https://doi.org/10.36785/BUITEMS.JAES.338

Pashtu Numerals Recognition through Convolutional Neural Networks

Khalil Khan, Bilawal Shah, Muhammad Waheed, Adeel Shams, Daniyal Munir
Department of Electrical Engineering, University of Azad Jammu and Kashmir, Pakistan
Correspondence: khalil.khan@ajku.edu.pk

Abstract--**In the proposed paper we introduce a new Pashtu numerals dataset having handwritten scanned images. We make the dataset publically available for scientific and research use. Pashtu language is used by more than fifty million people both for oral and written communication, but still no efforts are devoted to the Optical Character Recognition (OCR) system for Pashtu language. We introduce a new method for handwritten numerals recognition of Pashtu language through the deep learning based models. We use convolutional neural networks (CNNs) both for features extraction and classification tasks. We assess the performance of the proposed CNNs based model and obtained recognition accuracy of 91.45%.**

Keywords-**Optical character recognition, Convolutional neural networks, Pashtu numerals recognition**

Date Received 04-12-2019
Date Accepted 25-12-2019
Date Published 01-01-2020

I. Introduction

WITH the advancement of information technology, online and offline usage of digital text is increasing day by day. A system that converts machine written and handwritten scanned images to editable form is called Optical Character Recognition (OCR). OCR is the most researched area in computer vision and pattern recognition. OCR for most of the languages got a very mature position in the last 15 years [1-3]. These text images primary sources are scanned documents, images from scenes, and broadcasted videos.

The very first OCR system was investigated 65 years ago [4]. Since then, efforts are made by researchers, which leads to a nearly complete OCR system for advanced languages of the world. Although very large scale deployments have been made for OCR systems of non-cursive scripts languages. But a mature OCR for cursive scripts languages is still a challenging task.

Pashtu is spoken by around fifty million people around the globe [55]. Pashtu is a national language of Afghanistan and spoken in most part of Pakistan as well. This language has rich literature and diversity. There is a large written material in Pashtu, which addresses very diverse topics such as education, politics, religion, poetry, and much more. Instead of all these, Pashtu still needs a mature OCR system.

Pashtu OCR (POCR) is far away due to certain major problems this language is facing. For example, it is a cursive language written from right to left-hand side. Very little variation occurs in characters' shape for non-cursive script languages. Unlike non-cursive script languages, characters in the Pashtu language have significant variations. Similarly, different formation rules are very complicated for Pashtu [6, 7]. In cursive script languages, when various characters combine, an intermediate shape is obtained called ligature. These ligatures are missing in non-cursive script languages, which further make OCR system complex.

We have a long term research strategy, which will lead to a complete OCR system for Pashtu. As no standard database exists for OCR development, as an initial step we make the first Pashtu numerals database, we called PHND V-0. We also introduce a CNNs based recognition algorithm for these numerals.

In a nutshell, our current research work has the following contributions:

- Introducing a new database for numerals recognition of Pashtu language.
- Introducing a CNNs based model for the recognition of the numerals of Pashtu text.

II. RELATED WORK

OCR has been addressed through two prominent methods previously, including holistic and analytical methods. We discuss both of these methods as we proceed. Holistic methods have no specific typography rules. These methods are generic, as can be applied to any language. An image having text is considered as one dimension vector, and features are extracted from the image. No segmentation is needed for such kind of methods. One of the main drawbacks of these methods is the requirement of a large amount of training data. These algorithms are robust to scale and changes in rotation. Moreover, a rich set of features are needed for building a model.

A popular OCR system based on holistic methods is BBN Byblos OCR system [12]. Multiple languages have been tested on this system. For synthetic data, a very low error rate has been reported with these methods. These methods fail to perform when applied to a comparatively larger database, as very little training data has been used in the development stage.

For Pashtu text, a method developed on the holistic algorithm is reported in [13]. The authors of the paper used Noori Nastaliq script of the language during this work. This OCR system was evaluated on the synthetic database. Some methods developed for OCR can be explored in the references [14-19].

The second class of methods is analytical methods, which are advanced methods and are constructed through specific grammatical rules for the respective language. A unique set of features are used to identify a character. Segmentation at atomic level is performed for these methods. The performance of these methods is better when results of the prior segmentation is easy. For non-cursive script languages boundary of a character can easily be located; hence results are much better. For getting acceptable performance for these algorithms, better segmentation is mandatory, which is itself a big challenge in analytical methods. For the Pashtu language, still, no algorithm has been developed, which is based on analytical methods.

Some methods which are based on Hidden Markov Models and Neural Networks are reported in [8, 9] for other cursive script languages. Some excellent papers have also been published for cursive script languages in ICDAR [10, 11].

A database for Pashtu ligatures is also reported in [21]. Authors of the paper used Recurrent Neural Networks to develop a Pashtu OCR. Tests are performed on a limited set of images in [21]. Authors named their introduced database KPTI. The KPTI consists of 17, 015 images of Pashtu text. To the best of our knowledge, this [21] is the best research work reported particularly for Pashtu language. Some other works which used deep learning-based methods for cursive script languages can be explored in references [22-25].

III. PASHTU NUMERALS DATABASE

A main drawback of the deep learning-based method is the requirement of large training data. In this paper, we introduce a first handwritten database for Pashtu numerals. The database is freely available for research and can be provided upon request.

We collected data from different regions to bring diversity in writing style. We collected these images from faculty members, staff, and students of three universities, namely, the University of Azad Jammu and Kashmir, University of Malakand, and the University of Peshawar. The total participants in the data collection were 750. Every participant wrote each digit four times. A form was distributed among the participants to write Pashtu digits with hands.

The age range of all the participants was between 18-60 years. We scanned the written form with a 300 dpi resolution and then did some pre-processing step as described below;

- We corrected the inclination of each page with a horizontal histogram.
- We detected the center of each numeral. For the localization of center we used a connected component algorithm.
- We extracted each numeral from the image and then rescaled to 30 × 30.
- We converted all images to binary form after all the above-mentioned steps. An image is shown in Figure 1, where a complete folder from the database is shown for one single participant.

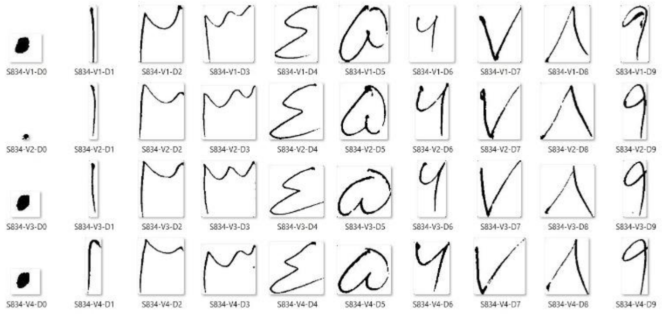

Fig. 1. One participants hand written numerals in the folder.

Each participant handwritten images are in one folder. One folder of a single candidate can be seen in Figure 1.

Each name has three parts i.e., S. D, and V. The alphabet S represents subject number which is in the range 1-750, D shows digit number and V represent version of writing which is from 0-4. We collected the written forms from 750 participants. The database can be freely downloaded for research use.

IV. PROPOSED METHOD

The details of the proposed method is discussed in this section. We used CNNs based method for feature extraction and classification. More details can be seen in the following paragraphs.

A. Architecture

The performance of the CNNs based model depends on several parameters. For example, the size of the kernels used, the convolutional layers numbers, and filters in every layer. Our proposed architecture is shown in Figure 2. In our model, we used two convolutional layers having filters specification as 24 (5×5) and 48 (5×5). For activation function, we used rectified linear unit (ReLu). After each convolutional layer, we embedded the pooling layer. For the pooling layer, we used Max-pooling.

A CNNs model has three main parts i.e., convolutional layers, pooling layers and fully connected layers. We represented the kernels as N×M×C. N and M are representing height a width of the filter and C channel. The pooling layers filters are represented by P×Q, where P and Q represent height and width of the filter, respectively. The fully connected layer is the final layer which performs the task of classification. Our complete framework is shown in Figure 2.

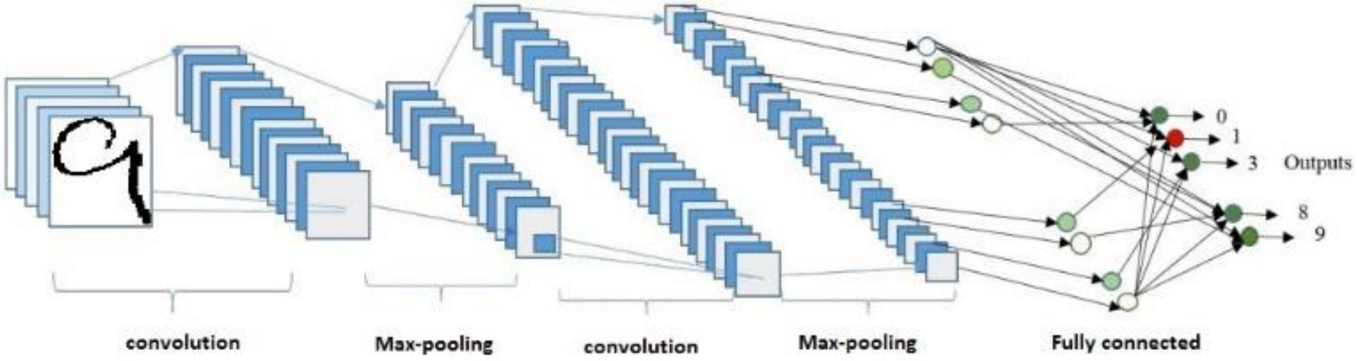

Figure 2. Proposed CNNs Architecture.

Layers

1) Convolutional Layers

Certain features from images are extracted through convolutional layers. These features include edges, corners, edge points, etc. We used a stride of 1 pixel for feature extraction. Input to the set of the convolutional layer can be computed as;

$$X_i^l = A_i^l + \sum_{c=1}^{M} W_i Y_{i+b,j+c}^{l-1}$$

where A_i^l represents the bias matrix and W_i represents the filter moving through the image. The activation function used is;

$$Z_i^l = \sigma X_i^l$$

The above Equation shows the applied activation function, which in our case, is ReLu.

$$\sigma X_i^l = \max(0, X_i^l)$$

The ReLu helps to increase the non-linear properties of the decision function and the overall network.

2) Pooling Layers

Small patches are taken from the output of convolutional layers, then down-sampled to produce a single output in the pooling layer. Different kinds of pooling layers are reported by the literature, in our work, we used maximum pooling. Maximum pooling takes the maximum value of the whole block. For pixel window, we fixed the size as 3 × 3.

3) Fully Connected Layers

The final content from the convolutional and pooling layer is given to the fully connected layer. First, the data is flattened and then given to the fully connected layer. A fully connected layer connects all neurons from the previous layer to its own neurons. The complete architecture with connectivity are shown in Figure 2.

TABLE I

Parameters setting for CNNs training.

Parameters	Values
Batch size	100
Epochs	20
Momentum	0.9
Base learning rate	10^{-4}

Similarly, information about each convolutional layer is in Table 2

TABLE II

Information about each CNNs layer.

Layer	Filter size (stride)	Output size
Input	-	30×30×1
Convolution-I	5×/1	26×26×24
Max-pooling-I	3×3/1	24×24×24
Convolution-II	5×5/1	20×20×48
Max-pooling-II	3×3/1	10×10×48
Fully connected layer (SoftMax)	-	Number of classes (10)

B. CNN Optimization

1) Learning Rate

For updating the weight of the network, we use the learning rate, which is α. α determines how the convergence of the network is done. If the value of α is slow, the convergence rate will be less, and if sufficiently large, divergence will occur. We selected the value of α with extreme care.

2) Activation Function

ReLu is commonly used as an activation function in deep learning models. We also used ReLu for activation. This function helps to increase the non-linear properties while taking a decision. The ReLu also helps in the generalization ability of the CNNs network and also reduces the computational cost of the model.

3) Stochastic Gradient Dsecent

We used Stochastic Gradient Descent (SGD) for weights and biases updating. A small step was taken by the SGD towards a negative gradient, which further minimize the error function.

$$P_{j+l} = P_j - \alpha \Delta E P_j$$

In the above equation, we represent the iteration number with j, learning rate with α, which must be > 0, vector parameter with P, and lastly, the loss function by EP_j. The whole dataset is used by the SGD once.

4) Mini-batch

The gradients of the proposed CNNs model is evaluated by SGD, which also updates all parameter through some part of training data. We called the subset of data as mini-batch. In the optimization process of the network, the whole database is divided into batches. For each batch, the gradient descent is calculated. After updating the network, the next batch is considered. In this way, the loss function is minimized with each iteration. Epoch is the full pass of the whole training data through small subsets, i.e., min-batch. We fixed the mini-batch as 100 and the Epochs 30 during our work.

5) Momentum

In some cases the descent algorithm oscillates to the steepest path when moving to the optimum value. We added momentum term for oscillation prevention. The SGD in such cases will be;

$$P_{j+1} = P_j - \alpha\nabla E(P_j) + \gamma(P_j - P_{j-1})$$

In this equation, the γ symbol decides how the gradient step used previously contributes to the current iteration. The data shuffles in all this process.

6) Regularization

During training process of the supervised learning, overfitting is a common problem. To the loss function we added a regularization factor for the weights. The loss function after regularization takes the form:

$$E_R P_j = EP_j + \lambda\Omega(\omega)$$

In this equation, the weight factor is represented by ω, the regularization co-efficient by λ, and the regularization function by $\Omega(\omega)$:

$$\Omega(\omega) = \frac{1}{2}\,\omega^t\omega$$

7) Softmax Classifier

In most of the CNNs model, Softmax classifier is frequently used for multi class classification tasks. For probabilistic cases, it is particularly helpful. We also applied Softmax for multi class classification in our work.

$$softmax(x_j)x_j = \frac{e_j^x}{\sum_{k=1}^{N} e_k^x}$$

In the above equation, the symbols x represents the input image to the network.

8) Network Parameters

TABLE III

Detailed results in the form of confusion matrix for ten classes.

True class \ Predicted class	0	1	2	3	4	5	6	7	8	9
0	95.0	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00
1	2.00	98.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	95.0	3.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	2.00	98.0	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	99.0	1.00	0.00	0.00	0.00	0.00
5	6.00	0.00	0.00	0.00	0.00	94.0	0.00	0.00	0.00	2.00
6	0.00	0.00	0.00	0.00	0.00	0.00	98.0	0.00	0.00	2.00
7	0.00	0.00	0.00	0.00	0.00	0.00	3.00	97.0	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.0	2.00
9	0.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	93.0

The network parameters play a vital rule in assessing the complexity of an architecture. These parameters make a clear comparison between various architectures. We computed the dimension of the feature map as;

$$Dimension_f = \frac{I - C}{S_{stride}} + 1$$

Where the symbol $Dimension_f$ represents the dimension of the used feature map for fully connected layer, I refers the feature map used as input, C is the filter which is convolved with I, and S_{stride} refers the stride used in the convolution process.

For each layer, we obtained the parameters through:

$$Q_j = (F \times F \times Fmap_{j-1}) \times Fmap_l$$

where Q_j represents the total parameters in the j_{th} layer, $Fmap_l$ is the output feature maps for j_{th} layer, and $Fmap_{j-1}$ are the total feature maps in the $(j-1)_{th}$ layer.

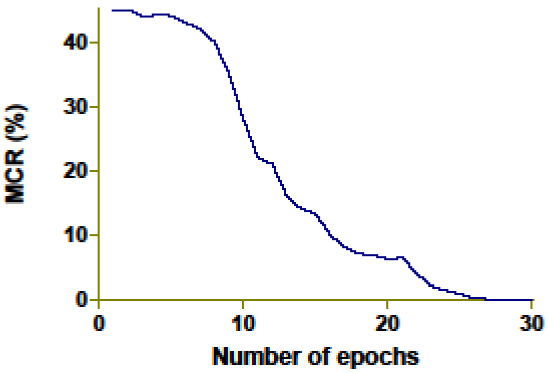

Figure 3. The mis-classification rate (%) for training data.

V. EXPERMENTS AND RESULTS

A. Experimental Setup

For experiments, we used Intel core i7 CPU. RAM of the system was 8GB, while GPU was NVIDIA 840M. We used TensorFlow and Keras for experiments. We trained the model for 20 Epochs, and the batch size was 100.

For Pashtu numerals recognition, the only dataset reported by the literature is PHND V-0. For the training stage we used 20,000 images, and the remaining 10,000 images for testing. We used subjects 1-1000 for training and 1001-1250 for testing.

B. Results Discussion

The performance of the proposed CNNs base model for numerals recognition is investigated and discussed in this section of the paper.

We cannot compare our results with any other Pashtu numerals database, as no database is still reported by the literature. We already presented details about the PHND in Section 3. The database consists of 30,000 images; we used 20,000 images for training and 10,000 for the testing phase.

From Figure 3 it is clear that the mis-classification rate reduces as the number of Epochs are increased. At 20 Epochs, the miss classification rate for training data almost reaches to 0.

For more details, we provided the confusion matrix for all ten classes in Table 3. From Table 3, it can be noted that the classification accuracy of some classes such as 0, 2, 3, and 5 is comparatively weak. The obvious reason for these poor results is the shape of these digits. For example, 0 is confused mostly with 5 and vice versa. These details can be studied in Table 3 for each class. As a whole, we obtained a classification accuracy of 91.64%.

In a nutshell, the results reported are encouraging and confirm the effectiveness of the newly proposed CNNs based model for the Pashtu numerals recognition.

VI. CONCLUSION AND FUTURE WORK

In the proposed work we introduce a new Pashtu numeral database named PHND V-0. The database consists of 30,000 images collected from three different universities in Pakistan. We make the database freely available for downloading and research purpose. We also introduce a deep learning-based recognition system for Pashtu digits. The deep learning model is based on concepts of convolutional neural networks. We use two layers of convolutional neural networks followed by a maximum pooling layer. The fully connected layer completes the classification task. For classification we used SoftMax.

Our current research work is part of our long term research strategy regarding cursive script languages. As future work, we are planning several directions. Firstly, we will build a complete Pashtu database for ligatures. In the next step, we will move towards complete OCR for the Pashtu language. We will explore the OCR system both for offline and online Pashtu text recognition. We also have planning towards translation of Pashtu language text to other languages. We also intend to apply our proposed deep learning-based methods to other cursive script languages passing through the same un-developed phase, such as Sindhi, Punjabi, etc.

VII. References

1. P. A. Stubberud, J. Kanai, and V. Kalluri, “Improving optical character recognition accuracy using adaptive image restoration,” Journal of Electronic Imaging vol. 5, no. 3, pp. 379–388, 1996.

2. Y. Du, C.-I. Chang, and P. D. Thouin, “Automated system for text detection in individual video images,” Journal of Electronic Imaging vol. 12, no. 3, pp. 410–423, 2003.

3. Q. Ye and D. Doermann, “Text detection and recognition in imagery: A survey,” IEEE transactions on pattern analysis and machine intelligence vol. 37, no. 7, pp. 1480–1500, 2015.

4. H. P. VC, “Method and means for recognizing complex patterns,”. US Patent 3,069,654, 1962 .

5. H. Penzl and I. Sloan, “A Grammar of Pashto: A Descriptive Study of the Dialect of Kandahar, Afghanistan”. Ishi Press, 2009.

6. Naz S, Hayat K, RazzakMI, AnwarMW,Madani SA, Khan SU. “The optical character recognition of Urdu-like cursive scripts. Pattern Recognition”. vol. 47, no. 3, pp. 1229–1248, 2014.

7. Durrani N, Hussain S. Urdu word segmentation. In: Human Language Technologies: “The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics”. Association for Computational Linguistics; pp. 528–536, 2010.

8. Lorigo LM, Govindaraju V. “Offline Arabic handwriting recognition: a survey”. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2006; vol. 28, no. 5, pp. 712–724, 2006

9. Gillies A, Erlandson E, Trenkle J, Schlosser S. “Arabic text recognition system”. In: Proceedings of the Symposium on Document Image Understanding Technology; 1999.

10. Margner V, Abed HE. “Arabic handwriting recognition competition”. In: Document Analysis and Recognition, 2005. ICDAR 2005. Eight International Conference on. vol. 2, pp. 70–74, 2005.

11. Margner V, Abed HE. “Arabic handwriting recognition competition”. In: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on. vol. 2, pp. 1274–1278, 2007.

12. Decerbo M, MacRostie E, Natarajan P. “The bbn byblos pashtu OCR system”. In: Proceedings of the 1st ACM workshop on Hardcopy document processing. ACM; 2004. pp. 29–32.Version November 28, 2019

13. Husain SA. A multi-tier holistic approach for Urdu Nastaliq recognition. In: Multi topic conference, 2002. Abstracts. INMIC 2002. International. IEEE; 2002. p. 84–84.

14. Mostefa, D., Choukri, K., Brunessaux, S. and Boudahmane, K., 2012. New language resources for the Pashto language. LREC 2012.

15. Ahmad, R., Amin, S.H. and Khan, M.A., 2010, October. Scale and rotation invariant recognition of cursive Pashto script using SIFT features. In 2010 6th International Conference on Emerging Technologies (ICET) (pp. 299-303). IEEE.

16. Wahab, M., Amin, H. and Ahmed, F., 2009, October. Shape analysis of pashto script and creation of image database for OCR. In 2009 International Conference on Emerging Technologies (pp. 287-290). IEEE.

17. Khan, Khalil, Rehan Ullah, Nasir Ahmad Khan, and Khwaja Naveed. "Urdu character recognition using principal component analysis." *International Journal of Computer Applications* 60, no. 11 (2012).

18. Khan, Khalil, Muhammad Siddique, Muhammad Aamir, and Rehanullah Khan. "An efficient method for Urdu language text search in image based Urdu text." *International Journal of Computer Science Issues (IJCSI)* 9, no. 2 (2012): 523.

19. Khan, K., R. Ullah Khan, Ali Alkhalifah, and N. Ahmad. "Urdu text classification using decision trees." In *2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET)*, pp. 1-4. IEEE, 2015.

20. Ahmad, R., Afzal, M.Z., Rashid, S.F., Liwicki, M., Breuel, T. and Dengel, A., 2016, October. Kpti: Katib’s Pashto text imagebase and deep learning benchmark. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 453-458). IEEE.

21. Alom, M.Z., Sidike, P., Taha, T.M. and Asari, V.K., 2017. Handwritten bangla digit recognition using deep learning. arXiv preprint arXiv:1705.02680.

22. El-Sawy, A., Loey, M. and Hazem, E.B., 2017. Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research, 5, pp.11-19.

23. Ram, S., Gupta, S. and Agarwal, B., 2018. Devanagri character recognition model using deep convolution neural network. Journal of Statistics and Management Systems, 21(4), pp.593-599.

24. Koyuncu, B. and Koyuncu, H., 2019. Handwritten Character Recognition by using Convolutional Deep Neural Network; Review. International Journal of Engineering Technologies, 5(1), pp.1-5.