Balochi Non Cursive Isolated Character Recognition using Deep Neural Network

Ghulam Jan Naseer[1], Abdul Basit[2], Imran Ali[3], Arif Iqbal[4]
Department of Computer Science and IT
University of Balochistan
Quetta, Pakistan.

***Abstract*—The text recognition research in artificial intelligence has enabled machines not only to recognize the human spoken languages but also to interpret them. Optical character recognition is a subarea of AI that converts scanned text images into an editable document. The researchers proposed various text recognition techniques to identify cursive and connected scripts written from left to right but their correct recognition is still a challenging problem for the visual methods. The Balochi language is one of them spoken by a significant part of the world population and no research conducted on the recognition this regional language of Pakistan. In this paper, we propose a convolutional neural network based model for Balochi script recognition for non-cursive characters. Our model optimized small VGGNet model and achieved exceptional precision and speed over the state of the art methods of machine learning. We experimented and compared the proposed method with the baseline LeNet model, the results showed the proposed method improved over the baseline method with a precision of 96%. We additionally collected and processed the Balochi characters dataset and made it public to carry further research in the future.**

***Keywords*—Convolutional neural network; data augmentation; character recognition; cursive character recognition; detection; text segmentation**

I. Introduction

Manipulation of the scanned document images remained a challenging task for the machines as the images are in pixel format known as raster graphics. The techniques of optical character recognition (OCR) transform printed and handwritten data into digital format so the machine can further control and process them.

Text recognition is more straightforward for non-cursive scripts such as the Latin script compared to cursive character recognition. The researchers have proposed approaches for the identification of both forms of script. Cursive and connected scripts recognition still needs a lot of attention, since the development of OCR for such scripts is still under research. Character recognition expands its umbrella to transform other spoken and written regional languages around the globe.

Balochi language is spoken in south-western Pakistan, especially in the province of Balochistan and Sindh by a large number of people. The language is also a great source of communication for the people who settled in the north-eastern regions of Khorasan and Sistan Balochistan which is the second largest province in Iran. It is also spoken by smaller communities settled in Afghanistan, Oman, United Arab Emirates, Turkmenistan, India, East Africa, and Bahrain [1].

The Balochi script is cursive and written from right to left. It inherits some of its characteristics from Arabic, Farsi, and Urdu scripts, additionally it has a larger number of characters such as differentiating characters, dot characters, range of location and dot orientations that have been minimized over time due to advancement in Balochi script. The Balochi Language is made up of 40 alphabets and 7 special characters, see Fig. 1.

Number of dots	=	Characters
With single dot	10	ض، ظ، غ، خ، ب، ج، ز، ذ، ف، ن
With two dots	02	ت، ق
With three dots	05	پ، ٹ، چ، ژ، ش
With small(ط)	04	ٹ، ڈ، ڑ، ط
Without dot	19	آ، ا، ح، د، ر، س، ع، ک، گ، ل، م، و، ه، ی، ے، ء، ص، ھ، ں
Total no of characters	40	

Fig. 1. List of the 40 non-cursive Balochi script alphabets.

The Balochi script also consists of seven special characters, these special characters contain dots or a diacritic above and below the characters. It makes the write-up and the pronunciation of the character different from the other characters, see Fig. 2.

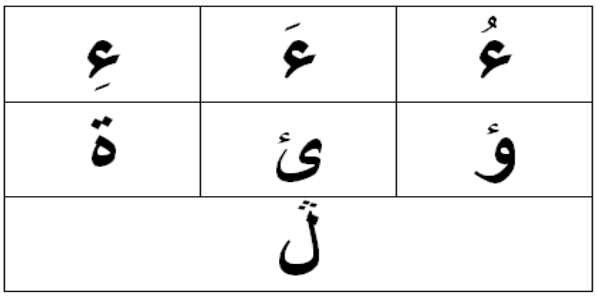

Fig. 2. Balochi script special characters with dots and other alphabets above and below. It is different in shape from other languages' script such as Urdu and Arabic.

Various position of the dots make the characters complex as a result segmentation and recognition also becomes difficult and challenging for the visual methods.

We propose a vision-based deep learning technique for the real-time classification and recognition of Balochi characters.

Vision-based methods extract meaningful features appeared on sophisticated printed and handwritten scanned images whereas, the machine learning helps in computational learning and pattern recognition tasks with justifiable accuracy.

II. Related Work

Researchers discussed various approaches and techniques in their research for the recognition of regional language characters and performed segmentation on these characters. In this section, we discuss different existing work carried on optical character recognition technique used for the right to left scripting languages especially focusing on Urdu, Farsi, and Arabic.

S.M Lodhi et al. [2] used the Fourier descriptor technique for the recognition of Urdu characters. The descriptor characterized the shape and features even in the presence of noise. The descriptor also performed the scaling and interpretation even if the shape and position of the characters are changed.

Lorigo and Govindaraju [3] used a combination of different methods based on artificial neural networks. They used Hidden Markov model and contour-based approach for the recognition of Arabic handwritten script. Arabic script is written from right to left and characters are joined in a machine-readable format. They also discussed the representation of Arabic letters, words, and analyzed handwritten methods. Mohammad et al. [4] also explored a segmentation and recognition technique for Arabic text and using a contour-based approach that found out edges and the region of interest for the sub-words and claimed improvements over the finding of the skeleton of the word.

Solimanpour et al. [5] explored the contour-based method for the recognition of Farsi character recognition. They prepared the Farsi language dataset comprised of characters, dates, and numerical strings. They also created the Farsi dataset for further research. Experiments were performed on the dataset and claimed better recognition results on Farsi characters and digits.

Shamsher et al. [6] explored the supervised learning to train a feed-forward neural network, later used the network for the identification of non-cursive characters. The proposed technique performed better during the training phase and claimed better recognition results.

Sattar et al. [7] used a Markov model (MM) for the recognition of the Urdu alphabets. They selected the full paragraph rather than isolated characters. The Markov model process a word as a chain of individual characters and considered sentence another chain of words. They extracted the features of each character and calculated the probability of recognizing each character.

Akhbari et al. [8] presented projection-based technique in which horizontal and vertical (x, y) histogram projection is applied to each line. The method detected the words and characters to divide the text lines. They proposed three steps to perform division such as segmentation of text lines from Arabic script images, divided the lines into words using blank spaces present between the words and finally used vertical projection technique for the segmentation of connected words.

Shaikh et al. [9] proposed a technique for the extraction of characters from the sub-words of cursive text. The algorithm used height profile vector (HPV) in which the difference between the first most pixel in each column of the sub-word and the baseline pixel for the segmentation of thinned stroke sub-words. The method helped them to find the location of the segmented points of the characters.

Alaei et al. [10] explored a method for Persian handwritten character recognition. The shapes are categorized into 8 different shapes out of a total of 32 Persian characters using a bitmap technique. In the bitmap technique, each character is recognized by a sliding window of size 7×7 to extract features. Finally, the support vector machine (SVM) algorithm is used for the classification of the text.

Taha et al. [11] proposed a method for the Arabic printed text character recognition. The proposed method consists of the following steps; image acquisition and preprocessing, segmentation of characters, feature extraction, and finally, character recognition.

D.N Hakro et al. [12] used optical character recognition technique to recognize Sindhi characters. The proposed method consists of basic image preprocessing steps to remove noise present in the target images and used a template matching technique for the character recognition.

Ahmed et al. [13] compared three different classifiers for the recognition of Urdu printed characters. The method consisted of a scale-invariant feature transform (SIFT), long short-term memory (LSTM) and Markov model. They analyzed that LSTM outperformed the other two baseline methods for character recognition.

A. Convolutional Neural Network

In this subsection, we discuss the existing techniques of character recognition based on the deep neural network.

Convolutional neural network [14] (CNN) is a profoundly deep learning algorithm used to identify image patterns. Neural network layers identify the corners and lines. Then, pass forward these extracted features to the neural net and begin to recognize more complex features. The algorithm performs the learning of extracted features and also the classification of target objects.

Al-Jawfi et al.[15] proposed a neural network for Arabic handwritten character recognition. The proposed method composed of two steps; feature extraction and character recognition.

Ahranjany et al.[16] used convolutional neural network with gradient descent algorithm for the recognition of handwritten Arabic and Farsi digits. The proposed method based on two steps; the first one is the extraction of input pattern features by using CNN and the second one is fusing the recognition results of the classifier to compensate the recognition errors.

Niu et al.[17] proposed a hybrid method support-vector machine and ConvNet for the recognition of the MNIST dataset. They used a neural network for the feature extraction and support vector machine for character recognition. Zamani et al.[18] used convolutional neural network and random forest (RF) algorithms for the recognition of Persian handwritten isolated characters on the Hoda dataset.

Elleuch et al. [19] claimed improvements in the hybrid model support vector machine and convolutional neural network by incorporating an additional dropout layer to get rid of over-fitting. They used the HACDB dataset for the training and validation.

Ashiquzzaman et al.[20] proposed an algorithm based on deep learning neural network with appropriate regularization layer and activation function for the recognition of handwritten Arabic numeral and claimed significant improvement over the other existing numeral recognition methods.

Elsawy et al.[21] proposed a convolutional neural network model for the recognition of Arabic characters. A comparison of neural network was made with deep learning techniques and analyzed that the proposed model outperformed in comparison of deep learning techniques.

Ali et al.[22] proposed a convolutional neural network approach for the detection and recognition of Urdu isolated characters in natural scenes that could not be handled by the traditional optical character recognition techniques. For this purpose, they presented a dataset of Urdu characters segmented from images of signboards, street scenes, shop scenes, and advertisement banners containing Urdu text.

The methods mentioned above proposed various solutions for the character recognition of cursive and right to left languages especially considering Arabic, Farsi, Urdu and other languages but we find no research work carried over the bigger spoken Balochi script. We generated the Balochi script dataset and tested with the existing baseline models such as Lenet, a convolutional neural network based model. We reached to a conclusion that the existing models are not suitable for the recognition of Balochi script characters and also generate poor results with the Blochi script dataset.

In this paper, we propose a customized ConvNet model and drawn better and improved results compared to the existing models. We focused on neural network based solutions to recognize Balochi script non-cursive isolated characters. Our proposed model performs better compared to the existing state of the art baseline machine learning approaches. Additionally, we prepared the Balochi language characters dataset to conduct our initial research and also provide a baseline ground for future research work.

We incorporated batch normalization and dropout layer between the two consecutive convolutional layers. The output consists of 47 classes of the Balochi script. In our research, we propose a small VGGNet [23] for the recognition of Balochi isolated characters. The model consists of the following layers: convolutional layers, rectified linear unit, batch-normalization, max polling layer, dropout layer, and a single fully-connected layer with a softmax classifier.

III. Dataset Preparation

We do not have any standard dataset for the Balochi language non-cursive or cursive characters to apply character recognition techniques. Therefore, we additionally prepared the Balochi characters dataset. The Balochi dialect comprised of 47 non-cursive characters. We prepared and created more than 70 samples of each character with various font styles [24].

We prepared 3290 total image samples in our customized dataset. The resolution of each sample image is 32×32 pixel in PNG format. We further applied image preprocessing operations to remove the noise and improve samples of the dataset.

IV. Methodology

In our proposed model, after preparing the dataset, we feed forward the input images having a resolution of 32×32 pixel to the convolutional neural network [14] for training the model to recognize the characters.

We first normalize the input layer so it results in improved accuracy of the model. Then, the pooling layer transforms the stack of filtered images into smaller patterns by using the max-pooling technique. Later, to avoid the chances of overfitting, we applied the dropout layer in our model. Finally, the fully connected layer connects each neuron of one layer to another preceding layer. In this layer, each neuron assigned a value that gets voted for the final character recognition. The softmax classifier depending upon the weights assigned to each neuron, takes the final classification decision, See Fig. 3. We discuss the details of each step in the next sections of the paper.

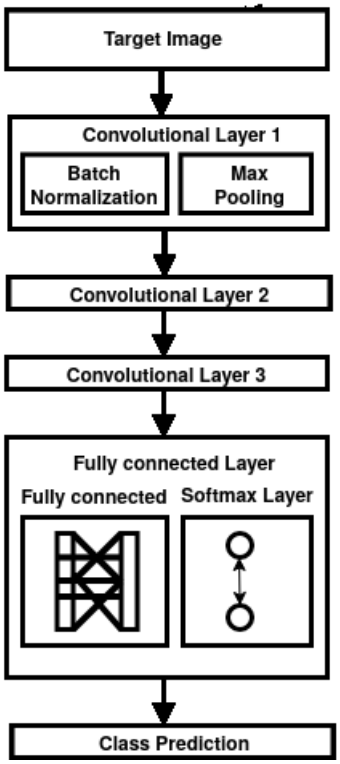

Fig. 3. Proposed method block diagram showing various stages of the algorithm. After input of the target image, series of convolutional layers extracts features that are feed-forward to batch-normalization and max-pooling layers. Then, the fully connected layer and softmax classifier perform classification.

A. Image Preprocessing and Formatting

In the first step, we crop and resize the character input images into $32 \times 32 \times 1$ having resolution $W \times H \times D$. Where W stands for the width of the image, H for the height and D shows the number of channels of the input image.

B. Proposed Network Architecture

After preprocessing and formatting the target image in the dataset, we feed the image to our proposed model. The first layer is the *convolutional layer* of the model. We feed the input image having a resolution of $32 \times 32 \times 1$ into this layer, and a kernel of $N \times N$ is applied to extract features.

In convolutional layer, the input image is divided into overlapping matrix with the help of various filters such as (5×5, 3×3 and 1×1) that produce feature maps. The layer extract features like corners and edges from the input target image.

Then, we used batch normalization to increase the training efficiency of our model. The technique of batch normalization standardizes the input image to a layer of mini-batch by changing negative values to zeros to apply the filters properly to extract deep features that accelerates the training process.

After batch normalization, we applied the max-pooling mechanism to reduce the dimensionality of the target image. It is a method of sub-sampling. The technique consists of several types, such as min, max, and average pooling.

In our proposed model, we prefer the max-pooling technique to extract maximum matrix values from extracted features. To prevent the overfitting problem, we added *dropout layer* in our model. The dropout layer processes the neurons randomly and ignore them during the training phase. It dropouts the ignored neurons to overcome overfitting issues in convolutional neural network based models.

The input target image having a resolution of $32 \times 32 \times 1$ is feed forward into a series of several convolutional layers based on different kernel sizes applied on the target image to find the location of features for extraction and to built feature maps. The final class prediction combined to the *fully connected layer*. The layer takes all the information of extracted features from the previous layers and feed-forward to the output layer. The fully connected layer with *softmax classifer* computes the maximum probability of each character and performs character classification, see Fig. 4.

C. Data Augmentation Technique

The generated image samples 3290 of Balochi Language characters are insufficient to train our proposed convolutional neural-based on small VGGnet model accurately on our custom-built characters dataset. To overcome the problem of over-fitting, we applied data augmentation technique during the training phase. The technique increased dataset size to 90000 by translation, scaling, rotation, weight and makes 1915 number of alphabet sample images of each character and we find the improved accuracy results in the character recognition.

D. Training Phase

Once the Balochi script character dataset is prepared, we can feed it to our customized model for training. We picked 3290 sample images of $32 \times 32 \times 1$ resolution and number of channels. We split the dataset into 80% that is 2632 sample images for training, and 20% of dataset 658 character images for validation. The training is performed for 32 epochs with 7000 steps per epoch. Once the training is completed, the proposed model generates a classifier for character recognition, see TABLE I.

TABLE I. TRAINING PARAMETERS FOR BALOCHI NON-CURSIVE ISOLATED CHARACTER RECOGNITION.

Training Samples	80%	2632
Testing Samples	20%	658
Total No of Samples	100%	3290
No of steps per epoch = 7000		

V. EXPERIMENTS AND RESULTS

After training the proposed model with our custom build Balochi script characters dataset, we set up a challenging experimental environment to test the robustness of the model. For testing, we collected a large amount of Balochi character images with varying fonts, styles and resolution.

We tested both speed and accuracy of the proposed small model and compared it with the other baseline ConvNet based models to make the fair and modest comparison and report improvements.

We built the two experimental setups to validate the proposed model. In the first experiment, we collected the images randomly online and created a small test dataset. We applied both the proposed and the baseline method over the collected images dataset and recorded the performance of every method. In the second experiment, we analyzed the training phase of the proposed and the baseline method to record the precision of the models.

A. Experiment I

In the first experiment, we built the 100 images dataset to test the proposed method and also the baseline methods. We compared the proposed model with the other baseline methods and recorded the model accuracy.

As already mentioned, we carried the training of the model on the Google GPU (Tesla K80) to save the time, but we conducted the tests both on CPU and Google GPU. For testing the model on the local CPU, we downloaded the Google GPU generated trained classifier with extension `.hdf` a binary file for the character recognition. We also tested the baseline LeNet model with the same collected test dataset to find the improvements of the proposed method and compare the performance, see Fig. 5.

The proposed method outperformed the baseline LeNet model in terms of accuracy, the proposed method present 96% whereas, the baseline LeNet performs with 86% accuracy.

B. Experiment II

We carried further tests in experiment II to ensure the robustness of the proposed method during the training phase.

We used a web-based precision evaluator platform called Tensorboard to test model accuracy. The board is used to visualize the performance of the methods by reading the TensorFlow event log files generated during the training phase. The training parameters were interconnected with the board,

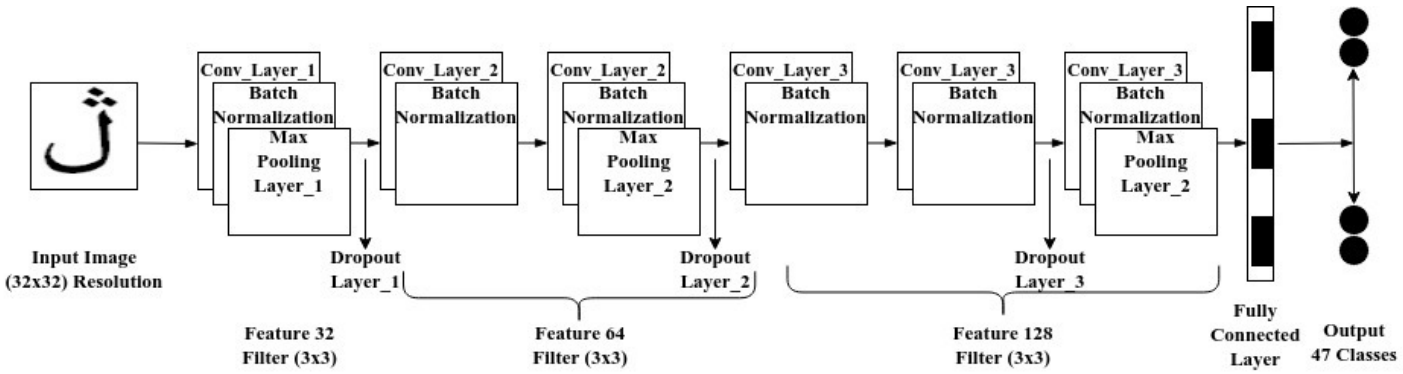

Fig. 4. The proposed model with various convolutional layers. An image is feed forward into the series of convolutional layers where kernel of varying sizes are applied to extract meaningful features. The batch normalization coverts the input image into a layer of mini-batch by converting negative values to zeros, this way it accelerates the overall training process, whereas, max-pooling mechanism extracts maximum matrix values from extracted features. The dropout layer uses the neurons randomly to avoid over-fitting. Finally, the fully connected layer and softmax classifier takes all the extracted features information from the previous layers and computes maximum probability of character for recognition.

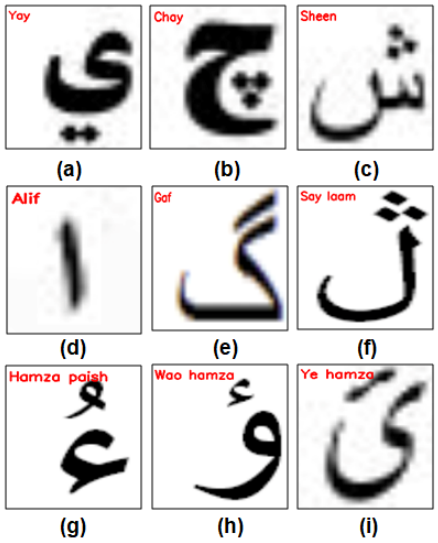

Fig. 5. The result of Balochi script non-cursive character recognition with the proposed model. We tested the method with randomly collected images and the model correctly recognized the characters with 96% accuracy.

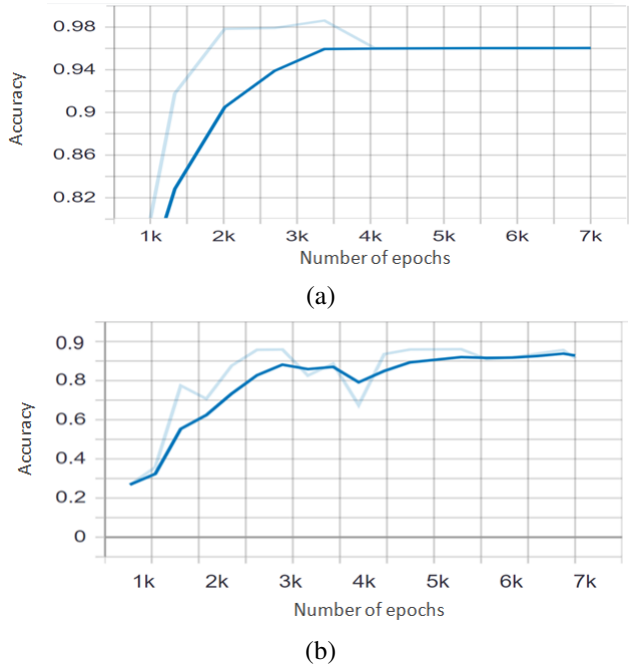

Fig. 6. Precision graph generated by the TensorBoard during the training phase. (a) The proposed method shows 96% accuracy. (b) The baseline LeNet model shows 86% accuracy.

which updates the precision and recall graphs during the training depending on the number of training epochs.

While discussing the speed and accuracy, the proposed method precision increases rapidly after 1200 iterations and quickly reaches to the accuracy of 96% after 4500 epochs and remains almost the same till 7000 epochs, on the other hand, the baseline LeNet method improves slowly and reaches to 86% accuracy for 7000 epochs. The proposed method for Balochi character recognition shows the improved result and outperformed the baseline method, see Fig. 6.

VI. Conclusion

In our research, we proposed a customized fast and accurate neural network based model for Balochi script non-cursive character recognition. We performed various experiments, and the results showed that the proposed method outperformed the baseline method both in accuracy and speed,

Additionally, we created the Balochi printed characters images dataset and made it available online [24] and also, applied data augmentation technique to get rid of overfitting. The proposed method trains rapidly compared to the baseline method and shows the precision of 96%.

In the future, we will extend our research approach to use vision-based methods to recognize the handwritten text and precisely segment the cursive detected characters.

Acknowledgments

We are thankful to the Higher Education Commission Pakistan and University of Balochistan for supporting Ghulam

Jan Naseer to carry his research.

REFERENCES

[1] C. Jahani, "Is there an" urban mind" in balochi literature?" 2010.

[2] S. Lodhi and M. Matin, "Urdu character recognition using fourier descriptors for optical networks," in *Photonic Devices and Algorithms for Computing VII*, vol. 5907. International Society for Optics and Photonics, 2005, p. 59070O.

[3] L. M. Lorigo and V. Govindaraju, "Offline arabic handwriting recognition: a survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 5, pp. 712–724, 2006.

[4] K. Mohammad, M. Ayyesh, A. Qaroush, and I. Tumar, "Printed arabic optical character segmentation," in *Image Processing: Algorithms and Systems XIII*, vol. 9399. International Society for Optics and Photonics, 2015, p. 939911.

[5] F. Solimanpour, J. Sadri, and C. Y. Suen, "Standard databases for recognition of handwritten digits, numerical strings, legal amounts, letters and dates in farsi language," 2006.

[6] I. Shamsher, Z. Ahmad, J. K. Orakzai, and A. Adnan, "Ocr for printed urdu script using feed forward neural network," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 23. Citeseer, 2007, pp. 172–175.

[7] S. A. Sattar, S. Haque, M. K. Pathan, and Q. Gee, "Implementation challenges for nastaliq character recognition," in *International Multi Topic Conference*. Springer, 2008, pp. 279–285.

[8] Z. Al Aghbari and S. Brook, "Hah manuscripts: A holistic paradigm for classifying and retrieving historical arabic handwritten documents," *Expert Systems with Applications*, vol. 36, no. 8, pp. 10 942–10 951, 2009.

[9] N. A. Shaikh, G. A. Mallah, and Z. A. Shaikh, "Character segmentation of sindhi, an arabic style scripting language, using height profile vector," *Australian Journal of Basic and Applied Sciences*, vol. 3, no. 4, pp. 4160–4169, 2009.

[10] A. Alaei, P. Nagabhushan, and U. Pal, "A new two-stage scheme for the recognition of persian handwritten characters," in *2010 12th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2010, pp. 130–135.

[11] S. Taha, Y. Babiker, and M. Abbas, "Optical character recognition of Arabic printed text," in *2012 IEEE Student Conference on Research and Development (SCOReD)*. IEEE, 2012, pp. 235–240.

[12] D. N. Hakro, I. A. Ismaili, A. Z. Talib, Z. Bhatti, and G. N. Mojai, "Issues and challenges in sindhi ocr," *Sindh University Research Journal (Science Series)*, vol. 46, no. 2, pp. 143–152, 2014.

[13] R. Ahmad, M. Z. Afzal, S. F. Rashid, M. Liwicki, and T. Breuel, "Scale and rotation invariant ocr for pashto cursive script using mdlstm network," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1101–1105.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] R. Al-Jawfi, "Handwriting arabic character recognition lenet using neural network." *Int. Arab J. Inf. Technol.*, vol. 6, no. 3, pp. 304–309, 2009.

[16] S. S. Ahranjany, F. Razzazi, and M. H. Ghassemian, "A very high accuracy handwritten character recognition system for farsi/arabic digits using convolutional neural networks," in *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*. IEEE, 2010, pp. 1585–1592.

[17] X.-X. Niu and C. Y. Suen, "A novel hybrid cnn–svm classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.

[18] Y. Zamani, Y. Souri, H. Rashidi, and S. Kasaei, "Persian handwritten digit recognition by random forest and convolutional neural networks," in *2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*. IEEE, 2015, pp. 37–40.

[19] M. Elleuch, R. Maalej, and M. Kherallah, "A new design based-svm of the cnn classifier architecture with dropout for offline arabic handwritten recognition," *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016.

[20] A. Ashiquzzaman and A. K. Tushar, "Handwritten arabic numeral recognition using deep learning neural networks," in *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. IEEE, 2017, pp. 1–4.

[21] A. El-Sawy, M. Loey, and E. Hazem, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS Transactions on Computer Research*, vol. 5, pp. 11–19, 2017.

[22] A. Ali, M. Pickering, and K. Shafi, "Urdu natural scene character recognition using convolutional neural networks," in *2018 IEEE 2nd international workshop on Arabic and derived script analysis and recognition (ASAR)*. IEEE, 2018, pp. 29–34.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] G. J. Naseer and A. Basit. (2020) Balochi characters dataset for machine learning. [Online]. Available: https://github.com/ghulamjannaseer/Balochi_characters