vector combining both structural and statistical features of MNIST achieves a 98.42% accuracy in [2]. To further improve the performance of a single SVM classifier, multiple SVM classifiers are combined in [3], based on the NIST (National Institute of Standards and Technology) handwritten digit database.

2.2. Deep Learning Approach

The success of deep learning models applies to handwritten character recognition tasks and the strong learning ability enables them to deal with larger and more complex problems, e.g., handwritten character recognition problems in multiple languages and with more classes. As a feed-forward neural network, a multi-layer perceptron is used for the case mixed Bangla, Devanagari and English numbers recognition in [10]. For Bangla handwritten digit recognition, a pre-trained auto-encoder and deep convolutional neural networks are used in [11]. Using the edge feature, two deep learning models that use the structure of Siamese network are proposed and evaluated on 7 handwritten digit datasets in [12].

3. Dataset Description

In this paper, we would use a pioneer dataset of Urdu handwritten characters [9]. This dataset fills in the gap of a standard dataset for this task and introduces a more diverse range of writing styles, compared to other Urdu-related datasets, e.g., the UCOM dataset [13]. The dataset consists of 10 digits and 40 characters and is collected from 900 different individuals with different ages from 22 to 60 years and goes through various pre-processing steps, e.g., RGB to grey-scale conversion, noise removal and segmentation.

After pre-processing, the images are transformed into a data format of 28×28 grey-scale images. And they are further divided into training and test sets. Specifically, for digits, the training set has 6,606 samples and the test set has 1,414 samples. And for characters, the training set has 28,328
samples and the test set has 4,880 samples. The examples of different digits and characters are shown in Figure 1 and Figure 2, respectively.

Figure 1. The digit examples of the Urdu dataset.

Figure 2. The character examples of the Urdu dataset.

4. Models

4.1. Deep Learning Models
In this paper, we would use both sophisticated and simple deep learning models and compare their performance for Urdu handwritten character recognition. The simple model we use is a benchmark model convolutional neural network (CNN). This benchmark CNN model consists of four convolutional layers and standard max pooling operation is added after the second and the fourth convolutional layer. The output of the fourth convolutional layer is flattened and fed into a following densely connected layer. The final output layer has a dimension that corresponds to the class numbers, i.e., 10 for digits and 40 for characters. For some of the models have minimum requirements for the input shape, we resize the input images to satisfy those requirements.

Specifically, we use the following sophisticated models: LeNet [1], AlexNet [4], VGG [14], DenseNet [15], Xception [16], ResNet [17], Inception [18], and MobileNet [19].

4.2. Data Augmentation
To achieve a satisfactory performance, deep learning models usually require a large dataset for training. However, it is expensive and time-consuming to collect high-quality images in practice. Data augmentation appears as an alternative approach that tries to increase the training set with some image transformation techniques. It has been proved to bring improvements for image classification and object detection tasks.

To validate whether data augmentation techniques are useful for Urdu handwritten characters recognition, we use three data augmentation techniques as in [12].

1. Rotation: With the center fixed, the image is randomly rotated, within a range of -45 to +45 degrees.
2. Block Effect: To loss some information, the image is firstly resized into 14×14. Then to resume the original size, the image is resized back to 28×28.
3. Translation: The image is applied with a translation matrix that is randomly selected between -5 to +5 pixels.

The original images as well as the generated images after applying these data augmentation techniques on the digits are shown in Figure 3.
4.3. Transfer Learning
Transfer learning is another option when dealing with the small training set, when the pre-trained parameters from a larger training dataset, e.g., the ImageNet Challenge [5], can be used and fine-tuned on the new cases. In this paper, we use the pre-trained models on ImageNet of 9 deep learning models and further evaluate two cases with and without data augmentation techniques.

4.4. Evaluation Metrics
Since the Urdu handwritten text recognition problem is a classification problem, we use accuracy as our major evaluation metrics, which is defined as the percentage of test samples that are correctly classified. Formally, accuracy can be defined as follows:

$$\text{Accuracy} = \frac{TP}{TP + TN}$$

where TP means true positive which is the number of samples correctly identified and TN means true negative which is the number of samples incorrectly classified by the model.

5. Experiments
We use Python and its packages of Keras and TensorFlow for the implementation of the deep learning models. Another package named OpenCV is used to implement the data augmentation techniques. During the training phase, Adam [20] is used as the optimizer for all the models. Adam can be regarded as a combination of RMSprop and stochastic gradient descent with momentum. Compared with other optimizers, Adam has several advantages, e.g., more efficient computation and a lower memory requirement. We use batch training, and a batch size of 128 samples is used for a total 100 epochs. The model weights with the best validation accuracy are chosen for evaluation on the test set. And to choose these best weights, we further divide the original training set into training and validation sets with a ratio of 4:1, while preserving the percentage of samples for each class during the splitting process. During all experiments, we use GeForce RTX 2070 with 8GB RAM as our graphical accelerated processing (GPU), for the acceleration of the convolution operations and reduce the training time of deep learning models.

We show the results of applying different deep learning models with/without data augmentation techniques and with/without pre-trained models in Table 1. Our first observation is that after a proper training, both simple and sophisticated models work well for Urdu handwritten digits/characters recognition, even though the recognition for characters with 40 classes is more challenging than digits with only 10 classes, as most of deep learning models show a lower recognition accuracy for characters than digits. Our second observation is the verification of the performance improvement brought by the data augmentation techniques. And the improvement is minor for those deep learning models which already perform very well without data augmentation techniques and there is little space for further improvement.
Table 1. Accuracy lists of different deep learning models.

| Models     | Without Transfer Learning | With Transfer Learning |
|------------|---------------------------|------------------------|
|            | Without Data Augmentation | With Data Augmentation | Without data Augmentation | With Data Augmentation |
|            | Digits | Characters | Digits | Characters | Digits | Characters | Digits | Characters |
| CNN        | 0.9767 | 0.9418     | 0.9830 | 0.9605     | Not covered |
| LeNet      | 0.9731 | 0.9545     | 0.9781 | 0.9641     | Not covered |
| AlexNet    | 0.9795 | 0.9561     | 0.9802 | 0.9686     | Not covered |
| VGG16      | 0.9809 | 0.9723     | 0.9851 | 0.9844     | 0.9851 | 0.9840 | 0.9867 | 0.9885 |
| VGG19      | 0.9788 | 0.9795     | 0.9809 | 0.9848     | 0.9844 | 0.9830 | 0.9880 | 0.9865 |
| DenseNet169| 0.9837 | 0.9705     | 0.9880 | 0.9801     | 0.9873 | 0.9869 | 0.9880 | 0.9877 |
| DenseNet201| 0.9809 | 0.9723     | 0.9851 | 0.9844     | 0.9851 | 0.9840 | 0.9867 | 0.9885 |
| Xception   | 0.9816 | 0.9619     | 0.9851 | 0.9762     | 0.9837 | 0.9848 | 0.9873 | 0.9900 |
| ResNet50   | 0.9823 | 0.9852     | 0.9837 | 0.9865     | 0.9859 | 0.9887 | 0.9866 | 0.9893 |
| MobileNet  | 0.9604 | 0.9742     | 0.9632 | 0.9779     | 0.9851 | 0.9891 | 0.9887 | 0.9908 |

A third observation after comparison is the verification of the performance improvement brought by the transfer learning technique. And most of the deep learning models achieve their best performance by using data augmentation and pre-trained weights simultaneously. The best accuracy of 98.94% for digits is achieved by Xception, and the best accuracy of 99.08% for characters is achieved by MobileNet. This result is a little surprising since MobileNet is a light weight deep neural network designed for mobile devices, with a simpler structure than many other deep learning models we use too. This could be explained that even though deeper and larger neural networks have a stronger learning ability, they are also more prone to problems such as overfitting, which conversely limit their final performance.

6. Conclusion
In this study, we give an exploration for different deep learning models on the problem of Urdu handwritten characters recognition. We also leverage data augmentation and transfer learning techniques to further improve the recognition accuracy. We find that sophisticated models may not outperform simpler models for this problem. For the newly released Urdu dataset, we contribute the state-of-the-art results by recognizing digits and characters with an accuracy of 98.94% and 99.08%, respectively.

References
[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[2] Babu, U. R., Chinthala, A. K., & Venkateswarlu, Y. (2014). Handwritten digit recognition using structural, statistical features and k-nearest neighbor classifier. International Journal Information Engineering and Electronic Business, 6(1), 62-68.
[3] Gorgevik, D., & Cakmakov, D. (2005, November). Handwritten digit recognition by combining SVM classifiers. In EUROCON 2005-The International Conference on" Computer as a Tool" (Vol. 2, pp. 1393-1396). IEEE.
[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[5] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[6] Jiang, W., & Zhang, L. (2018). Geospatial data to images: A deep-learning framework for traffic forecasting. Tsinghua Science and Technology, 24(1), 52-64.

[7] Jiang, W. (2020). Time series classification: nearest neighbor vs deep learning models. SN Applied Sciences.

[8] Jiang, W. (2020). Applications of deep learning in stock market prediction: recent progress. arXiv preprint arXiv:2003.01859.

[9] Ali, H., Ullah, A., Iqbal, T., & Khattak, S. (2020). Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network. SN Applied Sciences, 2(2), 152.

[10] Bhattacharya, U., & Chaudhuri, B. B. (2008). Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. IEEE transactions on pattern analysis and machine intelligence, 31(3), 444-457.

[11] Shopon, M., Mohammed, N., & Abedin, M. A. (2016, December). Bangla handwritten digit recognition using autoencoder and deep convolutional neural network. In 2016 International Workshop on Computational Intelligence (IWCI) (pp. 64-68). IEEE.

[12] Jiang, W., & Zhang, L. (2020). Edge-SiamNet and Edge-TripleNet: New Deep Learning Models for Handwritten Numeral Recognition. IEICE Transactions on Information and Systems, 103(3), 720-723.

[13] Ahmed, S. B., Naz, S., Swati, S., Razzak, I., Umar, A. I., & Khan, A. A. (2017). UCOM Offline Dataset-An Urdu Handwritten Dataset Generation. International Arab Journal of Information Technology (IAJIT), 14(2).

[14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[15] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[16] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[18] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[19] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[20] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.