Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.036426
Article

Baseline Isolated Printed Text Image Database for Pashto Script Recognition

Arfa Siddiqu, Abdul Basit*, Waheed Noor, Muhammad Asfandyar Khan, M. Saeed H. Kakar and Azam Khan

Department of Computer Science & Information Technology, University of Balochistan, Quetta, 87300, Pakistan
*Corresponding Author: Abdul Basit. Email: drabasit@um.uob.edu.pk
Received: 29 September 2022; Accepted: 13 December 2022

Abstract: The optical character recognition for the right to left and cursive languages such as Arabic is challenging and received little attention from researchers in the past compared to the other Latin languages. Moreover, the absence of a standard publicly available dataset for several low-resource languages, including the Pashto language remained a hurdle in the advancement of language processing. Realizing that, a clean dataset is the fundamental and core requirement of character recognition, this research begins with dataset generation and aims at a system capable of complete language understanding. Keeping in view the complete and full autonomous recognition of the cursive Pashto script. The first achievement of this research is a clean and standard dataset for the isolated characters of the Pashto script. In this paper, a database of isolated Pashto characters for forty four alphabets using various font styles has been introduced. In order to overcome the font style shortage, the graphical software Inkscape has been used to generate sufficient image data samples for each character. The dataset has been pre-processed and reduced in dimensions to 32×32 pixels, and further converted into the binary format with a black background and white text so that it resembles the Modified National Institute of Standards and Technology (MNIST) database. The benchmark database is publicly available for further research on the standard GitHub and Kaggle database servers both in pixel and Comma Separated Values (CSV) formats.

Keywords: Text-image database; optical character recognition (OCR); pashto isolated characters; visual recognition; autonomous language understanding; deep learning; convolutional neural network (CNN)

1 Introduction

Various languages spoken around the world have no proper visual dataset available publicly. Whereas, the character recognition algorithms critically depend on the proper character dataset to train the machine learning algorithms and generate accurate results. In this perspective, researchers spent their valuable time and energy compiling various text-image databases and making them available for carrying out the research.

The scripts written from right to left are complex, connected, and less popular. They generally have no proper online database and thus, there is no rich scientific research found to detect and recognize the script both in the printed and handwritten formats.

The Pashto script is like other popular cursive scripts such as Arabic and Urdu written from right to left and spoken by a large group of people in Pakistan, Iran, and India. Additionally, it is also the official language of Afghanistan.

The main focus of this study is to build an isolated characters image database for the Pashto script to lay down a baseline for future research. Graphical software and various fonts were used to generate the database. Later, the database was processed and stored publicly to enable modern engines and tools to easily pull the data and process them. In future work, the study will be further extended to build a connected characters database which will help the visual algorithms to recognize words and sentences in the script.

In the remainder of the paper, the overview of the Pashto script and its alphabets are presented in Section 2. In Section 3, we analyze the existing literature on the cursive text-image database and its recognition. In Section 4, we discuss the contribution of our study and also summarize the complete flow of the dataset generation and preparation. Section 5 covers the detail of Pashto characters, generating the image database, preparing the database, and working with the online database. Finally, we conclude our work and future extension in Section 6.

2 Overview of Pashto Alphabets

Pashto script consists of 44 alphabets with 10 common numerals. The alphabets are case-sensitive, cursive, and also connected in nature. The script also contains dots both above and below the alphabet. This makes the language difficult for the visual algorithms to detect and recognize. Among those 44 alphabets, the script consists of 6 special characters that are not found in any other right-to = left cursive scripts. The characters not only vary in shape but also in pronunciation. This makes the script distinguished from other right-to-left scripts, see Fig. 1.

Figure 1: The Pashto script 44 alphabets with their pronunciation in English. The characters double-underlined are the special characters of the script not found in Arabic and Urdu, whereas the single-underlined text shows the characters varying in shape only

In addition to the six special characters, we have five other alphabets that vary in shape but have similar pronunciations to the characters present in other languages, such as the Urdu language. So, conclusively we have 11 characters that are varying in shape, among them six are special characters, see Fig. 2.

English	Urdu	Pashto
Te	ٹ	ټ
Dall	ڈ	ډ
Rhe	ڑ	ړ
Ga'f	گ	ګ
Yay	ے	ی

Figure 2: Pashto script alphabets differ in shape but have similar pronunciation to the Urdu script alphabets

3 Related Work

In this section, a detailed discussion of the existing work on cursive languages is presented. Beginning with a general text-image database available for cursive languages, their existing recognition techniques, the importance of annotating image databases, existing image databases for Pashto script, and concluding remarks.

3.1 Image Databases for Cursive Languages

Jaiem et al. [1] introduced an image database for Arabic printed text recognition. They used to test a system for recognizing Arabic printed texts with a large vocabulary. The database can also be utilized for word segmentation and font identification studies. 387 pages of Arabic printed documents were scanned in grayscale format at 300 dpi resolutions to create an Arabic text image database. There were 1,845 text blocks retrieved from these texts. For each texts-block, a ground truth file is also provided. The database additionally includes a 27,402 sample Arabic printed character image dataset.

Bouressace et al. [2] presented the Printed Arabic Text Database (PATD), a new comprehensive database that contains eight hundred and ten images scanned in grayscale format at various resolutions, resulting in two thousand and nine hundred and fifty-four images (smartphone-captured images) under various capture conditions (blurred, at different angles and in a different light). It's based on ten distinct newspapers with diverse structures, as well as an open-vocabulary, multi-font, multi-size, and multi-style text.

Al-Sheikh et al. [3] evaluated the Arabic Optical Character Recognition (OCR) dataset and discovered state-of-the-art methodologies in segmentation and recognition using Recurrent Neural Networks (RCNN) and the Connectionist Temporal Classification (CTC). This comprises a deep learning model as well as Gatted Recurrent Unit (GRU) implementation in the Arabic domain.

Chandio et al. [4] presented and analyzed a large dataset for Urdu text identification and recognition in natural scene photos. More than 2500 natural scene photos were acquired using a digital camera and a built-in mobile phone camera to create the dataset. We created three different datasets: isolated Urdu character photos, cropped word images, and end-to-end text spotting. Text detection

and recognition models were built using state-of-the-art machine learning and deep neural networks to evaluate the datasets and attain the best classification accuracies.

Hakro et al. [5] claimed that among languages that have adopted the Arabic script, Sindhi has the largest extension of the original Arabic alphabet, with 52 letters compared to 28 in the original Arabic alphabet, to enable additional sounds for the language. There are 24 distinct characters, some of which have four dots.

A database is required for Sindhi OCR research and development in order to train and test Sindhi text pictures. Authors created a massive database with over 4 billion words and 15 billion characters in 150 different fonts in four different weights and styles. Content for the database was gathered from a variety of sources, including websites, publications, and theses. The database considers words, characters, characters with spaces, and lines.

3.2 Image Databases with Character Recognition

Al-Muhtaseb et al. [6] explored a Hidden Markov Models (HMM) based technique for automatic recognition of off-line printed Arabic text. For each vertical sliding strip, multiple widths of overlapping and non-overlapping hierarchical windows were employed to generate 16 features.

For testing, eight different Arabic typefaces were employed (viz. Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic). Varying fonts have their maximum recognition rates for different numbers of states (5 or 7) and codebook sizes, according to experiments (128 or 256). Each form was assigned to a distinct class in this study, yielding a total of 126 classes (compared to 28 Arabic letters). The researchers also carried out research to eliminate the stop words [7,8] from the text.

Dahi et al. [9] tended to consider an automatic Optical Font Recognition (OFR) stage to improve the accuracy of Arabic character recognition before proceeding with the usual OCR phases. Scale Invariant Feature Transform (SIFT) descriptors were used to accomplish this. They further integrated statistical features and selected a Random Forest Tree (RFT) classifier for the classification stage. The classifiers are trained using a combination of features. As a result, each text font is assigned to a specific classifier tree. The proposed method was put to the test on a 30000 sample primitive Arabic characters noise-free dataset (PAC-NF).

Ul-Hasan et al. [10] explored using Recurrent Neural Networks (RNN) to print Urdu text in nastaleeq script. To recognize printed Urdu text, a bidirectional Long Short Term Memory (BLSTM) architecture with a Connectionist Temporal Classification (CTC) output layer was used.

Uddin et al. [11] extracted features and performed classification using convolutional neural networks (CNNs) trained on high-frequency ligature clusters. A query ligature is first separated into primary and secondary ligatures, which are identified separately and then combined in a post-processing step to recognize the entire ligature. Transfer learning on pre-trained networks is used in the experiments. The technique is evaluated on ligatures extracted from two standard databases of printed Urdu text, Urdu printed text image (UPTI) and Center of Language Engineering (CLE), as well as by combining the ligatures of the two datasets.

Naseer et al. [12] collected and analyzed the non-cursive Balochi characters dataset for the first time and made it publicly available. They also customized the convolutional neural network-based small VGGNet model for the recognition of the characters and achieved 96% precision over the baseline LiNet model.

Qaroush et al. [13] presented an omni font, open-vocabulary OCR for printed Arabic text based on segmentation. To deal with the problem of character overlapping, they introduced a baseline-dependent segmentation method that uses a hybrid, three-step character segmentation algorithm. Furthermore, it employs a collection of topological features that have been constructed and generalized to make the segmentation approach font agnostic. For feature extraction and recognition, the segmented characters are fed into a convolutional neural network. For testing and assessment, the Arabic Printed Text Image Database-Multi-Font (APTID-MF) data set was employed.

3.3 Handwritten Image Databases

After generating printed image databases, the researchers extended their work to generate the database for the handwritten text.

In the cursive and connected scripts, Arabic is the most famous and popular script. Researchers spent time generating datasets and explored various methods to recognize the text. Some of the handwritten image database research includes [14,15].

Ahmed et al. [16] worked on generating an Urdu text image database. Whereas, Zhang et al. [17] detailed how they generated a Chinese character database to carry the optical character recognition.

3.4 Pashto Image Databases Literature

As we already mentioned, the Pashto script possesses less scientific research and also lacks a proper text-image dataset to help the modern available techniques to recognize and translate the widely spoken language. Yet few researchers worked on the image dataset and on the recognition techniques of the text. We briefly discuss them in this section.

Ahmad et al. [18] scanned the Pashto books and extracted words to prepare the database. Later, they used the convolutional neural network-based algorithm to test the method. The author later presented optical character recognition technique [19] to recognize the cursive Pashto script.

Khan et al. [20] developed a database of moderate size, containing 4488 pictures derived from 102 differentiating samples for each of Pashto's 44 letters. Individual letters were classified using a zonal feature extractor, followed by K-Nearest Neighbour (KNN) and Neural Network (NN) classifiers. According to the results, KNN achieved an overall classification accuracy of around 70.05% percent, whereas NN achieved 72%. They further mentioned [21] accuracies of various methods that included support vector machine (SVM) at 56%, Artificial Neural Network (ANN) at 78%, and convolutional neural network (CNN) at 80.7%.

3.5 Annotated Image Databases

Data annotation or labeling is another step ahead after the data acquisition used by many popular machine learning techniques such as Single Shot multi-box Detector (SSD). The technique helps the learning algorithms to efficiently and quickly train the network with less training dataset.

Annotating or labeling the dataset is a technique to mark the target object in the image for training the dataset. Various applications are used to label the target in the database. The application generates an XML file containing information about the target object in the corresponding image. The network uses both images and XML files to perform training.

Guruprasad et al. [22] stated that annotated or semi-automated annotations take a significant amount of user intervention. These procedures take a long time and are prone to human mistakes. They highlighted the need for automatic data annotation techniques for offline handwritten text

recognition. In their work, they presented an end-to-end pipeline that uses deep learning and user interaction approaches to annotate offline handwritten manuscripts written in both print and cursive English.

In this section, the literature that was discussed only related to the text-image databases for the scripts written right to left and cursive in nature. The popular cursive and right-to-left scripts include Arabic, Urdu, and Farsi. This section has been ended by discussing available Pashto script databases and annotating such databases. Our focus is the available Pashto text-image databases to find a gap and add our contribution.

In the literature, there is less or no existing proper image database that leads to further language mining, translating, and understanding. The existing databases are neither properly managed nor meet the requirements of modern-day practices and the existing tools cannot use them easily. We are the first to initialize the language recognition by generating isolated printed characters image database for Pashto script just like other popular databases such as Modified National Institute of Standards and Technology (MNIST) [23] or CIFAR-10. So that, the database can easily be usable by any online tool such as Kaggle or Google co-lab.

4 Our Contribution

In this paper, a Pashto script was brought out from the literature to the proper optical recognition platform. A baseline isolated printed image database has been introduced for the Pashto script alphabets to enable modern tools to process them. Open-source graphical software Inkscape has been used for generating images using various available font styles. Standard procedures have been followed to pre-process the data and are made available online to facilitate future research.

The dataset was processed and converted to a format readily usable by the algorithms to extract useful information. The dimensions of sample images were reduced to 32×32 pixels, further, the images were converted to binary format with a standard black background and white text. The binary format occupies less space in the memory which helps in swiftly loading and processing datasets.

Further, the dataset was converted from pixel to the standard CSV format with a proper class label. The dataset in CSV format can be easily fetched and trained with any machine learning algorithm.

The dataset was stored in pixel format on GitHub [24] and in CSV format on Kaggle [25]. The dataset can be easily pulled from the servers and used in machine learning algorithms, see Fig. 3.

5 Methodology: Baseline Image Database Generation

This section details how the image database has been generated, the storage format, and how to fetch and access the online image database and use it for the research work.

5.1 Generating Image Database

Inkscape, an open-source tool has been used to generate the dataset. Additional fonts were installed to overcome the deficiency of the font styles in the application. Each Pashto character was typed and saved in a Portable Network Graphics (PNG) format, see Fig. 4.

Sixty sample images were generated for each alphabet with the various available font styles. It totals 2568 images in the image database.

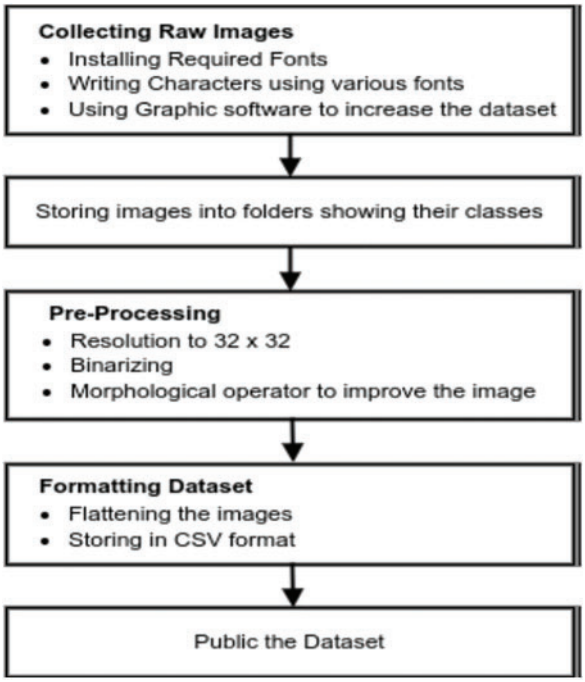

Figure 3: Steps involved in generating the Pashto text image database. Starting from collecting raw images using various font styles, and storing the generated images in their respective class folders followed by pre-processing. Finally, the dataset was formatted, converted to CSV format, and uploaded to the public server

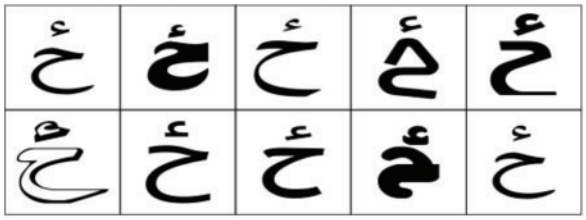

Figure 4: Pashto printed isolated alphabets. Pashto script characters were typed using various font styles and stored in raster graphics PNG format

5.2 Pre-Processing and Preparing Dataset

The generated dataset is in a raw format and unusable by algorithms straight away. Pre-processing becomes an important step for preparing the dataset for further processing and drawing useful information.

This section covers the details of the steps involved in preparing the dataset to store on the server correctly and easily accessible by the data science tools and applications.

5.2.1 Formatting Dataset

After storing the dataset in PNG format, the images might have been saved with an alpha channel that made the image background transparent. The alpha channel fails images to convert to binary format correctly.

An algorithm has been proposed which removed the alpha channel to get rid of the transparent background and converted the 4-channel image to a 3-channel image. The algorithm further converted each image sample to grayscale format reducing the 3-channel image to 1-channel. Further, each image sample was reduced to 32×32 dimensions and the image was smoothed with a Gaussian filter to suppress the noise.

5.2.2 Binarizing and Normalizing Dataset

In this phase, the images were converted to binary format using the Otsu algorithm [26]. The dataset in binary format occupies less space in the memory. This made the dataset easily accessible and trainable with limited computing resources such as Google co-ab which provides only 15GB of RAM. The sample images were inverted to a black background and white text to resemble the standard MNIST database, see Fig. 5.

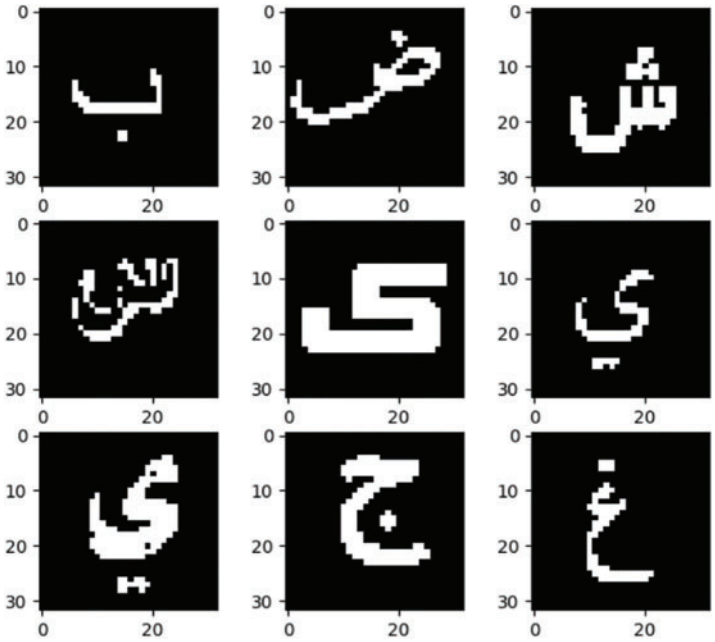

Figure 5: Pashto printed alphabets converted to binary format with 32×32 pixels dimensions. The images were stored with a standard black background and white text

Finally, the images were stored in their respective class folders, so the dataset can be easily accessible and later processed by the recognition algorithms in the future, see Fig. 6a.

5.2.3 Converting Dataset to CSV Format

Most of the popular and huge databases are stored in (Comma Separated Values) (CSV) format because it is easily readable by linear algebra programming scripts, faster, and smaller in size.

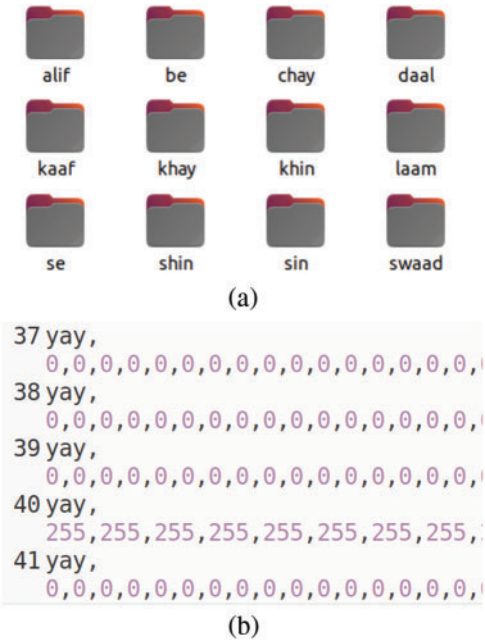

Figure 6: Processed dataset and image format. (a) Images in their respective class folders. (b) Image class (yay) in CSV format

In this phase, we flattened the binary image to transform the matrix data into a row vector. The same process was repeated for all sample images and stored each image as rows in the CSV file. Before saving data to a CSV file, we added a column representing the label or class of the sample image in text format such as yaya. The encoders can easily transform the text labels into integers later in the code. Thus, the first column entry in the CSV file represents the class labels for each data row, see Fig. 6b.

5.3 Working with Database

Image databases must have been stored and placed in a format so that every practitioner or researcher could easily pull and process them for various applications or research. After pre-processing and formatting the dataset, in this step, we uploaded the dataset to a public server and how to access it for further processing.

5.3.1 Setting Database to Public

We uploaded the file to a web-based service such as Kaggle. It is the data science and machine learning practitioner community that allows the users to find, store, and share the dataset. It also provides the tools to manage and process the online dataset. Both printed and handwritten datasets have been placed on the Kaggle platform for public use to encourage further research [25]. Additionally, the image database is made available on GitHub [24] in pixel format.

5.3.2 Using Database

The image database has been stored both on GitHub [24] in pixel format and on Kaggle [25] in CSV format. The reason to save CSV on Kaggle is that it provides the best APIs to make the data

easily accessible. One can directly download the dataset or use the APIs credentials to make the data access easy.

In order to use the public dataset from the GitHub server, one has to simply generate the access token and clone the dataset into Google co-lab and start using the image database for the research.

The database was made public on the Kaggle engine, one can use it in Google co-lab by generating the access token and downloading it in the JavaScript Object Notation (JSON) file format.

6 Conclusion

In this paper, a benchmark image database of Pashto printed characters has been introduced with the aim of initializing and enabling optical character recognition (OCR) for the Pashto script recognition. Various fonts and graphical software were used to produce sufficient image data points for the dataset creation. The dataset was processed using scripts and prepared both in pixel and CSV format to match the popular MNIST dataset. The database was stored on public servers GitHub and Kaggle for further research.

In future work, this study will be extended to build an image database not only for the isolated handwritten text but also for connected characters. A baseline platform will be established to apply scientific research to it and receive a complete autonomous understanding of the language.

Acknowledgement: This research and Arfa Siddiqu were supported by the Higher Education Commission Pakistan (HEC), and the University of Balochistan.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. K. Jaiem, S. Kanoun, M. Khemakhem, H. E. Abed and J. Kardoun, "Database for Arabic printed text recognition research," in *Int. Conf. on Image Analysis and Processing*, Springer, Naples, Italy, pp. 251–259, 2013.

[2] H. Bouressace and J. Csirik, "Printed Arabic text database for automatic recognition systems," in *Proc. of the 2019 5th Int. Conf. on Computer and Technology Applications*, Istanbul, Turkey, pp. 107–111, 2019.

[3] I. S. Al-Sheikh, M. Mohd, and L. Warlina, "A review of Arabic text recognition dataset," *Asia-Pacific J. Inf. Technol. Multimedia*, vol. 9, no. 1, pp. 69–81, 2020.

[4] A. A. Chandio, M. Asikuzzaman, M. Pickering, and M. Leghari, "Cursive-text: A comprehensive dataset for end-to-end Urdu text recognition in natural scene images," *Data in Brief*, vol. 31, pp. 105749, 2020.

[5] D. N. Hakro and A. Z. Talib, "Printed text image database for sindhi OCR," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 15, no. 4, pp. 1–18, 2016.

[6] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji, "Recognition of off-line printed Arabic text using hidden markov models," *Signal Processing*, vol. 88, no. 12, pp. 2902–2912, 2008.

[7] K. S. Dar, A. B. Shafat, and M. U. Hassan, "An efficient stop word elimination algorithm for Urdu language," in *2017 14th Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE, Phuket, Thailand, pp. 911–914, 2017.

[8] K. Shaukat, M. U. Hassan, N. Masood, and A. B. Shafat, "Stop words elimination in Urdu language using finite state automation," *International Journal of Asian Language Processing*, vol. 27, no. 1, pp. 21–32, 2017.

[9] M. Dahi, N. A. Semary, and M. M. Hadhoud, "Primitive printed Arabic optical character recognition using statistical features," in *2015 IEEE Seventh Int. Conf. on Intelligent Computing and Information Systems (ICICIS)*, IEEE, Cairo, Egypt, pp. 567–571, 2015.
[10] A. Ul-Hasan, S. B. Ahmed, F. Rashid, F. Shafait, and T. M. Breuel, "Offline printed Urdu nastaleeq script recognition with bidirectional LSTM networks," in *2013 12th Int. Conf. on Document Analysis and Recognition*, IEEE, Washington, DC, USA, pp. 1061–1065, 2013.
[11] I. Uddin, N. Javed, I. A. Siddiqi, S. Khalid, and K. Khurshid, "Recognition of printed Urdu ligatures using convolutional neural networks," *Journal of Electronic Imaging*, vol. 28, no. 3, pp. 033004, 2019.
[12] G. J. Naseer, A. Basit, I. Ali, and A. Iqbal, "Balochi non cursive isolated character recognition using deep neural network," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 717–722, 2020.
[13] A. Qaroush, A. Awad, M. Modallal and M. Ziq, "Segmentation-based, omnifont printed Arabic character recognition without font identification," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 3025–3039, 2022.
[14] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. T. Parvez *et al.,* "KHATT: An open Arabic offline handwritten text database," *Pattern Recognition*, vol. 47, no. 3, pp. 1096–1112, 2014.
[15] J. H. AlKhateeb, "A database for Arabic handwritten character recognition," *Procedia Computer Science*, vol. 65, pp. 556–561, 2015.
[16] S. B. Ahmed, S. Naz, S. Swati, I. Razzak, A. I. Umar *et al.,* "UCOM offline dataset-an Urdu handwritten dataset generation." *International Arab Journal of Information Technology (IAJIT)*, vol. 14, no. 2, pp. 239–245, 2017.
[17] H. Zhang, J. Guo, G. Chen, and C. Li, "HCL2000-A large-scale handwritten Chinese character database for handwritten character recognition," in *2009 10th Int. Conf. on Document Analysis and Recognition*, IEEE, Barcelona, Spain, pp. 286–290, 2009.
[18] R. Ahmad, M. Z. Afzal, S. F. Rashid, M. Liwicki, T. Breuel *et al.,* "Kpti: Katib's Pashto text imagebase and deep learning benchmark," in *2016 15th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR)*, IEEE. Shenzhen, China, pp. 453–458, 2016.
[19] R. Ahmad, S. Naz, M. Z. Afzal, S. H. Amin, and T. Breuel, "Robust optical recognition of cursive Pashto script using scale, rotation and location invariant approach," *PloS one*, vol. 10, no. 9, pp. e0133648, 2015.
[20] S. Khan, H. Ali, Z. Ullah, N. Minallah, S. Maqsood *et al.,* "Knn and ann-based recognition of handwritten pashto letters using zoning features," arXiv preprint arXiv:1904.03391, 2019.
[21] S. Khan, A. Hafeez, H. Ali, S. Nazir, and A. Hussain, "Pioneer dataset and recognition of handwritten pashto characters using convolution neural networks," *Measurement and Control*, vol. 53, no. 9–10, pp. 2041–2054, 2020.
[22] P. Guruprasad, S. Sujith Kumar, C. Vigneswaran, and V. S. Chakravarthy, "An end-to-end, interactive deep learning based annotation system for cursive and print English handwritten text," in *ICDSMLA 2020*, Springer, Pune, India, pp. 567–583, 2016.
[23] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[24] A. Siddique, A. Ali, and A. Basit, *Isolated Printed Text Image Database for Pashto Script*, 2022. [Online]. Available: https://github.com/abasitkhan/pashto_chrs_printed
[25] A. Basit, *Pashto Isolated Alphabets and Numerals*, 2022. [Online]. Available: https://www.kaggle.com/dsv/3357299
[26] X. Xu, S. Xu, L. Jin, and E. Song, "Characteristic analysis of Otsu threshold and its applications," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 956–961, 2011.