Pak. J. Engg. Appl. Sci. Vol. 25 July, 2019 (*p.* 36–42)

Multi-Font and Multi-Size Printed Sindhi Character Recognition Using Convolutional Neural Networks

Asghar Ali Chandio[1,2], Mehwish Leghari[2,3], Mehjabeen Leghari[3], Akhtar Hussain Jalbani[2]

1. *School of Engineering and Information Technology, University of New South Wales, Australia*
2. *Department of Information Technology Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah Pakistan*
3. *Department of Information Technology, University of Sindh, Pakistan*

* **Corresponding Author:** Email: a.chandio@student.adfa.edu.au

*

Abstract

In this paper, a problem of multi-font and multi-size offline printed character recognition of Sindhi language is addressed. Although previous studies for offline handwritten isolated Sindhi character recognition with unique font and size have achieved satisfactory results, the problem of multi-fonts and multi-size character recognition is still a major challenge. This is due to the various varieties in the shape, style, and layout of the character. A synthetic dataset with background color image consisting Sindhi characters with multi-fonts, multi-size, and multi-colors is created. Three types of experiments with Convolutional Neural Networks (CNN) are performed separately. The first CNN network uses max-pooling layer after every two convolutional layers, the second network applies max-pooling layer after the last convolutional layer and the third network is created without applying any max-pooling layer. The experimental results demonstrate that the max-pooling layers used after every two convolutional layers improve the performance significantly. The recognition results of 99.96%, 97.94%, and 98.72% are achieved with first, second and third networks respectively, which shows that CNN outperforms than the traditional machine learning algorithms.

Key Words: Multi-Font Sindhi Character Recognition, Multi-Size Sindhi Character Recognition, Printed Sindhi Character Recognition, OCR, CNN

1. Introduction

Optical Character Recognition (OCR) is one of the active areas of pattern recognition, and document analysis and recognition fields, which has been around the world for several decades. Several OCR systems for English, French, Chinese, Indic and Arabic scripts have been developed where high accuracies are reported. Many researchers are working to further improve the accuracy of Indic, Arabic, Persian and their derived scripts. However, the character recognition system for Sindhi script is still in its infancy stage. This is mainly due to the language complexities, variations in writing styles, similarity in the shape of two characters, variations in the number of dots or their positions over the shape and the joining of two or more characters. In cursive scripts, character recognition in unconstrained environment is even more complex and challenging task due to the accurate segmentation of characters from sub-words. Furthermore, it is more challenging task to segment the characters from cursive text when the text overlaps on either with the above line or the bottom line. In [1] a segmented technique is proposed for Sindhi sub-words where the images are first converted into binary form, and are thinned using a thinning algorithm [2]. The text lines are segmented by applying horizontal projection histogram and a connected component-based technique is used to extract the sub-words from the text lines, which are then segmented into isolated characters. OCR systems are used to recognize both the handwritten and printed letters and many systems such as Sindhi Dictionary [3], Sindhi script OCR [4-7], Unicode based word processor for Sindhi language [8] and a database for Sindh image text [9] have been developed, but to the best of authors knowledge, no work for multi-font and multi-size printed Sindhi character recognition has been reported yet. Multi-font and multi-size printed Sindhi character recognition is considered more challenging problem not only due to the larger set of Sindhi characters but also due to the large irregularities in the shape of the characters, styles and their layouts. Each character is printed in different style and has variations in the layout with different font type. Some of the multi-font, multi-size and multi-color Sindhi printed characters are shown in Fig 1. It can be observed that the characters in Fig 1 have variations in the stroke width, shape and styles.

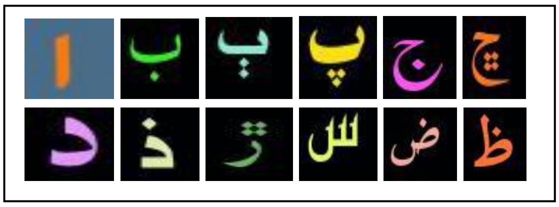

Fig. 1: Sample printed characters of Sindhi language with multi-fonts, multi-color and multi-size.

Sindhi is a type of cursive language written in Perso-Arabic script and has 52 letters. Most of the letters are taken from Arabic and Persian scripts and has 18 new letters. It is spoken by more than 40 million people [10].

Accurate recognition of multi-font and multi-size printed Sindhi characters will play important role to recognize text in documents, business cards, bank cheques, faxes as well as photographed scenes, etc. Furthermore, the fonts and colors of business cards, bank cheques and faxes may vary within a single word or text lines. This further underpins the importance of multi-font and multi-size Sindhi character recognition.

CNNs are widely used in many computer visions, image processing, pattern recognition and other visual applications [11-13]. Recently, CNNs have been applied for handwritten Chinese character recognition [14], printed Chinese character recognition [15], Arabic character recognition in natural scene images [16], Arabic handwritten character recognition [17], Urdu character recognition in natural scenes [18] and are combined with Recurrent Neural Network (RNN) for Urdu nastaliq recognition [19-20]. In this research work, a CNN is proposed where three separate CNN models are trained on multi-font and multi-size printed Sindhi character dataset. First CNN model uses a max-pooling layer after every two convolutional layers, the second CNN model applies max-pooling layer after the last convolutional layer and the third CNN model uses no any max-pooling layer. All three CNN models are evaluated on the test set separately and their results are compared.

The organization of the paper is as follows: Next section highlights the related work. The details of the proposed methodology and the implementation process are described in Section III. Section IV describes the experimental results on synthetic dataset and their discussions. Finally, Section V describes the concluding summary of the paper and possible future directions.

2. Related work

Printed text is not usually limited to a single font, but it can contain multiple fonts. Each font type has its own properties and the same character printed with multi-fonts has variations in the style. The OCRs proposed so far are efficient to recognize the characters with unique font type, but their accuracy decreases when applied for multi-font and multi-size character recognition. It is therefore, needed to develop an efficient OCR system which can recognize characters or text with multiple fonts and sizes. So far some research works are reported for handwritten Sindhi character recognition and several research studies have been performed to solve the problem of multi-font printed character recognition in many languages including English, Arabic, Kannada, Gurumukhi and Chinese scripts. In [4] a neural network based approach is proposed for handwritten Sindhi character recognition, where vertical projection is used to segment the characters from the lines and a zoning method is applied to extract the features. A multi-font system proposed in [15] uses multi-pooling convolutional neural network to recognize printed Chinese characters. A multi-pooling layer after the final convolutional layer in CNN is reported to be more efficient in spatial layout variations and those characters which have many distortions. A neural network-based method proposed in [21] applies similarity measure network which identifies the features of characters and the indicators associated with them. An accuracy of 98.56% is obtained on a database of multi-font printed English characters with 24 different fonts in 11 font sizes. In [22] the statistical features on the shape of isolated printed Arabic characters are calculated first. Both the K-Nearest Neighbor (KNN) and Random Forest Tree (RFT) classifiers are used to recognize the characters. The results that are reported show that the recognition accuracy of RFT is 11% more than KNN. In [23] a system for isolated printed Arabic character recognition is proposed using support vector machine where 99.08% of accuracy is achieved. In [24] a segmentation free approach for printed Urdu text recognition is proposed where statistical features are obtained, and Hidden Markov Model is employed for the classification. The proposed system is evaluated on Urdu Printed Text Images dataset and an accuracy of 92% is obtained. A neural network model proposed in [25] achieves 97% accuracy for machine printed Arabic character recognition.

3. Proposed Methodology

The baseline methodology of the multi-font and multi-size printed Sindhi character recognition is shown in Fig 2.

48x48 Printed Sindhi Character

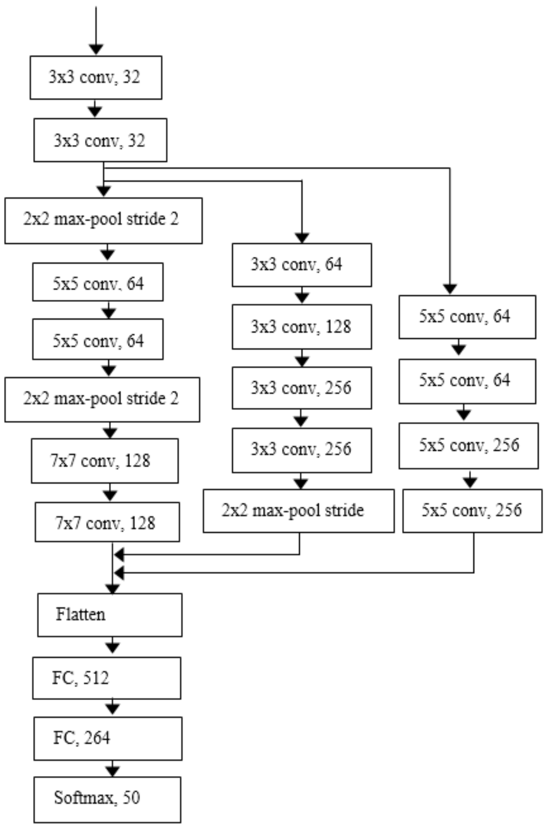

Fig. 2: Architecture depicting the proposed networks. Left: Network with max-pooling layers, Middle: Network with one max-pooling layer, Right: Network without max-pooling layers

For each experiment, same baseline architecture is used with same number of convolutional and fully connected layers. However, each architecture differs with the number of max pooling layers, dropout layers, the number of filters of convolutional layers and the kernel sizes. The max pooling layer down-samples the feature map, in order to reduce the number of coefficients of feature map for processing. The max pooling layer usually down-samples the feature map by a factor of 2. It first makes a dense map of the features and then takes the maximum activation value of the features over the smaller values. The network shown in Fig. 2 is similar to the architecture proposed by [15], however, the number of weighted layers are less than the layers in [15]. The proposed network consists of nine weighted layers, where initial six layers are the convolutional layers used to extract the features, the final three layers are fully connected dense layers including the last layer used as a classification layer. The width and height of the character images are different. However, the CNN takes a fixed size input image. Therefore, the width and height of the character images are resized to 48x48 pixels. The first convolutional layer convolves the 48x48x1 pixels image with 32 filters and a kernel size of 3x3. The second convolutional layer uses 32 filters with a kernel size of 3x3. Each convolutional layer is followed by an activation function with Rectified Linear Unit (ReLU). The ReLU has either the *zero* output if its input is negative or the *raw* output. That is, the output of the ReLU is same as the input value if its greater than zero as shown in Eq. (1).

$$f(x) = \max(x, 0) \tag{1}$$

The third and fourth convolutional layers apply 64 filters with a kernel size of 5x5 and the fifth and sixth convolutional layers apply 128 filters with a kernel size of 7x7. A dropout layer with a dropout ratio of 0.5 is used after the first fully connected layer. To reduce the parameters of the network, every two convolutional layers are followed by the max-pooling layer with a value of 2x2 and a stride of 2x2. To preserve the width and height of the image same after convolving, the padding with the "SAME" value is applied. This improves the classification performance by keeping the image border information. The two fully connected layers have 512 and 264 neurons and the last output layer has 50 neurons respectively.

The output layer uses a Softmax function for activation that gives a categorical probability equivalent output between 0 and 1, which is equal to 1 when the sum of total output is calculated. To measure the probability of error in classification tasks, a categorical cross entropy loss function is used during model compilation. The loss values are calculated as shown in Eq. (2).

$$l(y_k, y_j) = -\frac{1}{m} + \sum_{i=1}^{m}\sum_{j=1}^{n} y_j^i \log(y_k^i) + (1 - y_j^i)\log(1 - y_k^i) \tag{2}$$

Where, is the i^{th} training label for the output node j, y_j^i y_k^i is the prediction for the output node j, m is the number of batch or training samples, n is the probabilities for each of the output class, $1/m$ is mean of batch / training samples. To update the weights of convolutional layers and minimize the cost/loss function, a stochastic gradient descent optimizer is applied. This optimization function updates the weights of the convolutional layers for each sample or batch of the samples after cost function is evaluated as shown in Eq. (3).

$$W = W - \alpha \nabla L\left(W, b, x^i, y^i\right) \quad (3)$$

Where, W are the weights of the layers, α is the learning rate and ∇ is gradient value of the cost function $L(W, b)$ with respect to the update in the weights and bias values of the network, i is the i^{th} training sample. The model is trained using Stochastic Gradient Decent optimizer with a learning rate of 0.005, momentum of 0.9, decay of 0.0000625 and a batch size of 64. All the experiments are performed with Keras in Python and the training is performed on a CPU with 16 GB of RAM. The hardware and software facilities are provided by the UNSW, Canberra.

4. Experimental Results

A custom synthetic dataset for isolated Sindhi characters with multi-fonts, multi-colors and multi-sizes is created from forty different fonts of Sindhi and Urdu scripts. All the fonts are downloaded from the internet. The dataset is shuffled and divided into training set of 35000 samples and a test set of 15000 samples. The convolutional network gets a fixed size of input image, however, the width and height of the character images in proposed dataset are different. Therefore, all the character images are resized to a fixed size of 48x48 pixels before passing to the convolution network. The convolutional networks work more efficiently on smaller input values [20]. Therefore, all the pixel values of the training and test dataset are normalized from 0-255 to 0-1. In first experiment max pooling layers are used after every two convolutional layers and the recognition accuracy is 99.96%. In the second experiment, the max pooling is added only on the top of the last convolutional layer and the recognition accuracy obtained is 97.94%. In the last experiment, the max-pooling layer is not used, and the recognition accuracy achieved is 98.72%. Without max pooling layers, the total trainable parameters of the network were more than 70 million when a dropout of 0.3 was added after every two convolutional layers. These parameters are too large to train a small network and the result is overfitting of the model. When the max-pooling layer was added after every two convolutional layers, the total trainable parameters of the network were 10.09 million. A comparative analysis of recognition accuracy of the proposed model and other related works is shown in Table 1.

Table 1: Comparison between proposed model and others

Script	Feature Extraction	Classification Method	Recognition Accuracy %
Chinese [14]	CNN	Softmax	99.74
English [20]	Pixel by pixel	SOM NN	98.56
Arabic [21]	Statistical	KNN and RFT	98.25
Arabic [22]	Statistical	SVM	99.08
Urdu [23]	Statistical	HMM	92.00
Arabic [24]	Statistical	NN	97.00
Sindhi[4]	Zoning	NN	85%
Proposed-I	CNN	Softmax	99.96
Propose d-II	CNN	Softmax	97.94
Proposed-III	CNN	Softmax	98.72

All the three networks are trained with same learning rate of 0.005, momentum of 0.9, a decay of 0.0001 and a batch size of 64. Each network is trained with stochastic gradient descent optimizer and categorical cross-entropy as a loss function. The loss of training and test data with first network in Fig 2 is shown in Fig 4.

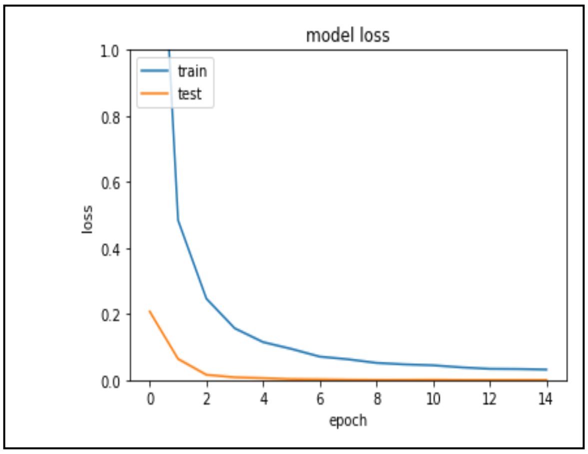

Fig. 3: Training and Test loss

The confusion matrix of the proposed network model (left in Fig 2) on test samples of the printed Sindhi character images with Softmax classifier is shown in Fig 4. It can be observed from Fig. 4 that there is only one pattern of diagonal line, which indicates that almost every character is correctly recognized by the classifier.

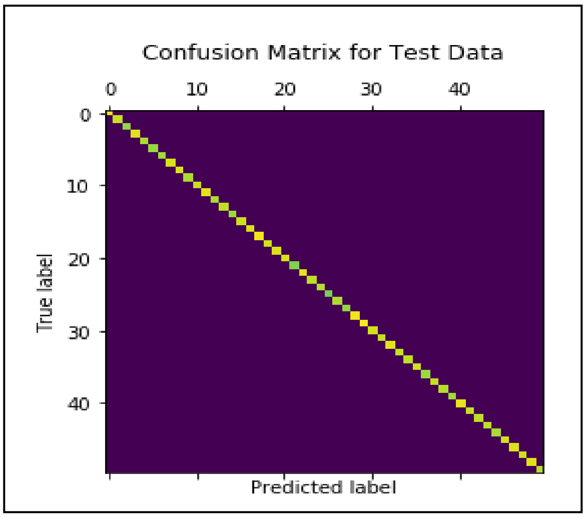

Fig. 4: Confusion matrix of printed Sindhi character images on test samples

There are very few Sindhi fonts available over the internet [26]. It is possible to create new fonts of Sindhi characters and generate a printed Sindhi character dataset with all the possible fonts to further test the proposed system. Some randomly selected results of the correctly recognized characters are shown in Fig 5.

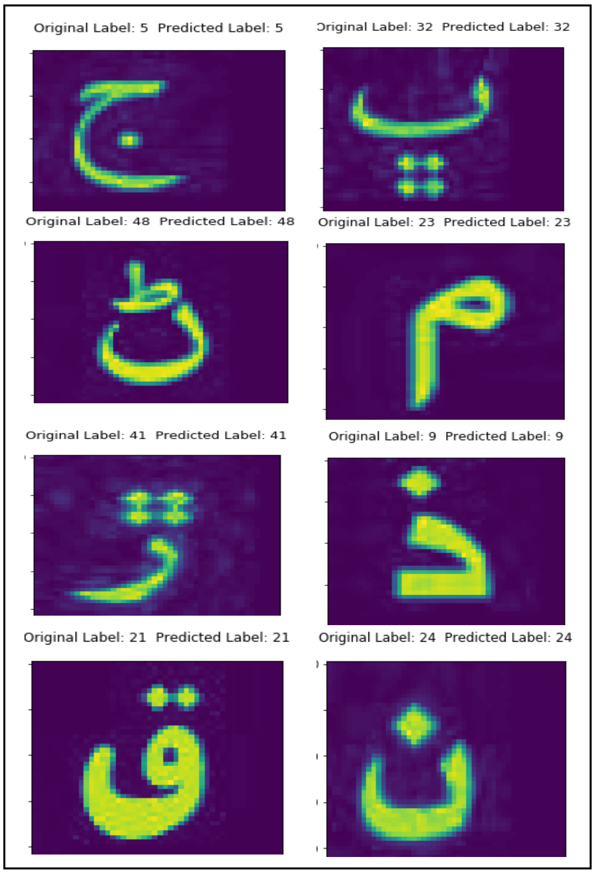

Fig. 5: Some randomly selected results of correctly recognized characters

5. Conclusions and Future Work

In this paper, a baseline research work for multi-size and multi-font printed Sindhi character recognition from synthetic images using deep convolutional neural network is proposed. All the character images are synthetically generated with different font sizes. The features are extracted using convolutional layers and the classification is performed by Softmax function. Three type of experiments are performed where in first experiment the max pooling layers are used after every two convolutional layers, in second experiment the max pooling layer is used on top of the last convolutional layer and in the third experiment, the max pooling layers are not used at all. The experimental results show that the network with max pooling layers after every two convolutional layers perform better than others. This work will further be extended to recognize whole words of printed Sindhi text. The recognition of Sindhi characters in real natural scene images is more complex and challenging task due to background complexity, variations in font size, styles, alignment and shape of the character. This research work will also be extended to recognize the Sindhi characters in real natural scene images and offline handwritten characters.

6. Acknowledgement

The authors are thankful to the University of New South Wales, at Canberra for providing computing facilities to perform experimental work.

7. References

[1] Shaikh NA, Mallah GA, Shaikh ZA. (2009). 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. 4160-9 4169.

[2] Shaikh ZA, Shaikh NA. A universal thinning algorithm for cursive and non-cursive character patterns (2006). MEHRAN UNIVERSITY RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY, vol. 25, no. 2, pp. 163.

[3] Bhatti, Z., Ismaili, I.A., Hakro, D.N., and Waqas, A.,(2014) “Unicode Based Bilingual Sindhi-English Pictorial Dictionary for Children”, American Journal of Software Engineering, Volume 2, No. 1, pp. 1-7.

[4] Awan SA, Abro ZH, Jalbani AH, Hameed M (2018) “Handwritten Sindhi Character Recognition Using Neural Networks”. Mehran University Research Journal of Engineering and Technology, vol. 1, no. 37, pp. 1-6.

[5] Solangi YA, Solangi ZA, Raza A, Shaikh NA, Mallah GA, Shah A (2018) “Offline-printed Sindhi Optical Text Recognition: Survey”. In 5th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), pp. 1-5.

[6] Hakro, D.N., Ismaili, I.A., Talib, A.Z. Bhatti, Z., and Mojai, G.N., (2014) “A Recognition”, Sindh University Research Journal (Science Series), Volume 46, No. 3, pp. 323-334, Jamshoro, Pakistan.

[7] Soomro WJ, Ismaili IA, Shoro GM (2018). Optical Character Recognition System for Sindhi Text: A Survey. University of Sindh Journal of Information and Communication Technology. Vol. 28, No. 2, pp. 1-7.

[8] Bhatti, Z., Ismaili, I.A., Soomro, W.J., and Hakro, D.N., (2014) “Word Segmentation Model for Sindhi Text”, American Journal of Computing Research Repository, Volume 2, No. 1, pp. 1-7.

[9] Hakro, D.N., Talib, Z., and Mojai, G.N., (2014) “Multilingual Text Image Database for OCR”, Sindh University Research Journal (Science Series), Volume 47, No. 1, pp. 181-186, Jamshoro, Pakistan.

[10] Mehwish Leghari & Mutee U Rahman (2015). Towards Transliteration between Sindhi Scripts Using Roman Script, Linguistics and Literature Review, vol. 1, no. 2, pp. 95- 104

[11] Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A survey of deep neural network architectures and their applications. Neurocomputing. Vol. 234, pp. 11-26.

[12] Schmidhuber J. Deep learning in neural networks (2015): An overview. Neural networks. Vol. 61, pp. 85-117.

[13] LeCun Y, Bengio Y, Hinton G. (2015) Deep learning. nature. Vol. 521, No, 7553, pp. 436.

[14] Xiao X, Jin L, Yang Y, Yang W, Sun J, Chang T (2017). Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition. Pattern Recognition, vol. 1, no. 72, pp. 72-81.

[15] Zhong Z, Jin L, Feng Z (2015), Multi-font printed Chinese character recognition using multi-pooling convolutional neural network. In 13th IEEE Internaltional Conference on Document Analysis and Recognition (ICDAR), pp. 96-100.

[16] Ahmed SB, Naz S, Razzak MI, Yousaf R (2017), "Deep learning based isolated Arabic scene character recognition,"In Proceedings of 1st IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR), pp. 46-51, Nancy.

[17] El-Sawy A, Loey M, Hazem EB (2017), Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research. Vol. 5, pp. 11-9.

[18] Chandio AA, Pickering M, Shafi K., (2018) “Urdu Natural Scene Character Recogntion using Convolutional Neural Networks”. In Proceedings of 2nd IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR), London.

[19] Naz S, Umar AI, Ahmad R, Ahmed SB, Shirazi SH, Siddiqi I, Razzak MI (2016), Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks. Neurocomputing. Vol. 177, pp. 228-41.

[20] Naz S, Umar AI, Ahmad R, Siddiqi I, Ahmed SB, Razzak MI, Shafait F (2017) Urdu nastaliq recognition using convolutional–recursive deep learning. Neurocomputing. Vol. 243, pp. 80-87.

[21] Samadiani N, Hassanpour H (2015), A neural network-based approach for recognizing multi-font printed English characters. Journal of Electrical Systems and Information Technology. Vol. 2, No. 2, pp. 207-218.

[22] Rashad M, Semary NA (2014), Isolated printed Arabic character recognition using KNN and random forest tree classifiers. InInternational Conference on Advanced Machine Learning Technologies and Applications, Springer, Cham, pp. 11-17.

[23] Yamina OJ, El Mamoun M, Kaddour S (2017) Printed Arabic optical character recognition using support vector machine. In IEEE International Conference on

Mathematics and Information Technology (ICMIT). pp. 134-140.

[24] Din IU, Siddiqi I, Khalid S, Azam T (2017), Segmentation-free optical character recognition for printed Urdu text. EURASIP Journal on Image and Video Processing, Vol. 1, pp. 62.

[25] Zheng L (2006), Machine printed arabic character recognition using s-gcm. In IEEE 18th International Conference on Pattern Recognition (ICPR), Vol. 2, pp. 893-896.

[26] http://www.bhurgri.com/bhurgri/amar/index.php/sindhi-computing/sindhi-font.