Machines and Algorithms
http://www.knovell.org/mna

Research Article

Advancing Handwritten Urdu Character Recognition: A Deep Learning Approach

Saima Sattar[1*], Fatima Yousaf[1] and Abbas Mubarak[3]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan
[2]Department of Computer Science, Institute of Southern Punjab, Multan, Pakistan
*Corresponding Author. Email: saimasattar02@gmail.com

Received: 21 April 2022; Revised: 09 June 2022; Accepted: 22 July 2022; Published: 17 August 2022
AID: 001-02-000007

Abstract: Optical Character Recognition (OCR) has emerged as a prominent field within Artificial Intelligence (AI) and is extensively researched in the domain of pattern recognition (PR). In recent times, OCR has garnered significant attention due to its pivotal role in facilitating the computer's ability to identify and interpret script present in images and documents. A wide variety of scripts, each requiring digitization and recognition, adds to the complexity of the OCR task. In this research, we present the development of a deep learning-based model specifically tailored for recognizing handwritten isolated Urdu characters. Our model employs Convolutional Neural Networks (CNNs) for efficient feature extraction from the input images. To evaluate the model's performance, we utilized the UHAT dataset, which consists of 28,328 training images and 4,880 testing images. The CNN model achieved an impressive recognition rate of 99.39% over 100 training epochs on the UHAT dataset. Furthermore, we curated a custom dataset, categorizing it into distinct training and testing subsets. The custom dataset encompasses 6,300 images, partitioned into an 80% training set and a 20% testing set. Our proposed model underwent training on 80% of the custom dataset and achieved a commendable recognition rate of 99.19% on the handwritten character images during testing. The results of this study demonstrate the effectiveness of our deep learning-based approach for Urdu character recognition, paving the way for enhanced OCR capabilities in handling diverse scripts and contributing to the advancement of pattern recognition technologies.

Keywords: Optical Character Recognition (OCR), Deep Learning, Handwritten Urdu Characters, Convolutional Neural Networks (CNNs), UHAT Dataset.

1 Introduction

Artificial Intelligence (AI) has witnessed remarkable growth in recent years, significantly impacting various aspects of our daily lives. One of the key components of AI, Machine Learning (ML), enables machines to learn from data and improve their decision-making capabilities over time. Pattern Recognition (PR) is a crucial field within ML, including the use of algorithms to find patterns and abnormalities in data. An important field in pattern recognition called optical character recognition (OCR) uses computer algorithms to extract data from digital photographs of printed or handwritten text.

OCR systems have been created for many other scripts, including Latin, English, and French, but it has been difficult to create accurate OCR programs for scripts like Arabic, Urdu, and Pashto because of their cursive character. For instance, the Urdu language, which has at least 38 alphabets, is a sophisticated fusion

of Persian, Arabic, and Turkish. The vast number of printed and handwritten documents in Urdu poses unique challenges for pattern recognition, especially in the case of handwritten content, where variations abound depending on individual writing styles.

The motivation behind this research stems from the need to address the existing challenges in recognizing handwritten Urdu script accurately. Despite significant progress in Urdu OCR, extracting relevant information from both printed and handwritten documents remains a challenge. Optical Character Recognition, which aims to convert handwritten or printed images into editable text using machine learning approaches, has seen considerable study, yet several difficulties persist in achieving high accuracy rates for Urdu OCR. Techniques have been employed for recognizing isolated characters or partial words, but attaining 100% accuracy remains an ongoing pursuit.

OCR systems can be divided into offline and online categories, as indicated in Figure 1, were offline OCR deals with scanned images of printed or handwritten documents and online OCR handles input directly from digital devices. OCR can also be separated into Handwritten and Printed OCR, with Handwritten OCR focusing on scanned documents and Printed OCR comprising notes, letters, and dictation. Notably, it is more challenging to identify handwritten materials due to the inherent variations in people's handwriting styles.

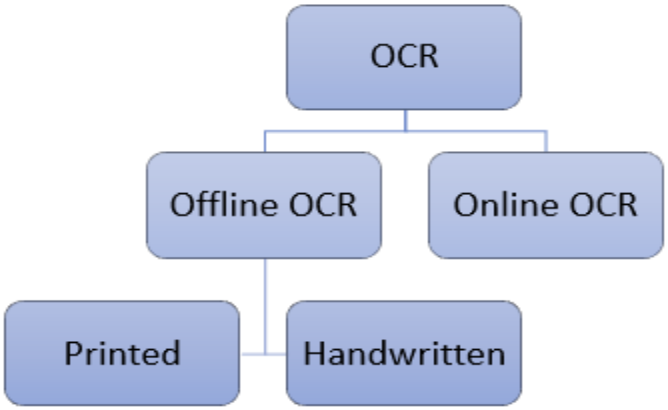

Figure 1: Classification of OCR

In light of the aforementioned challenges and the need for accurate Urdu OCR solutions, this research proposes the development of a comprehensive Urdu OCR system for recognizing handwritten isolated characters. The main objectives of this research are as follows:

- To create a powerful and thorough Urdu OCR for reading isolated handwritten characters.
- To develop a reliable method employing machine learning techniques for correctly identifying solitary Urdu characters.
- To assess how well the suggested machine learning technique performs using a dataset of handwritten isolated characters.

The subsequent sections of this research paper will delve into literature review, methodology, dataset, and experimental results, thereby shedding light on the proposed approach's efficacy in addressing the challenges of Urdu handwritten character recognition.

2 Literature Review

The historical evolution of the optical character recognition (OCR) system has been marked by a series of research initiatives aimed at overcoming challenges in the realm of pattern recognition. These endeavors have led to significant advancements in OCR technology, achieving nearly flawless accuracy rates—close to 100%—across various languages. Notably, languages using the Latin script have demonstrated

remarkable progress. Even languages with intricate cursive characteristics, like Arabic, have shown impressive recognition rates for both printed and handwritten text. Nevertheless, despite these remarkable achievements, a host of challenges remain to be addressed.

Despite the notable strides in OCR technology, certain cursive scripts such as Urdu, Pashto, and Persian have not received the same level of attention from researchers. This discussion delves into the intricate complexities and challenges that render the development of an Urdu OCR system a particularly demanding task. Additionally, this discourse offers a comprehensive exploration of the ongoing research efforts aimed at advancing Urdu OCR technology.

2.1 Introduction to the Urdu Script

Urdu, recognized as the national language of Pakistan, boasts a historical lineage interwoven with Persian, Turkish, and Arabic influences that date back to the Mughal Empire era. The etymology of "Urdu" can be traced to the Turkish term "Ordu," signifying 'army.' Comprising a total of 58 characters, the Urdu alphabet (as depicted in Figure. 2) encompasses 39 fundamental characters alongside 18 digraphs.

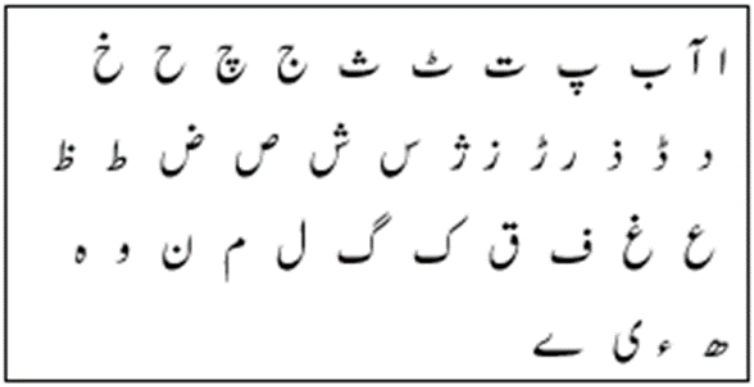

Figure 2: Set of Urdu Alphabets

The characters within the Urdu alphabet can be conveniently divided into two distinct groups: Joiners (as illustrated in Figure. 3) and non-joiners (depicted in Figure. 4). Non-joiner characters maintain their autonomy and do not establish connections with other characters, existing independently. Conversely, Joiner characters undergo transformations in their appearance contingent upon their placement within the context of writing. These transformations are observed when Joiner characters are situated at the commencement, middle, or culmination of a word. Non-joiner characters, in contrast, exhibit just two forms: "isolated" and "end."

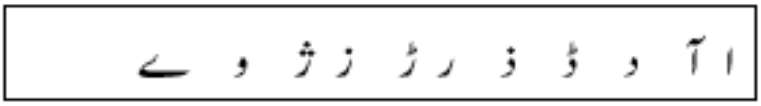

Figure 3: non-Joiners Characters

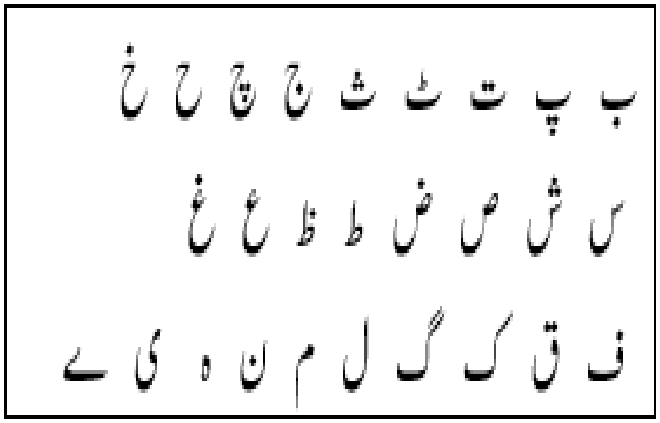

Figure 4: Joiners Characters

Furthermore, Urdu incorporates diacritics as a means to distinguish between characters. Diacritics encompass markings or accents that introduce alterations in the pronunciation or form of characters. The characters accompanied by diacritics within the Urdu language are depicted in Figure. 5 for reference.

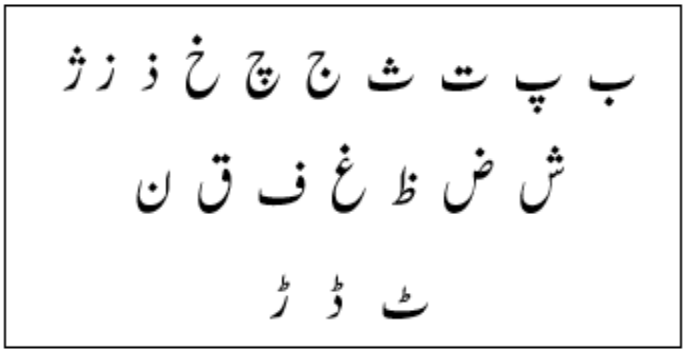

Figure 5: Diacritics Associated Characters

2.2 History of Urdu OCR

The journey of Urdu Optical Character Recognition (OCR) across history has been marked by significant advancements. In the realm of OCR systems tailored for cursive scripts like Urdu, two primary methodologies have emerged: Segmentation-based (Analytical) and Segmentation-free (Holistic) approaches. Different techniques prove effective in recognizing either isolated characters or ligatures (connected characters). The segmentation process for cursive scripts, such as Urdu, poses formidable challenges, leading to substantial research efforts and notable breakthroughs within the academic community [1]. An exploration of the literature is organized according to authors' segmentation techniques, which notably influence the efficacy of OCR systems designed for Urdu.

2.2.1 Ligature Segmentation Based OCRs

Using ligature-based segmentation, Syed Afaq Husain and Syed Hassan Amin [2] devised a multi-tier holistic technique to distinguish Urdu Nastalique. Incorporating variables including solidity, hole count, axis ratio, eccentricity, moments, normalized segment length, curvature, and bounding box width-to-height

ratio for recognition, they used connected component labeling for segmentation. To identify ligatures, their system made use of a feed-forward backpropagation neural network with 34 inputs, 65 hidden neurons, and 45 output neurons. Surprisingly, the system correctly identified all 200 samples in a collection of trained ligatures with 100% accuracy.

Addressing the intricacies of Nastalique script, including its cursive nature, diacritic placement, diagonality, and overlapping characteristics, Sobia Tariq Javed and Sarmad Hussain [3] employed horizontal and vertical pixel projections for line segmentation and the connected component method for sub-ligature segmentation. Their system achieved impressive results with 100% accuracy in line segmentation and 94% accuracy in ligature separation, successfully associating diacritics. Their dataset encompassed 3655 ligatures.

Sobia and colleagues [4] explored the distinctive attributes of Urdu characters, considering elements like context sensitivity, overlapping, diagonality, and shape variations in various ligatures. Given the complexities of segmentation, they opted for a segmentation-free approach. Their system incorporated global transformational features and a Hidden Markov Model as a recognizer, attaining a 92% accuracy using a dataset of 3655 ligatures.

Nazly Sabbour and Faisal Shafait [5] introduced Nabocr, an OCR designed to identify both Nastalique and Naskh scripts. They employed shape context as a feature descriptor and employed the K-Nearest Neighbor algorithm for ligature classification. Nabocr achieved error rates of 13.3% for ligature recognition and 11.2% for letter recognition. Notably, upon excluding foreign symbols, the error rates decreased to 9.1% for ligatures and 8.5% for letters Notably, the error rates fell to 9.1% for ligatures and 8.5% for letters when foreign symbols were removed.

Qurat and coauthors [6] proposed a comprehensive framework for Nastalique OCR, encompassing pre-processing, classification, and recognition stages. They developed two classifiers: Tesseract for ligature-level text recognition and classification, and a Segmentation-based method employing DCT coefficients and HMMs for classification. The system achieved accuracy rates of 90.10% for ligatures per page, 95.78% for the post-processing module, and 86.15% for the end-to-end system.

Based on a ligature segmentation method, Tofik and colleagues [7] proposed an OCR for printed Nastalique script. SIFT and SURF were utilized to compute descriptors for ligature extraction and classification. With a dataset comprising approximately 23,204 ligatures, the system achieved a commendable 95% accuracy.

Addressing font-size independence for Nastalique-based OCR, Qurat ul Ain Akram and Sarmad Hussain [8] devised a system accommodating font sizes ranging from 14 to 28. Tesseract was employed for main body recognition, while the C4.5 algorithm facilitated the division of main bodies into subsets for enhanced accuracy. Their system achieved main body recognition accuracy ranging from 95.13% to 97.20% across various font sizes.

Zaheer Ahmed and coauthors [9] introduced a ligature analysis-based Urdu OCR framework, incorporating a hybrid technique involving statistical correlation and pattern matching for ligature segmentation and classification. They compiled their own Urdu Nastalique ligature dataset, consisting of 3500 ligatures across 36 font sizes, yielding accuracy rates between 80.6% and 97.4% for isolated characters, 2-character ligatures, and 3-character ligatures.

Ibrar Ahmad and associates [10] underscored the significance of segmentation in recognition accuracy and emphasized the efficiency of line and ligature-based Urdu OCR compared to character-based methodologies. Their system used connected component approach for ligature segmentation, together with their patented curved line split algorithm and conventional horizontal projection. The system's exceptional accuracy rates for line segmentation and ligature segmentation were 99.17% and 99.80%, respectively.

Israr Uddin and colleagues [11] first proposed a strategy for Urdu Nastalique based on statistical characteristics and HMM classification for segmentation-free OCR. When HMMs were trained for ligature classification using Hu's moment, Zernike, and two-dimensional FFT energy features, the accuracy rates

for primary ligatures, secondary ligatures, and associated primary and secondary ligatures were 95.24%, 93.30%, and 92.26%, respectively.

Khawaja Ubaid Ur Rehman and Yaser Daanial Khan [12] developed an Urdu Nastalique ligature recognition OCR to address scale and rotation invariance. The system obtained accuracy of 96.474% on independent dataset testing and 96.922% on 5-fold cross-validation by using scale and position invariant moment approaches for feature extraction and a cascade forward-backpropagation neural network for classification.

Saud and team [13] presented a method for line segmentation in handwritten and printed Urdu text. By utilizing modified header and baseline detection, skew detection, and profile projection methods, the system achieved high accuracy ranging from 98.1% to 99.42% across different datasets.

2.2.2 Character Segmentation Based OCRs

The "Recognition of Printed Urdu Script," which was introduced by U. Pal and Anirban Sarkar [14], emphasized the difficulties brought on by the similarity of Urdu letter shapes. The system used the Hough transform for skew detection and correction, projection profiles for line segmentation, linked component labeling for character segmentation, and vertical projection profiles for skew detection and correction. A tree-based methodology was used for classification, relying on tropological variables, contour features, and water reservoirs. Surprisingly, the algorithm used a dataset of 3050 characters to achieve a character-level accuracy of almost 97.8%.

Faiza and coauthors [15] presented "Conversion of Urdu Nastalique to Roman Urdu" employing a character-based segmentation method. The approach encompassed image acquisition and binarization, followed by Binarization and Target Image Detection for pre-processing. Segmentation relied on template matching, while conversion was achieved through pattern matching. This approach demonstrated high effectiveness, boasting 99.8% accuracy in segmentation and 98% accuracy in conversion.

Based on moment invariant features, Imran Khan and colleagues [16] proposed "Recognition of Offline Handwritten Isolated Urdu Characters." The script was categorized into primary and secondary components, with invariant moments employed for segregating secondary components from primary ones. For classification, SVM was utilized, integrating 28 moments invariant featuresfor primary components and an additional 21 features for secondary components. The system showcased an average accuracy of 93.59% employing a dataset containing 36,800 handwritten characters.

Addressing the intricate nature of Nastalique writing style involving cursive attributes, compactness, overlap, and diagonality, Sarmad Hussain and associates [17] adopted a character-based approach for text segmentation into lines. A modified approach was employed for ligature segmentation, leveraging traversal along the thinned contour of the main body. Filtered DCT components from traversal were fed to HMM for recognition. The system achieved accuracy of 87.44% on diverse book images and an impressive 97.11% on 5249 main body classes.

A method for segmenting Nastalique-style Urdu letters using horizontal and vertical projection profile techniques was introduced by Aejaz Farooq Ganai and Faisal Rasheed Lone [18]. Vertical projection was used for ligature segmentation, while horizontal projection was used for line segmentation. Later ligature thinning and character segmentation using the chain code technique were carried out. The system's accuracy rates were 65.2% overall, 64.3% for visible data, and 70% for unseen data.

In a hybrid approach, Aejaz Farooq Ganai and Ajay Koul [19] combined horizontal and vertical projection profile methods for Nastalique script segmentation. The algorithm dynamically selected either vertical or horizontal segmentation based on peak values. Diacritic separation from ligatures was facilitated through connected component and horizontal projection profile methods. Hidden Markov Models were deployed as recognizers for diacritics. While the segmentation accuracy stood at 91.3%, recognition accuracy reached 78%, attributed to issues in diacritic association.

A modified Feed Forward Neural Network was developed by Sobia Habib and colleagues [20] for segmenting images with damaged Urdu and Devanagari script. The approach entailed thinning operations via two-pass algorithms prior to segmentation using the FFNN. The authors highlighted the superior performance of their approach in terms of accuracy and mean absolute error compared to Fuzzy C-means, k-means, and Threshold methods, although detailed training data specifics were not provided.

2.2.3 Implicit Segmentation Based OCRs:

Exploring the potential of Recurrent Neural Networks (RNNs) for the Nastalique script, Adnan and colleagues [21] drew inspiration from RNNs' success in recognizing English and Arabic scripts. Utilizing Connectionist Temporal Classification (CTC), they managed the requirement for pre-segmented data. With the evaluation of ligature shape variations, the error rate reached 13.574%, reducing to 5.15% when shape variations were excluded. However, due to an imbalanced dataset concerning ligatures with and without shape variations, the authors recommended further assessment using a more equitable dataset distribution.

Saad and coauthors [22] introduced the UNHD dataset, a compilation of handwritten samples for the Urdu Nastalique script. This large dataset, which included 10,000 lines, 3,12,000 words, and 1,87,200 characters from 500 authors, showed variations in slant and the presence or lack of baselines. The error rate for character recognition with BLSTM varied from 6.04% to 7.93%.

On the UPTI dataset, Saeeda Naz and associates [23] utilized the MDLSTM with CTC layer for training and testing. They used the sliding windows method to extract 15 various statistical or geometric properties, and they experimented to find the best feature set. This innovative endeavor using manual features with an RNN variation established a standard for prospective investigations, even if the greatest accuracy reached was 94.97%.

Saeeda Naz and coauthors later developed an implicit segmentation-based method for the Urdu Nastalique script [24]. Employing a sliding overlapped windows approach, they extracted 12 distinct features from lines of text, aggregating them into a unified feature vector. Character recognition was carried out using MDLSTM with CTC, culminating in an accuracy of approximately 96.40%. The unique nature of their methodology allowed comparison with only two other studies, where their approach secured the highest accuracy.

Addressing the efficiency of zoning features in Urdu script recognition, Saeeda Naz and associates [25] integrated zoning features with a 2D-LSTM network as a learning classifier, leveraging the UPTI dataset. Partitioned into training, validation, and testing sets, the approach achieved peak accuracy of 93.38% with a maximum zoning feature size of 9x9.

In a subsequent exploration [26], Saeeda Naz and colleagues acknowledged the triumph of convolutional networks across classification tasks. Merging CNN with MDLSTM for character recognition, they employed CNN to extract foundational features from the MNIST dataset. These features were then convolved with Urdu text images to extract informative features, subsequently classified by MDLSTM. The hybrid method attained an impressive 98.12% accuracy rate on the UPTI dataset. The authors suggested that the success of their hybrid approach could potentially be extended to other writing styles by utilizing datasets in the relevant language, as opposed to MNIST.

2.2.4 Miscellaneous Approaches:

By using a supervised learning strategy, Inam and colleagues [26] trained a feed-forward neural network (MLP) to recognize specific Urdu letters. They used a three-layer neural network with 150 neurons in the input layer, 250 in the hidden layer, and 16 in the output layer. They impressively achieved a 98.3% accuracy percentage for the Urdu alphabets using the Ariel typeface. Although they claimed that font size and style were invariant, they did not disclose any specific results.

In order to recognize printed Urdu language, Sobia Tariq Javed and Sarmad Hussain [27] strategically segmented ligatures at branching locations. Their approach involved the extraction of main bodies (ligatures devoid of diacritics) and subsequent skeletonization through the Jang Chin algorithm. Recognition relied

on HMM. On printed Nastalique text featuring a font size of 36 and a resolution of 150 dpi, they achieved an accuracy rate of 92.73%. However, they acknowledged that variations in shapes due to noise and inherent ligature segment similarity could potentially influence accuracy.

Qurat and colleagues [28] adapted the open-source OCR engine Tesseract for the Nastalique script. Their approach entailed separate training and recognition of main bodies and diacritics. By selectively deactivating specific functionalities within Tesseract, they managed to enhance accuracy. Their efforts notably yielded considerable accuracy improvements for text sizes 14 and 16.

Israr Uddin and coauthors [29] highlighted the advantages of holistic techniques over analytical methods, basing their approach on ligatures as fundamental units for both feature extraction and classification. They harnessed Discrete Wavelet Transform coefficients of ligatures as features for HMM training, culminating in an average accuracy of 89%.

Nizwa Javed and associates [30] explored the efficacy of Convolutional Neural Networks (CNNs) in ligature classification. Their efforts resulted in the creation of a dataset comprising 55,000 Urdu ligatures spanning 552 classes. Leveraging the CNN architecture for ligature classification, they achieved a commendable recognition rate of 95.3%.

Asma Naseer and Kashif Zafar [31] utilized thickness graphs to extract meta-features from ligatures. One CNN and one LSTM network model was trained using initial images, and the other was learned using thickness graphs. Their results showed that the usage of thickness graphs led to an average accuracy rate of 98.08%, outperforming the usage of raw images.

Muhammad Jawad Rafeeq et al. [32] created an augmented dataset of ligatures and used clustering to increase learning speed. They employed a deep neural network that outperformed a simpler neural network, achieving an accuracy rate of 95.02% with dropout regularization.

Syed Yasser Arafat and Muhammad Javed Iqbal [33] proposed a holistic segmentation-free approach for ligature categorization. They used a double-layer BLSTM with CNN and VGG16 features and achieved an accuracy of 80.46% for predicting partial sequences of characters in ligatures.

Mohammed Aarif K.O and Sivakumar Poruran [34] fine-tuned AlexNet and GoogleNet for handwritten Urdu character recognition. They achieved promising results, with AlexNet outperforming GoogleNet.

Atique Ur Rehman and Sibt ul Hussain [35] generated a synthetic dataset for Urdu ligatures to overcome the time-consuming and error-prone process of manually creating a large dataset. Their accuracy evaluations were 90% on 400 classes, 89% on 2000 classes, and 85% on the entire dataset using a CNN built on ResNet-18.

It's worth noting that while some of the mentioned research studies have achieved high accuracy rates on their specific tasks, the development of a comprehensive end-to-end Urdu OCR system involves addressing numerous challenges, such as segmentation, noise handling, and script variations, which can significantly impact the overall accuracy and performance of the system.

2.2.5 Discussions

The above-mentioned research papers present a diverse range of approaches and methodologies for Urdu Optical Character Recognition (OCR). Several studies have explored the effectiveness of different techniques, including traditional methods and deep learning-based approaches, to tackle the challenges associated with recognizing Urdu script, known for its cursive nature, ligatures, and complex shapes.

Some researchers have focused on character-based segmentation and feature extraction to achieve high recognition accuracy on isolated characters. Inam et al. [26] used a feed-forward neural network with promising results, obtaining 98.3% accuracy. However, the font size and style invariability claim would benefit from more detailed evaluation across different fonts and sizes.

Others have addressed the segmentation challenges for cursive script recognition. Sobia Tariq Javed and Sarmad Hussain [27] applied HMM after ligature segmentation, achieving 92.73% accuracy on printed text. However, variations in shape and similarity between ligature segments may impact recognition accuracy.

Several papers explored the effectiveness of holistic approaches, where ligatures are treated as minimum units for recognition. Israr Uddin et al. [29] used Discrete Wavelet Transform coefficients for HMM training, achieving an average accuracy of 89%. Nizwa Javed et al. [30] used CNN for ligature classification, reaching a recognition rate of 95.3%.

Deep learning techniques have shown promising results in Urdu OCR. Asma Naseer and Kashif Zafar [31] demonstrated that meta-features extracted from thickness graphs outperformed raw image features, achieving an average accuracy rate of 98.08%. Muhammad Jawad Rafeeq et al. [32] augmented the dataset and employed a deep neural network, obtaining 95.02% accuracy.

Segmentation-free approaches have also been explored. Syed Yasser Arafat and Muhammad Javed Iqbal [33] proposed a holistic segmentation-free approach using a double-layer BLSTM with CNN and VGG16 features, achieving 80.46% accuracy for predicting partial character sequences in ligatures.

While these papers demonstrate significant progress in Urdu OCR, it's essential to consider the challenges associated with variations in fonts, font sizes, and handwriting styles, which can significantly affect the performance of OCR systems. Additionally, the performance of the proposed approaches may vary across different datasets and script variations. Future research should focus on developing comprehensive end-to-end OCR systems that address segmentation challenges, script variations, and robustly handle noisy inputs, paving the way for practical and accurate Urdu OCR applications.

3 Research Methodology

The goal of this research is to create a system that can read offline handwritten Isolated Urdu characters. While many researchers have focused on dealing with segmentation challenges, this research aims to encompass all the contributions made in recognizing handwritten Urdu characters. The main objective is to make progress and improve accuracy in the area of urdu OCR. Below is a discussion of the main duties that our system entails:

3.1 Dataset Description

The dataset used in this study consists of two parts: the UHAT (Urdu Handwritten Text) dataset and a self-generated dataset for handwritten Urdu characters and their variants. The UHAT dataset contains images of Urdu characters with a resolution of 28x28 pixels, written by more than 900 individuals. There are 4,880 testing photos and 28,328 training images in this dataset. The UHAT dataset contains 700 training images and 140 testing images for each character.

Additionally, the researchers created their own dataset of handwritten Urdu characters and their alternate starts, middles, and ends. To build this dataset, data was collected from various writers, resulting in a total of 6,300 images. The self-generated dataset is further divided into training and testing sets, with 5,040 images used for training and 1,260 images for testing. By utilizing both the UHAT dataset and their self-generated dataset, the researchers aim to evaluate their proposed OCR system on a diverse and representative collection of handwritten Urdu characters, including variants of character positions. This comprehensive dataset enables thorough testing and validation of the OCR system's performance under various handwriting styles and positions of characters within words.

3.2 Dataset Preprocessing

In the realm of developing effective systems for handwritten Urdu character recognition, preprocessing of data stands as a foundational stage. In the context of Optical Character Recognition (OCR), preprocessing encompasses pivotal tasks such as binarization and normalization. Each algorithm within this preprocessing framework holds distinct prerequisites, vital for achieving the intended outcomes. This section elaborates on the diverse preprocessing steps integral to data preparation:

3.2.1 Image Preprocessing:

Image Binarization:

The commencement of image preprocessing is marked by the crucial process of binarization. Binarization's role is to undertake segmentation, effectively classifying pixels within the image into foreground and background categories. Foreground pixels are representative of white pixels, while their counterparts, the background pixels, embody black pixels within the image. Binarization, based on its approach, can be classified into local and global methods [36]. The local thresholding technique determines a threshold value for each individual pixel, while global thresholding computes a threshold value for the entire image. Leveraging local thresholding maintains essential data nuances by converting the image into grayscale [37]. Alternatively, global thresholding finds its application when a clear and evident distinction between foreground and background pixels is readily apparent [38].

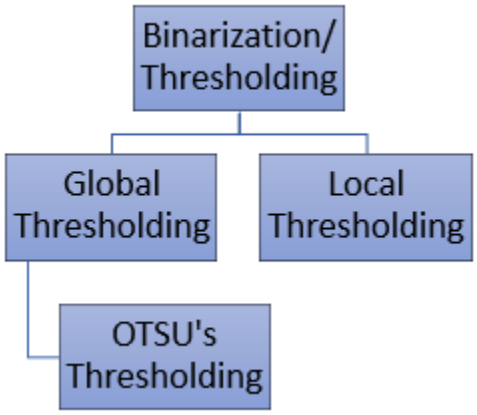

Figure 6: Binarization Techniques

OTSU's Thresholding:

In image processing tasks, the OTSU Thresholding approach has been used to perform automatic global thresholding. OTSU's thresholding is an advanced method of image binarization that effectively separates foreground and background pixels. The range of colors for thresholding is from 0 to 255 Within the landscape of image processing tasks, the utilization of OTSU's Thresholding methodology emerges as a prominent approach for automatic global thresholding. Distinguished as an advanced technique for image binarization, OTSU's thresholding effectively orchestrates the separation of foreground and background pixels within an image. This technique operates within a color spectrum ranging from 0 to 255.

Image Normalization:

A critical process within image preprocessing is image normalization, which serves the purpose of calibrating the intensity levels of input pixels to align them within a standard range [39]. This manipulation finds utility in noise reduction within the data and subsequently enhances the effectiveness of feature extraction. In the realm of input pixels, intensity variation spans from 0 to 255, where the extremities represent white and black pixels, respectively. To mitigate challenges like exploding gradients during the learning phase, a pivotal step is to rescale input images to a condensed range of 0 to 1 before integration into the network.

4 Proposed Architecture

Convolutional Neural Networks (CNNs), which perform classification to extract visual information using a number of hidden layers, are the foundation of our suggested architecture. The three Conv2D layers of the CNN model that we created each include 32, 64, 64, 128, and 128 filters. A MaxPooling2D layer is placed after each Conv2D layer to cut down on the number of parameters. The characteristics are aggregated and then converted into a one-dimensional linear vector using a Flatten layer. The next step is to generate two dense layers with ReLU activation functions, each containing 100 neurons. Finally, an output layer

with the softmax activation function is added to the model, serving as a fully connected layer for classification. The proposed model architecture is depicted in Fig. 7.

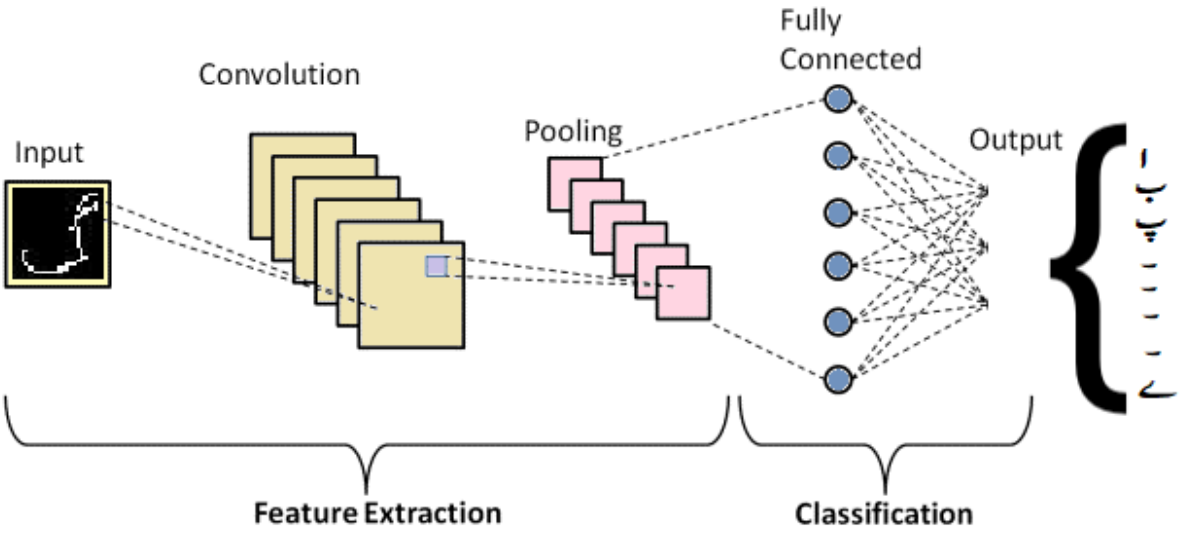

Figure 7: Proposed Architecture

The proposed model is composed of distinct layers, each contributing to the comprehensive process of feature extraction and classification. These layers include the Convolutional Layer, MaxPooling Layer, Dense Layer, and ReLU as the activation function. Their functionalities are elaborated below:

4.1 Convolutional Layer

The Convolutional Layer takes on the pivotal role of initiating feature extraction from the input image. By engaging in a mathematical operation between the input image and a filter matrix of dimension NxN, this layer executes sliding computations over the image. The outcome is a feature map that extracts features such as edges and corners from the image. Various filter sizes, like 3x3, 5x5, or 7x7, can be employed to tailor the convolutional operation's effects.

4.2 Pooling Layer

Post-convolutional processing, the pooling operation comes into play. Its primary function revolves around parameter reduction, thereby contributing to the reduction of computational time. The pooling layer independently samples and compresses both the height and width dimensions of each feature map. Prominent pooling operations encompass max pooling, which selects the highest value, and average pooling, which computes the mean of the elements within a feature map. Conventionally, a 2x2 window is used for pooling.

4.3 Flatten Layer

Subsequent to the pooling layer, the Flatten layer plays a significant role in the model. This layer transforms the pooled feature arrays from a two-dimensional representation into a linear, one-dimensional vector. This transformation readies the flattened features for input into the Fully Connected (FC) layer, where classification processes occur.

4.4 Fully Connected Layer

The penultimate layer before the output layer is the Fully Connected (FC) layer. Here, the pooled data undergoes flattening and proceeds through multiple additional FC layers, wherein further mathematical

computations are performed. This layer's primary objective is to grasp features from the input image and accurately classify them, thus underpinning the classification task of the model.

4.5 Activation Function

The activation function assumes a pivotal role within the hidden layers of the CNN model. Its core function involves determining whether network information should be activated in the forward direction. This step introduces non-linearity to the model. In this study, the ReLU (Rectified Linear Unit) and softmax activation functions are employed:

4.5.1 ReLU (Rectified Linear Unit)

ReLU enhances non-linearity and expedites training. It is widely adopted in CNN models, surpassing other activation functions like softmax in effectiveness.

4.5.2 Softmax

The softmax function finds application in multi-class classification scenarios that employ the categorical cross entropy loss function. It becomes relevant when the classification task involves categorizing more than two classes. Softmax computes relative probabilities for each class, effectively normalizing the output. It operates as a soft version of the argmax function by furnishing indices of the highest value.

4.6 Optimizer

The optimizer is a critical element responsible for diminishing network loss during training. In this study, the "Adam" optimizer, an acronym for Adaptive Moment Estimation, is employed. Adam is an amalgamation of momentum and RMSprop [40] techniques. It is widely embraced within neural networks and effectively calculates an adaptive learning rate.

4.7 Loss Function

The loss function functions as a metric to gauge the magnitude of errors within the prediction model during the training process. In this context, the categorical_crossentropy loss function is utilized. Specifically:

4.7.1 Categorical_crossentropy

Categorical_crossentropy is utilized when dealing with multiclass classification, where there are more than two labels' classes.

4.8 Training Parameters

We separated the UHAT and custom dataset into training and testing images for the CNN model. 80% of the total amount of UHAT and customized training data were used to train the model. During the training procedure, we used the categorical_crossentropy loss function and the Adam optimizer function. One hundred epochs were used for the training.

We monitored the validation loss to assess the training model's performance. The validation loss provides insights into how well the model generalizes to unseen data during the training process. During the training phase, the CNN model learned to extract relevant features from the handwritten Urdu characters and classify them into their respective categories. The use of Adam as the optimizer allowed for adaptive learning rates, optimizing the training process and helping to converge faster.

The categorical_crossentropy loss function was chosen as it is appropriate for multiclass classification tasks, which are required when dealing with more than two classes or labels. By training the model on a combination of UHAT and custom datasets and validating its performance using the validation loss, we aimed to achieve high accuracy and robustness in the recognition of offline handwritten Urdu characters.

The process of training and evaluating the model ensured that it could effectively recognize and classify characters from unseen data with high accuracy.

5 Results and Evaluation

Results from our suggested CNN model are assessed in this section. Below is a list of the systems requirements used for this work:

5.1 System Requirements

Table 1: System requirements

System Processor	2.70 GHz
RAM	8.00 GB
OS	MS Windows 11 Pro
Storage	1TB
Tool	PyCharm Community Edition 2021.3 x64
Language	Python

5.2 Tentative Analysis

In this study, we examined CNN's handwritten Urdu character recognition technique. We trained and tested it multiple times, varying the number of epochs (50, 70, and 100) on the UHAT dataset. We also trained it on a custom dataset of handwritten characters for 100 epochs. We monitored accuracy and loss during training and testing to optimize the model's performance. The results helped refine the CNN model for improved recognition of handwritten Urdu characters.

5.2.1 Loss during Training and Validation

The figures below show the training and validation losses:

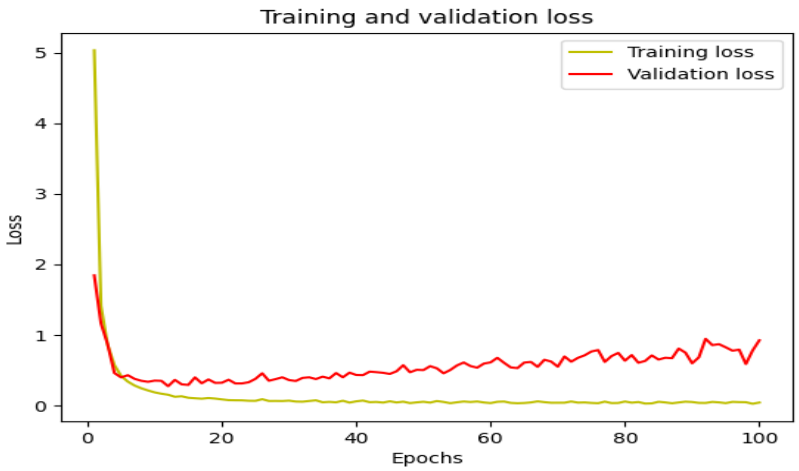

Figure 8: Training and Va1idation Loss on UHAT dataset

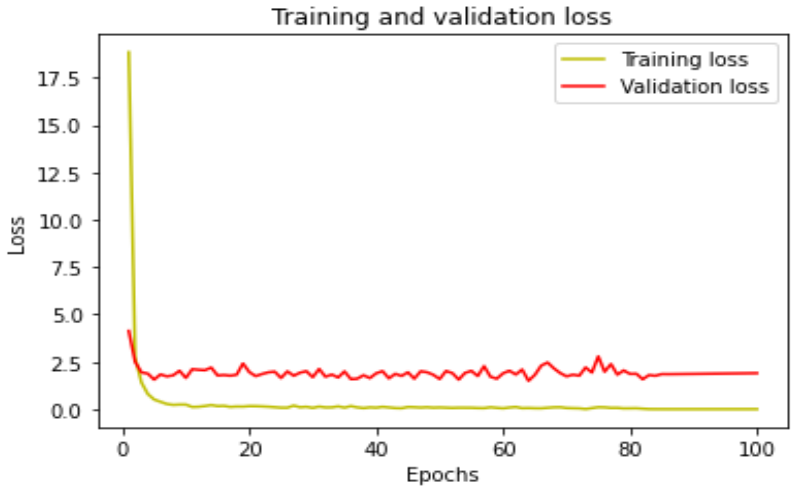

Figure 9: Training and Validation Loss on Custom dataset

5.2.2 Accuracy during Training and Validation

The following figures provide information on training and validation accuracy:

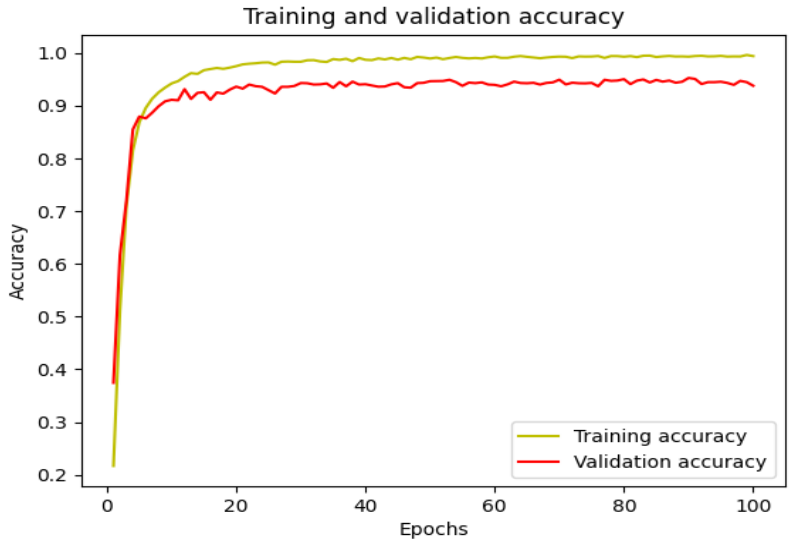

Figure 10: Va1idation and Training Accuracy on UHAT dataset

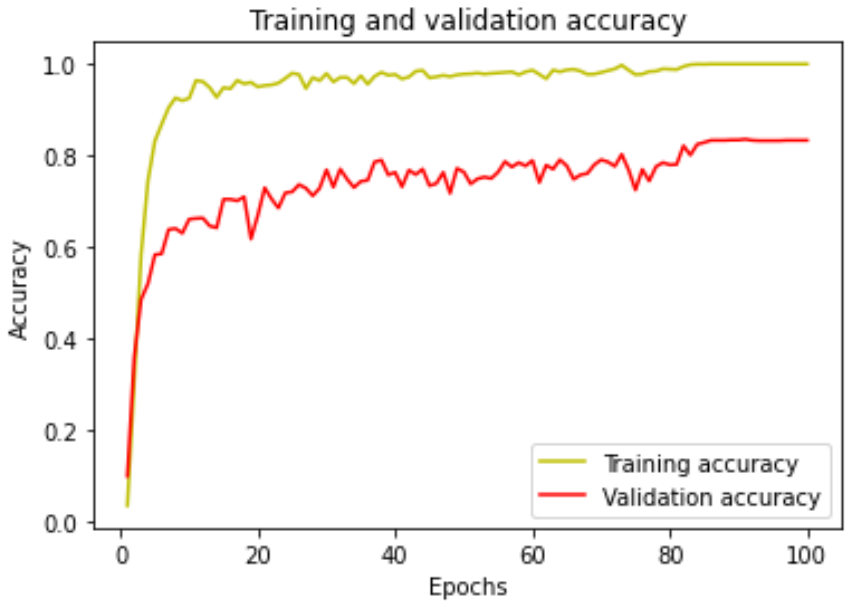

Figure 11: Training and Va1idation Accuracy on Custom dataset

During the initial epochs of training, the model is learning and adjusting its weights to minimize the loss on the validation dataset. As a result, the validation loss decreases, indicating that the model is improving its ability to generalize to unseen data. This is a common behavior during the early stages of training. However, as the number of epochs continues to increase, the model may start to overfit the training data. Overfitting occurs when the model becomes too specialized in fitting the training data, capturing noise and outliers in the process. This can lead to a degradation in performance on the validation dataset because the model's ability to generalize diminishes.

5.2.3 Predictions of Proposed Model

The prediction of our model on UHAT dataset test image is shown in following figure:

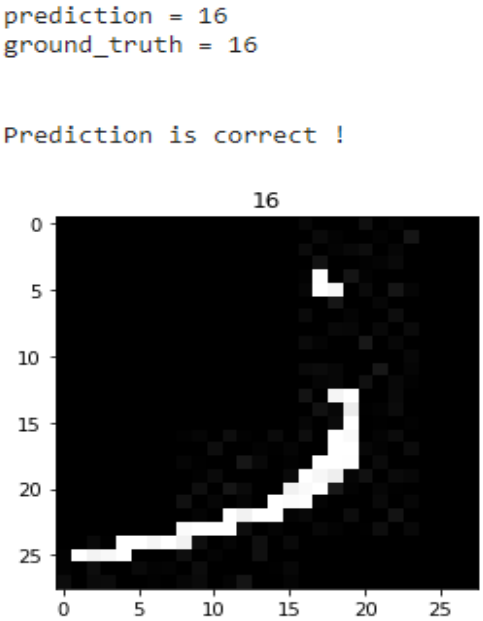

Figure 12: Predictions of Proposed Mode1 on UHAT dataset test image

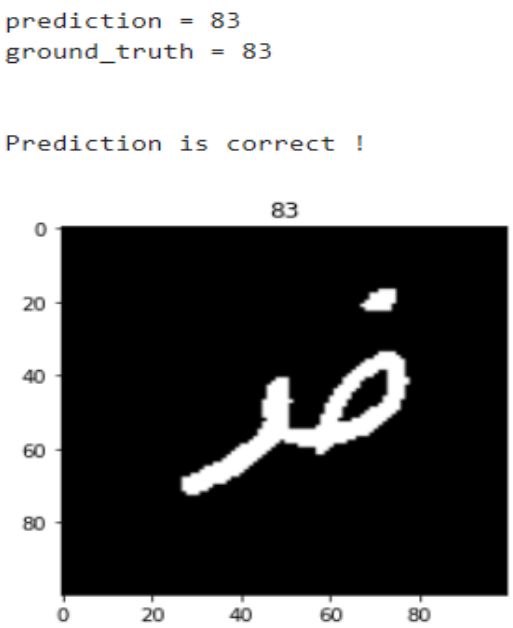

Figure 13: Predictions of Proposed Mode1 on Custom dataset test image

6 Conclusion and Future work

This research paper focused on developing an approach for handwritten character recognition using a CNN model. The results obtained were remarkable, achieving a high recognition rate. Additionally, a custom dataset containing 6,300 images was generated to further enhance the model's performance.

The CNN model was trained on both the UHAT dataset and the custom handwritten dataset, with the training and testing data split at an 80-20 ratio for each dataset. The proposed model achieved recognition rates of 99.19% and 99.39% on the custom and UHAT datasets, respectively. Despite the high recognition rates obtained, there is still room for improvement and future work. One area of focus should be on minimizing research gaps and advancing the proposed approach. Enhancements can be made to incorporate the recognition of combined characters, aiming to achieve even higher recognition rates.

Future work should also explore the possibilities of using larger datasets and incorporating data augmentation techniques to improve the model's robustness and generalization capabilities. Additionally, exploring advanced CNN architectures or other deep learning techniques could lead to further performance improvements. Overall, this research has laid a strong foundation for handwritten character recognition, but continuous efforts and innovative approaches are needed to push the boundaries and achieve outstanding results in this field.

Reference

[1] M. A. U. Rehman, "A New Scale Invariant Optimized Chain Code for Nastaliq Character Representation," 2010 Second International Conference on Computer Modeling and Simulation, 2010, pp. 400-403, doi: 10.1109/ICCMS.2010.493.

[2] S. A. Husain, "A multi-tier holistic approach for Urdu Nastaliq recognition," International Multi Topic Conference, 2002. Abstracts. INMIC 2002., 2002, pp. 84-84, doi: 10.1109/INMIC.2002.1310191.

[3] S. T. Javed and S. Hussain, "Improving Nastalique specific pre-recognition process for Urdu OCR," 2009 IEEE 13th International Multitopic Conference, 2009, pp. 1-6, doi: 10.1109/INMIC.2009.5383111.

[4] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, “Segmentation Free Nastalique Urdu OCR,” International Journal of Computer and Information Engineering, vol. 4, no. 10, pp. 1514–1519, Oct. 2010.

[5] N. Sabbour and F. Shafait, “A segmentation-free approach to Arabic and Urdu OCR,” NASA ADS, vol. 8658, p. 86580N, Jan. 2013, doi: 10.1117/12.2003731.

[6] Q. Akram, S. Hussain, F. A. Shafiq-ur-Rehman, and M. Saeed, “Framework of Urdu Nastalique Optical Character Recognition System,” www.semanticscholar.org, 2014.

[7] T. Ali, T. Ahmad and M. Imran, "UOCR: A ligature based approach for an Urdu OCR system," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 388-394.

[8] Q. U. A. Akram and S. Hussain, "Ligature-based font size independent OCR for Noori Nastalique writing style," 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), 2017, pp. 129-133, doi: 10.1109/ASAR.2017.8067774.

[9] Q. u. A. Akram, S. Hussain, A. Niazi, U. Anjum and F. Irfan, "Adapting Tesseract for Complex Scripts: An Example for Urdu Nastalique," 2014 11th IAPR International Workshop on Document Analysis Systems, 2014, pp. 191-195, doi: 10.1109/DAS.2014.45.

[10] Z. Ahmed, K. Iqbal, I. Mehmood and M. A. Ayub, "Ligature Analysis-based Urdu OCR Framework," 2017 International Conference on Frontiers of Information Technology (FIT), 2017, pp. 87-92, doi: 10.1109/FIT.2017.00023.

[11] I. Ahmad, X. Wang, R. Li, M. Ahmed and R. Ullah, "Line and Ligature Segmentation of Urdu Nastaleeq Text," in IEEE Access, vol. 5, pp. 10924-10940, 2017, doi: 10.1109/ACCESS.2017.2703155.

[12] I. Ud Din, I. Siddiqi, S. Khalid, and T. Azam, “Segmentation-free optical character recognition for printed Urdu text,” EURASIP Journal on Image and Video Processing, vol. 2017, no. 1, Sep. 2017, doi: 10.1186/s13640-017-0208-z.

[13] K. U. U. Rehman and Y. D. Khan, "A Scale and Rotation Invariant Urdu Nastalique Ligature Recognition Using Cascade Forward Backpropagation Neural Network," in IEEE Access, vol. 7, pp. 120648-120669, 2019, doi: 10.1109/ACCESS.2019.2936363.

[14] S. A. Malik, M. Maqsood, F. Aadil, and M. F. Khan, “An Efficient Segmentation Technique for Urdu Optical Character Recognizer (OCR),” Lecture Notes in Networks and Systems, pp. 131–141, Feb. 2019, doi: 10.1007/978-3-030-12385-7_11.

[15] U. Pal and A. Sarkar, "Recognition of printed Urdu script," Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., 2003, pp. 1183-1187, doi: 10.1109/ICDAR.2003.1227844.

[16] F. Iqbal, A. Latif, N. Kanwal and T. Altaf, "Conversion of urdu nastaliq to roman urdu using OCR," The 4th International Conference on Interaction Sciences, 2011, pp. 19-22.

[17] I. K. Pathan, A. A. Ali, and R. J. Ramteke, ‘‘Recognition of offline handwritten isolated Urdu character,’’ Adv. Comput. Res., vol. 4, no. 1, pp. 117–121, 2012.

[18] S. Hussain, S. Ali, and Q. ul A. Akram, “Nastalique segmentation-based approach for Urdu OCR,” International Journal on Document Analysis and Recognition (IJDAR), vol. 18, no. 4, pp. 357–374, Aug. 2015, doi: 10.1007/s10032-015-0250-2.

[19] A. F. Ganai and F. R. Lone, "Character segmentation for Nastaleeq URDU OCR: A review," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 1489-1493, doi: 10.1109/ICEEOT.2016.7754931.

[20] A. F. Ganai and A. Koul, "Projection profile based ligature segmentation of Nastaleeq Urdu OCR," 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), 2016, pp. 170-175, doi: 10.1109/ISCBI.2016.7743278.

[21] S. Habib, M. K. Shukla and R. Kapoor, "OCR Recognition System for Degraded Urdu and Devnagari Script," 2019 International Conference on contemporary Computing and Informatics (IC3I), 2019, pp. 245-251, doi: 10.1109/IC3I46837.2019.9055519.

[22] A. Ul-Hasan, S. B. Ahmed, F. Rashid, F. Shafait and T. M. Breuel, "Offline Printed Urdu Nastaleeq Script Recognition with Bidirectional LSTM Networks," 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 1061-1065, doi: 10.1109/ICDAR.2013.212.

[23] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, “Handwritten Urdu character recognition using one-dimensional BLSTM classifier,” Neural Computing and Applications, vol. 31, no. 4, pp. 1143–1151, Aug. 2017, doi: 10.1007/s00521-017-3146-x.

[24] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, S. H. Shirazi, and M. I. Razzak, “Urdu Nasta’liq text recognition system based on multi-dimensional recurrent neural network and statistical features,” Neural Computing and Applications, vol. 28, no. 2, pp. 219–231, Sep. 2015, doi: 10.1007/s00521-015-2051-4.

[25] S. Naz et al., “Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks,” Neurocomputing, vol. 177, pp. 228–241, Feb. 2016, doi: 10.1016/j.neucom.2015.11.030.

[26] S. Naz et al., “Zoning Features and 2DLSTM for Urdu Text-line Recognition,” Procedia Computer Science, vol. 96, pp. 16–22, Jan. 2016, doi: 10.1016/j.procs.2016.08.084.

[27] S. Naz et al., “Urdu Nastaliq recognition using convolutional–recursive deep learning,” Neurocomputing, vol. 243, pp. 80–87, Jun. 2017, doi: 10.1016/j.neucom.2017.02.081.

[28] I. Shamsher, Z. Ahmad, J. K. Orakzai and A. Adnan, “OCR for printed Urdu script using feed forward neural network,” in Proc. of World Academy of Science, Engineering and Technology, vol. 23, pp. 172–175, 200.

[29] S. T. Javed and S. Hussain, “Segmentation Based Urdu Nastalique OCR,” Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 41–49, 2013, doi: 10.1007/978-3-642-41827-3_6.

[30] Q. u. A. Akram, S. Hussain, A. Niazi, U. Anjum and F. Irfan, "Adapting Tesseract for Complex Scripts: An Example for Urdu Nastalique," 2014 11th IAPR International Workshop on Document Analysis Systems, 2014, pp. 191-195, doi: 10.1109/DAS.2014.45.

[31] I. Uddin, I. Siddiqi and S. Khalid, "A Holistic Approach for Recognition of Complete Urdu Ligatures Using Hidden Markov Models," 2017 International Conference on Frontiers of Information Technology (FIT), 2017, pp. 155-160, doi: 10.1109/FIT.2017.00035.

[32] N. Javed, S. Shabbir, I. Siddiqi and K. Khurshid, "Classification of Urdu Ligatures Using Convolutional Neural Networks - A Novel Approach," 2017 International Conference on Frontiers of Information Technology (FIT), 2017, pp. 93-97, doi: 10.1109/FIT.2017.00024.

[33] A. Naseer and K. Zafar, “Comparative Analysis of Raw Images and Meta Feature based Urdu OCR using CNN and LSTM,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 1, 2018, doi: 10.14569/ijacsa.2018.090157.

[34] M. J. Rafeeq, Z. ur Rehman, A. Khan, I. A. Khan, and W. Jadoon, “Ligature categorization based Nastaliq Urdu recognition using deep neural networks,” Computational and Mathematical Organization Theory, vol. 25, no. 2, pp. 184–195, Apr. 2018, doi: 10.1007/s10588-018-9271-y.

[35] S. Y. Arafat and M. J. Iqbal, "Two Stream Deep Neural Network for Sequence-Based Urdu Ligature Recognition," in IEEE Access, vol. 7, pp. 159090-159099, 2019, doi: 10.1109/ACCESS.2019.2950537.

[36] A. U. Rehman and S. U. Hussain, “Large Scale Font Independent Urdu Text Recognition System,” www.arxiv-vanity.com, May 2020, Accessed: Jun. 20, 2022.

[37] O. Singh, O. James, and T. Sinam, “Local Contrast and Mean based Thresholding Technique in Image Binarization,” International Journal of Computer Applications (0975 – 8887) Volume 51– No.6, Aug. 2012, Accessed: Jun. 20, 2022.

[38] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh, “A New Local Adaptive Thresholding Technique in Binarization,” arXiv:1201.5227 [cs], Jan. 2012, Accessed: Jun. 20, 2022.

[39] S. N and V. S, “Image Segmentation By Using Thresholding Techniques For Medical Images,” Computer Science & Engineering: An International Journal, vol. 6, no. 1, pp. 1–13, Feb. 2016, doi: 10.5121/cseij.2016.6101.

[40] S.-C. Pei and C.-N. Lin, “Image normalization for pattern recognition,” Image and Vision Computing, vol. 13, no. 10, pp. 711–723, Dec. 1995, doi: 10.1016/0262-8856(95)98753-g.