Computers, Materials & Continua
DOI: 10.32604/cmc.2022.029314
Article

Tech Science Press

Recognition of Urdu Handwritten Alphabet Using Convolutional Neural Network (CNN)

Gulzar Ahmed[1], Tahir Alyas[2], Muhammad Waseem Iqbal[3,*], Muhammad Usman Ashraf[4], Ahmed Mohammed Alghamdi[5], Adel A. Bahaddad[6] and Khalid Ali Almarhabi[7]

[1]Department of Computer Science, University of South Asia, Lahore, 54000, Pakistan
[2]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[3]Department of Software Engineering, Superior University, Lahore, 54000, Pakistan
[4]Department of Computer Science, GC Women University, Sialkot, 53310, Pakistan
[5]Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, 21493, Saudi Arabia
[6]Department of Information System, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[7]Department of Computer Science, College of Computing in Al-Qunfudah, Umm Al-Qura University, Makkah, 24381, Saudi Arabia
*Corresponding Author: Muhammad Waseem Iqbal. Email: waseem.iqbal@superior.edu.pk

Received: 01 March 2022; Accepted: 26 April 2022

Abstract: Handwritten character recognition systems are used in every field of life nowadays, including shopping malls, banks, educational institutes, etc. Urdu is the national language of Pakistan, and it is the fourth spoken language in the world. However, it is still challenging to recognize Urdu handwritten characters owing to their cursive nature. Our paper presents a Convolutional Neural Networks (CNN) model to recognize Urdu handwritten alphabet recognition (UHAR) offline and online characters. Our research contributes an Urdu handwritten dataset (aka UHDS) to empower future works in this field. For offline systems, optical readers are used for extracting the alphabets, while diagonal-based extraction methods are implemented in online systems. Moreover, our research tackled the issue concerning the lack of comprehensive and standard Urdu alphabet datasets to empower research activities in the area of Urdu text recognition. To this end, we collected 1000 handwritten samples for each alphabet and a total of 38000 samples from 12 to 25 age groups to train our CNN model using online and offline mediums. Subsequently, we carried out detailed experiments for character recognition, as detailed in the results. The proposed CNN model outperformed as compared to previously published approaches.

Keywords: Urdu handwritten text recognition; handwritten dataset; convolutional neural network; artificial intelligence; machine learning; deep learning

1 Introduction

Handwritten character recognition (HCR) is a fascinating research topic. However, it remains a challenging endeavour in pattern recognition and image processing because of some languages' cursive and longhand nature [1]. Language is essential for our daily communication and the execution of vital activities. Text recognition systems enable human interaction with machines to produce effective outcomes. HCR systems are used in every field of life that requires human and machine interaction. Researchers and scientists are continuously working on new methods and techniques to reduce the processing time to recognize handwritten work and provide more accurate predictions [2]. Typically, HCR can be grouped into two main types: online and offline techniques. In an offline system, the recognition is performed by scanning the handwritten images optically using Optical character recognition (OCR) from a static surface [3].

In an online system, successive points are presented as the order of strokes and time in two-dimensional coordinates (e.g., x-axis and y-axis) to identify the characters [4]. In this research, UHCR system is proposed to enhance the efficiency and accuracy of Urdu alphabet recognition. Urdu is a sophisticated language derived from various languages, such as Pushto, Dari, Arabic, Persian, and Turkish. The word "Urdu" is derived from the Turkish language, meaning "Legion" or "Army". Urdu adopts the cursive style of the Arabic language, which is known as the "Naskh" script. However, Urdu has its font, ligatures, characters, and symbols that are more difficult to recognize, a style often called "Nastalique" script [5].

Although Urdu is quite similar to the Arabic language and can be written using the Arabic script [6], it has more characters than the Arabic language although Urdu is quite similar to the Arabic language and can be written using the Arabic script [6], it has more characters than the Arabic language. Furthermore, Urdu differentiates between the same word and recognizes that word using multiple views at different positions (e.g., isolated, first, middle and last). Furthermore, Urdu differentiates between the same word and recognizes that word using multiple views at different positions (e.g., isolated, first, middle and last). The Urdu language consists of 38 basic alphabets (i.e., "Haroof e Thajji") [7]. Like Arabic, the Urdu alphabets are written from right to left [8].

The foremost step in Natural language processing (NLP) is resource development, and the Urdu language resource development began with Urdu zabta takhti (UZT) where the standard codes were designed for Urdu language alphabets and subsequently accepted by the government of Pakistan [9]. It is cumbersome to produce an appropriate handwritten character dataset and train it to use machine learning and deep learning techniques to recognize and classify the characters. The classification task in computing vision was performed using different machine learning models, such as the multi-layer network [10].

However, the performance of Machine Learning (ML) or Deep Learning (DL) models is highly reliant on the assortment and collection of better representative features [11]. On the other hand, deep neural networks do not require any feature to be defined explicitly since Deep neural network (DNN) generate raw pixels and use the best features to classify the input into different classes [12]. A DNN consists of multiple nonlinear hidden layers, where trainable parameters and connections are usually large [13].

Such networks require many instances to prevent overfitting and are generally not easy to train. However, one genre of DNN that is comparatively simpler to train is called the convolutional neural network [14]. CNN is typically used for speech recognition besides image recognition [15]. CNN can model compound and complex nonlinear connections between the inputs and outputs. It introduces the local receptive field concepts, replication, weight, and temporal subsampling, providing some degree of distortion and shift-invariance. However, CNN with the same depth has less trainable parameters than a fully connected feedforward network [16].

Tab. 1 shows the basic Urdu alphabet and its glyphs, international phonetic alphabet, Nastaliq, full diacritics, and alphabet names. A stroke is a continuous movement of a stylus or finger over the surface of a touch screen or touchpad. When the stylus or finger touches the screen, the stroke starts, and when it leaves the screen, the stroke ends. Each possible letter is constructed from one or many strokes. The stroke can be of two types, primary or secondary, as shown in Tab. 1.

Table 1: Urdu alphabet glyphs

Num	Glyphs	International phonetic alphabet (IPA)	Nastaliq	Full diacritics	Character names
1	ا	/ɑː//ā//ʔ//–/	الف	الِف	alif/alef
2	آ	/ɑː/ /ā/	الف مدہ	الِف مدِہ	alif maddah
3	ب	/b/	بے	بے	bē/beh
4	پ	/p/	پے	پے	Pē/peh
5	ت	/t̪/	تے	تے	tē/the
6	ٹ	/ʈ/	ٹے	ٹے	ṭē/tteh
7	ث	/s/	ثے	ثے	sē/s̱ē
9	ج	/d͡ʒ/	جیم	جِیم	jīm/jeem
10	چ	/t͡ʃ/	چے	چے	čē/cē/tcheh
11	ح	/ɦ/	بڑی حے	بڑِی حے	baṛī ħē/baṛī ḥē
12	خ	/x/	خے	خے	xē/khē /khah/k̲h̲ē
13	د	/d/	دال	دال	dāl/dal
14	ڈ	/ɖ/	ڈال	ڈال	ḍāl/ddal
15	ذ	/z/	ذال	ذال	zāl/z̤āl
16	ر	/r/	رے	رے	rē/reh
17	ڑ	/ɽ/	ڑے	ڑے	ṛē/rreh
18	ز	/z/	زے	زے	zē/zey
19	ژ	/ʒ/	ژے	ژے	žē/zhē
20	س	/s/	سین	سِین	sīn/seen
21	ش	/ʃ/	شین	شِین	šīn/shīn/sheen
22	ص	/s/	صاد	صِواد	swād/sad/ṣwād
23	ض	/z/	ضاد	ضِواد	żwād/z̤wād
25	ط	/t/	طوے	طوئے	tōʼē/tah/tōē/t..oʼē/toy
26	ظ	/z/	ظوے	ظوئے	zōʼē/zōē/z..oʼē
27	ع	/ɑː/ /oː/ /eː//ʔ/ /ʕ/ /Ø/	عین	عِین	‘ain/ain [1] ayn
28	غ	/ɣ/	غین	غِین	ğain/ghain [1]/g̲h̲ain
29	ف	/f/	فے	فے	fē/feh
30	ق	/q/	قاف	قاف	qāf/qaf
31	ک	/k/	کاف	کاف	kāf/k̲ā̲f̲
32	گ	/g/	گاف	گاف	gāf/gaf
33	ل	/l/	لام	لام	lām/lam

(Continued)

Table 1: Continued

Num	Glyphs	International phonetic alphabet (IPA)	Nastaliq	Full diacritics	Character names
34	م	/m/	میم	مِيم	mīm/meem
35	ن	/n/ /ɲ//ɳ/ /ŋ/	نون	نوِن	nūn/noon
36	و	/ʋ/ /ʊ/ /uː/ /oː/ /ɔː/	واؤ	واؤ	waw/wā’o
37	ہ	/ɦ/ /ɑː/ /eː/	گول ہے	گول ہے	gol hē
38	ء	/h/ or /ɦ/	ہمزہ	ہمزہ	Hēmza
37	ی	/j/ /iː/ /ɑː/	چھوٹی يے	چھوٹی يے	choṭī yē
38	ے	/ɛː/ /eː/	بڑی يے	بڑِی يے	yeh barree

A primary stroke is a skeleton or ghost and is the mandatory part of the alphabet. However, s secondary stroke is an additional part of the alphabet, differentiating one alphabet from other alphabets with the same primary stroke. Moreover, alphabets carrying the same primary strokes make secondary strokes mandatory.

Although CNN maintains an inexpensive and small architecture compared to the standard feedforward network with the same depth, CNN requires a considerable amount of computation and a large labelled dataset. CNN has proven to perform the traditional feedforward network in several ordering tasks by introducing a pre-training phase.

In CNN, the gradient descent algorithm is used where weights are randomly initialized to generate results and subsequently updated based on the poor results. There are three convolution layers used in the CNN architecture. The convolutional layer is the core building block of CNN. This layer bears the central part of the network computing load and performs the dot product between two matrices. Here, one matrix is a set of learnable parameters, a kernel, and the other matrix of the input image of the Urdu alphabet dataset. After that, fully connected layers are added, and the last layer is used to classify Urdu characters.

The background studies reveal that little work has been conducted on the Urdu language in computer vision compared to other languages. Despite millions of people speaking Urdu, no efficient technique for recognizing Urdu handwritten characters using images has been developed. Using multiple machine learning and deep learning techniques, this research proposes an Urdu handwritten character recognition model using image processing. The primary goal of this work is to conduct a comparative analysis of Urdu handwritten characters using an image collection. Tab. 2 shows the primary and secondary parts of alphabets.

Table 2: Primary and secondary parts of alphabets

Num	Alphabets	Primary strokes	Secondary strokes
1	آ	ا	~
2	ب	ٮ	.
3	پ	ٮ	∵

(Continued)

Table 2: Continued

Num	Alphabets	Primary strokes	Secondary strokes
4	ت	ٮ	..
5	ٹ	ٮ	ط
6	ث	ٮ	∴
7	ج	ح	.
8	چ	ح	∵
9	خ	ح	.
10	ڈ	د	ط
11	ذ	د	.
12	ڑ	ر	ط
13	ز	ر	.
14	ژ	ر	∴
15	ش	س	∴
16	ض	ص	.
17	ظ	ط	.
18	غ	ع	.
19	ف	ڡ	.
20	ق	ٯ	..
21	گ	ک	/
22	ن	ں	.

Because it has a different representation style, character recognition of cursive or a running hand script is considered a challenging undertaking. Urdu was derived from Arabic writing, which is why it is so similar to Arabic script. It faces the same obstacles and complications as Arabic script, but with a higher level of intensity. Urdu is written in a variety of forms, but the Nastalik script is the most often used.

2 Related Work

Exhaustive research was performed on handwritten recognition for several languages to empower easy human-machine interaction [17]. To this end, many ML and DL methods were planned and proposed, such as Support vector machine, DNN. Moreover, to enhance and improve the accuracy and efficiency of character recognition, many problems were encountered and solved in various ways like data augmentation, noise removing, etc. Numerous researchers have used different methods to solve Arabic and Urdu character recognition. For example, Ahmad T. Al-Taani presented a method to recognize online Arabic handwritten characters using Decision trees (DT) and Structural features (SF). This task was divided into three steps. First, users wrote the characters on a window screen, where the pixels' coordinates forming each character were captured and stored in a dedicated array. Second, a box of 5 × 5 dimensions was drawn around the character, and five features were extracted from the characters. Third, the recognition of characters was performed using decision trees.

Somaya Alma'adeed [18] presented a method to recognize offline handwritten Arabic characters using the Hidden markov model (HMM). Their system classified handwritten words from images after detailed experiments using the HMM. The first step removed variations from the image, then the classification process based on HMM is used. Novel method for recognizing online Urdu handwritten characters using geometric invariants like cosine angles, discrete Fourier, self-intersection, inflexion points, radial features, convex hull, and grid features. The hand linear support vector machine removed the noise distortion and movement. After training, the support vector machine model achieved a 97% accuracy score on the test dataset.

Ali et al. [19] presented a method for recognizing online handwritten Arabic alphabets using a neuro-fuzzy approach, where a fuzzy neural network classified the characters. After feature extraction of neurophysiological parameters, a fuzzy membership was assigned to each output—unique method for identifying Urdu characters using a Non-negative matrix factorization (NNMF). The authors address the challenges of handwritten numerals using OCR techniques.

Online Urdu character recognition using the single stroke characters and reported state-of-the-art studies for various phases, including image acquisition, segmentation, preprocessing, classification/recognition, post-processing, and feature extraction for the Urdu optical character recognition systems. An analysis of various online and offline approaches for Urdu character recognition is found in the proposed idea that presented in this paper. The system has been analyzed on the factors such as methodology, font, text type, sample recognition level, and accuracy level. The authors provided a historical review and a brief comparison between English, Urdu, and Arabic. Traditional machine learning and deep learning techniques were used for numeral recognition.

This paper introduced the pioneer dataset for Urdu handwritten characters and digits collected from more than 900 people. Ali et al. [20] presented the pioneer dataset and an automatic Urdu character recognition using a convolutional neural network and deep auto-encoder with an 82.7% accuracy on alphabets and digits. It also demonstrated that auto-recognition could be realized using CNN.

ML approaches for Urdu and Nastaliq style handwritten character recognition. Husain et al. [21] presented a method for online Urdu characters recognition using ligature-based identification instead of traditional OCR techniques. Backpropagation was used in a neural network for ligature-based recognition, and some special ligatures (e.g., Tay, Dot, Hamza, Mad, and Diagonals) were defined and associated with the basic ligatures. A word dictionary was used for the validation. Online input was captured using a stylus and tablet, while scanning images captured offline input.

Habib et al. [22] presented a paper on the preprocessing for identifying degraded Devanagari and Urdu printed scripts. The authors aimed to improve the quality of the input by image preprocessing and enhancing the results by removing noise using median and Gaussian filters. Hough transforms, modified wolf's algorithm, shearing, and gaussian filter, binarization, skew detection, and median filter techniques were used for the system.

The issue of automatically recognizing Urdu handwritten numerals and characters is difficult. It has uses in reading postal addresses, processing bank checks, and digitizing and preserving historic handwritten manuscripts. While significant work has been done on automatic recognition of handwritten English characters and other major global languages, the work done on the Urdu language is woefully inadequate.

The Urdu language is extremely important because it is one of the world's major languages and Pakistan's national language. Text in Urdu is similar to text in Arabic and Persian. This paper proposes a system for recognizing Urdu handwritten letters automatically. For Urdu, the task is less explored. One of the main reasons is that there is no dataset for Urdu handwritten text. We propose a new Urdu handwritten numerals and characters dataset to address this. The motivation stems from the lack of a standard collection of Urdu handwritten text that could be used as a starting point for study.

3 Proposed Methodology

The proposed UHAR model empowered with CNN architecture consists of two training and implementation modules: the application and implementation layers. One training and implementation module is application layer as shown in Fig. 1.

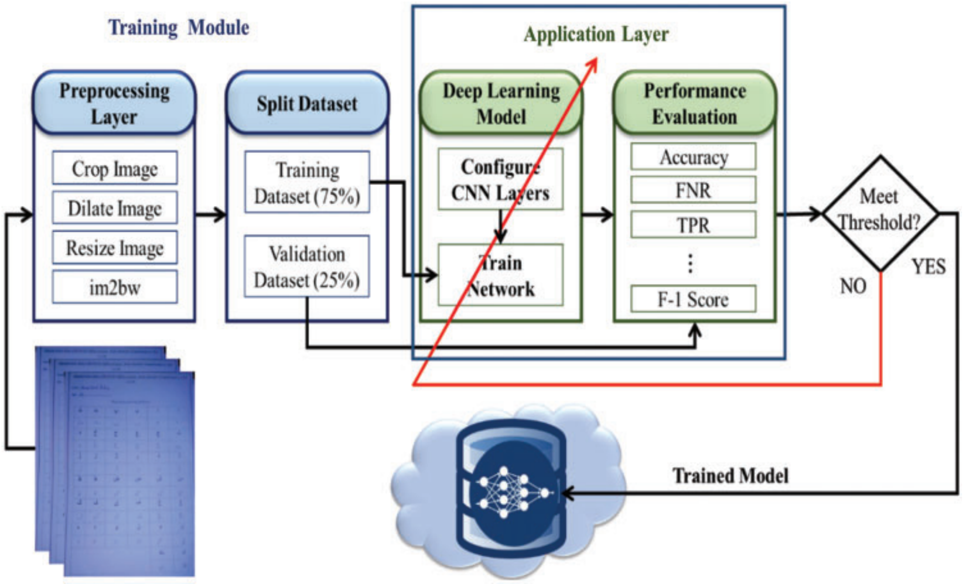


Figure 1: : Training module of the proposed UHAR model

The training module contains the two most essential elements: the data preprocessing and application layers. Different operations are applied to the raw images of Urdu handwritten alphabets in the data preprocessing layer, like cropping, dilation, resizing, and conversion from Red green blue (RGB) to grayscale. In our research, the Urdu handwritten image dataset is generated by filling out forms from different age groups. After applying the preprocessing operations, the dataset is split for training (75%) and validation (25%). The dataset contains a total of 38,000 images, 28,500 of which were used for training and 9,500 images for testing purposes. 75% training data and 25% testing data from each class are taken to avoid the unbalanced data division problem. These divided datasets were sent to the application layer, where the CNN architecture is configured with a specific number of convolutions, pooling, and fully connected layers.

The trained network was evaluated for performance accuracy, False negative rate (FNR), True positive rate (TPR), True negative rate (TNR), Positive predictive value (PPV), Negative predictive value (NPV), False positive rate (FPR), False omission rate (FOR), and F-1 score by using the testing dataset. These metrics are calculated by using the True positive (TP), False positive (FP), True negative (TN), False negative (FN) obtained by the confusion matrix shown in Fig. 1. If the trained network meets the threshold, it is stored in the local and cloud databases for further application. Otherwise, the network is retrained to achieve the performance measure in the retraining process; the layers of CNN are also reconfigured to obtain a better accuracy that meets the target threshold. The second implementation module of the proposed UHAR model collects the Urdu handwritten alphabets from various sources, like classrooms notebooks, using IoT sensors or scanned images. The preprocessing layer preprocesses these inputs. The trained network model is available in the cloud for the implementation module. Anyone can use this trained model according to their requirements. importing It is mandatory to import the trained network models into the implementation module from the cloud for Urdu handwritten alphabets to achieve the predictions. The output layer contains 38 possible outputs, and our CNN model matches the inputs to the best output character class. The second training and implementation module is the implementation layer, as shown in Fig. 2.

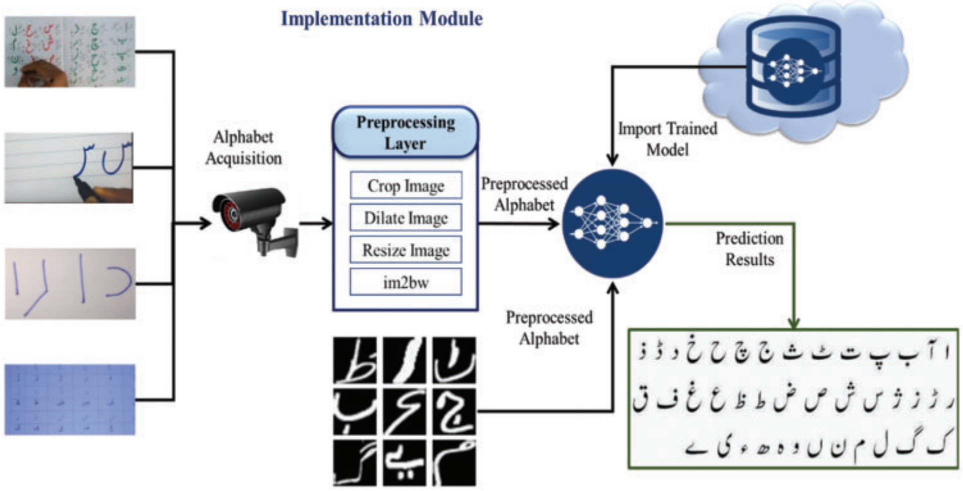
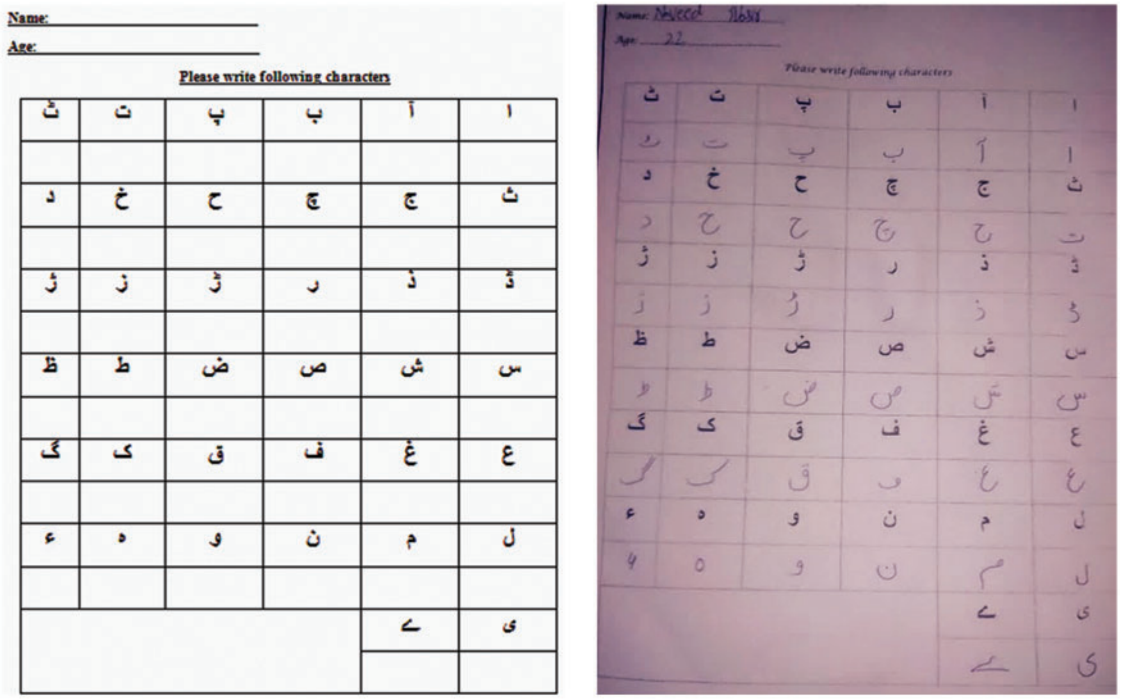


Figure 2: Implementation module of the proposed UHAR model empowered by CNN

Character arrangement and classification is a primary job in any vision problem for image processing and character recognition systems. However, there are no standard datasets for Urdu alphabet recognition publicly available for researchers [23]. In this research, we ought to create our own dataset for training and testing the CNN model. We applied the nonprobability sampling technique [24]. We collated 1000 samples for each alphabet, thus producing a large pool of 38000 samples from age groups ranging from 12 to 25 years old.

We designed a questionnaire named as a handwritten data collector for the proposed CNN-empowered UHAR model. To answer the questionnaires, participants were selected randomly among different age groups (e.g., students from primary school, O-levels, A-levels, and teachers). These all 1000 questionnaires are scan and stored in the computer system. These questionnaires were further cropped into distinct characters and stored into different classes. In preprocessing the images, several operations were performed, like converting images from RGB to grayscale and dilation operations to fill out the gaps within the images [25].

Fig. 3 shows an example of the data collected from the respondents. It was necessary to resize the images for perusal by the CNN architecture. Each image was resized into (28 × 28 × 1) height, width, and channels, respectively.



Name:

Age:

Please write following characters

ٹ	ت	پ	ب	آ	ا
د	خ	ح	چ	ج	ث
ژ	ز	ڑ	ر	ذ	ڈ
ظ	ط	ض	ص	ش	س
گ	ک	ق	ف	غ	ع
ء	ہ	و	ن	م	ل
				ے	ی

Name: Naveed Abbas

Age: 22

Please write following characters

ٹ	ت	پ	ب	آ	ا
ٹ	ت	پ	ب	آ	ا
د	خ	ح	چ	ج	ث
د	خ	ح	چ	ج	ث
ژ	ز	ڑ	ر	ذ	ڈ
ژ	ز	ڑ	ر	ذ	ڈ
ظ	ط	ض	ص	ش	س
ظ	ط	ض	ص	ش	س
گ	ک	ق	ف	غ	ع
گ	ک	ق	ف	غ	ع
ء	ہ	و	ن	م	ل
ء	ہ	و	ن	م	ل
				ے	ی
				ے	ی

Figure 3: Data collection questionnaire (left) and filled form (right)

The CNN architecture consists of the image input layer, Convolutional Layer, Batch Normalization Layer, ReLU Layer, Max Pooling Layer, Fully Connected Layer, SoftMax Layer, and Classification Layer for the proposed UHAR model, as depicted in Fig. 3. The size of each image was set to 28 × 28 × 1 and was used by the input layer. The channel size was set to 1 since images were grayscale. The input layer sends the images to the convolution layer, in which a 3-by-3 filter/kernel size is used, and this filter scans the image to extract the features from the image. After applying the convolution layer, the image output size depends on the filter, padding, and stride sizes.

The height and width of the output image after the convolutional layer is obtained by using Eq. (1), which is the same as the input image size because of the filter (= 3), padding (= 1), and stride (= 1) are used in each convolutional layer. In our costumed CNN architecture, 16, 32, and 64 filters, also known as neurons, are used in the first, second, and third convolution layers, respectively [26].

$$\mathrm{n} + 2\mathrm{p} - \mathrm{k} + 1 = \mathrm{n}$$

$$\mathrm{p} = \frac{\mathrm{k} - 1}{2} \tag{1}$$

The output image size after applying the max-pooling is based on either valid or the same condition. In the case of valid condition, the value of padding used is zero. The same condition is applied to get the same output as the input image size, depending on the specific padding value. Therefore, the padding value is required for this purpose and can be calculated using Eq. (2) [27].

A normalization layer is used to normalize the output from the previous layer and enable the learning process independently as shown in Fig. 4. This layer also scales the input layer, which improves learning efficiency. This process acts as a regularization to avoid overfitting. A Rectified Linear Unit (ReLU) layer [28] is used as an activation function to limit the input elements, with a threshold, converting the elements to zero if the value is less than 0, as shown in Eq. (2).

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

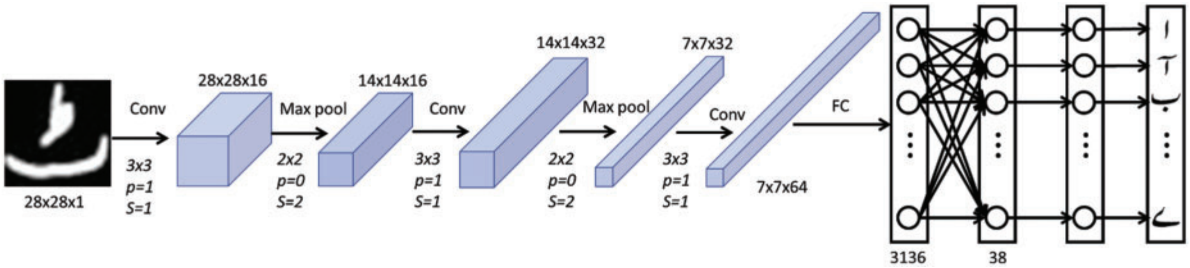


Figure 4: The CNN architecture of the proposed UHAR model

Another building block for CNN is a Pooling Layer. The pooling layer progressively reduces the spatial size of the representation to reduce the amount of computation and parameters in the network [29]. The essential features are fetched, while the irrelevant features and spaces are removed. The max-pooling is the most commonly used approach in pooling in the CNN, with the filter, padding, and strid sizes set to 2, 0, 2 respectively, as shown in Fig. 5. It operates on each layer independently. This process increases the number of filters but without augmenting the computation cost due to the down-sampling [30].

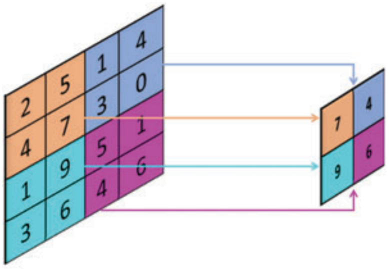


Figure 5: MAX pooling

The size of the last convolution layer is set to 7 × 7 × 74, which is further converted into a 1-dimensional array by flattening. After the flattening process, the convolution layer 7 × 7 × 74 inputs a 3136 array size into the next fully connected layer with 38 neurons, equal to the number of classes in the output layer [31].

The softmax layer is the last activation function used in our CNN architecture to normalize the output of the fully connected layer. SoftMax is used when several more classes are required for prediction. It is also called the normalized exponential function, as shown in Eq. (3), which takes a vector of K real numbers, also known as logits. It normalizes it into a probability distribution consisting of K probability proportion to the exponentials of the input numbers [32].

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{3}$$

The CNN architecture in the proposed UHAR model has three convolution layers trailed by two fully connected layers and accompanied by a sigmoid layer and completed with the output layer. The outcome of a fully connected layer is feedforward to the sigmoid layer, which provides the probability of all outcomes whose sum equals 1. The output layer has a one-dimension vector, i.e., 1 × 38, where 38 is the number of alphabet classes. The application of the implementation module of the proposed UHAR model depends upon the performance metrics calculated from the confusion matrix during the testing phase.

There are nine metrics to inspect the performance of the proposed UHAR model [33–35]. The confusion matrix provides the necessary statistics about the performance of the proposed model, i.e., how many alphabets are correctly and incorrectly classified. The model accuracy is the ratio of the true positive and true negative classifications to the total number of test images, as laid out in Eq. (4).

$$\text{Accuracy} = \frac{(\mathfrak{Tp} + \mathfrak{Tn})}{(\mathfrak{Tp} + \mathfrak{Tn} + \mathfrak{Fp} + \mathfrak{Fn})} \tag{4}$$

Our deployment model also considers the error rate, miss rate, or FNR of the UHAR model. The error rate is calculated by summing the false positive and false negative divided by the total predictions obtained from the confusion matrix, as shown in Eq. (5).

$$\text{FNR} = \frac{(\mathfrak{Fp} + \mathfrak{Fn})}{(\mathfrak{Tp} + \mathfrak{Tn} + \mathfrak{Fp} + \mathfrak{Fn})} \tag{5}$$

The sensitivity or recall or TPR of the proposed UHAR model is calculated using Eq. (6).

$$\text{TPR} = \frac{\mathfrak{Tn}}{(\mathfrak{Tn} + \mathfrak{Fp})} \tag{6}$$

$$\text{FPR} = \frac{\mathfrak{Fp}}{(\mathfrak{Tn} + \mathfrak{Fp})} \tag{7}$$

Eq. (7), is used to calculate the precision or PPV of the proposed UHAR model.

$$\text{PPV} = \frac{\mathfrak{Tp}}{(\mathfrak{Tp} + \mathfrak{FP})} \tag{8}$$

Eq. (8), is used to calculate the NPV performance metric for the proposed UHAR model.

$$\text{Nagative Predictive Value} = \frac{\mathfrak{Tn}}{(\mathfrak{Tn} + \mathfrak{Fn})} \tag{9}$$

False-positive rate (FPR) is calculated by dividing false-positive cases by the sum of true negative and false positive values of the confusion matrix, as shown in Eq. (9).

$$\text{False Omission Rate} = 1 - \text{NPV} \tag{10}$$

FOR is based on the NPV. It is the complement of the NPV, as shown in Eq. (10).

$$\text{F}_1 \text{ Score} = \frac{2}{(\text{Precision}^{-1} + \text{Recall}^{-1})} = 2*\left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{11}$$

F-1 score is an important performance metric; it is based on Precision and Recall and can be calculated by taking the geometric mean between Precision and Recall as shown in Eq. (11).

4 Experiments and Results

We used a dataset of 28,500 and 9,500 images for training and testing our model, respectively. Fig. 6 shows the training result of the proposed UHAR model. There are 2220 iterations used along the x-axis, and ten epochs were utilized in training, with each epoch consisting of 222 iterations. The training accuracy of the proposed model reached 99.80%, and validation accuracy reached 98.61% along the y-axis with a 0.01 learning rate. The dark blue curve represents the smoothed training, whereas the light blue curve with dots denotes training, and finally, the black dotted curve shows the validation accuracy of the proposed UHAR model.

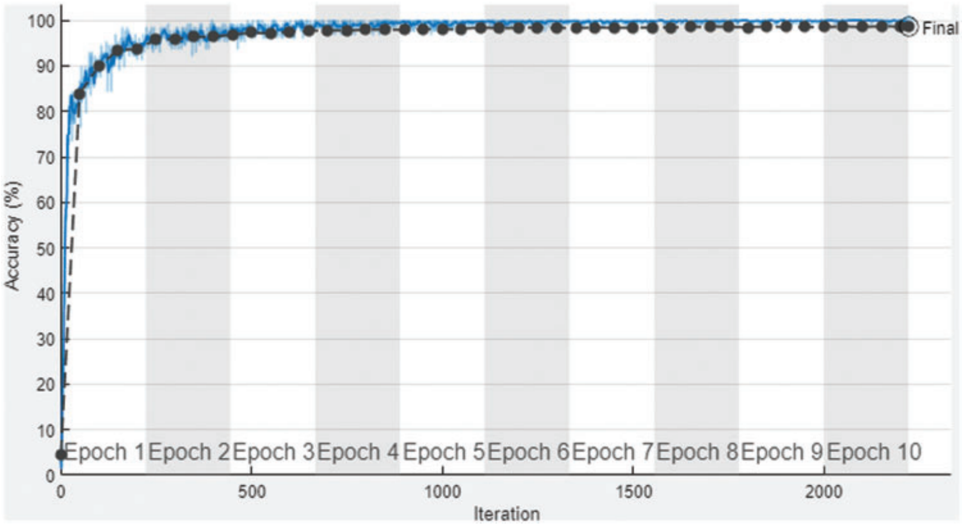


Figure 6: Training and validation accuracy graph

In ML, learning curves are a common diagnostic tool for algorithms that learn progressively from a training dataset. After each update during training, the model may be tested on the training dataset and a hold out validation dataset, and graphs of the measured performance can be constructed to display learning curves. Fig. 7 represents the loss graph of the proposed UHAR model empowering with CNN.

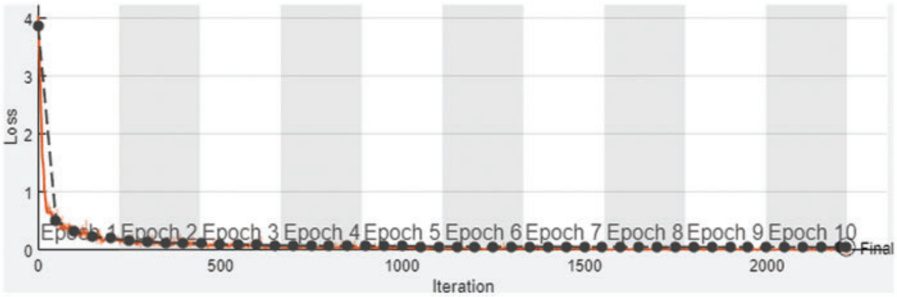


Figure 7: Training and validation loss graph of the proposed UHAR model

The dark yellow smoothed curve represents the training loss, i.e., 0.02%, and the light-yellow curve with a dot showing the actual training. Moreover, the black dotted curve represents the validation loss.

The present state in Fig. 7 shows the learning model in term of loss may be examined at each step of the training algorithm during training. On the training dataset, it may be tested to see how effectively the model is "learning." It may also be tested on a separate validation dataset that isn't included in the training set.

The confusion matrix contains the prediction result in the form of a table. Based on the test dataset, the confusion matrix is generated by using the trained classifier model. In the confusion matrix, true positive, true negative, false positive, and false negative values produced by the classifier model are indicated.

The performance of the trained classifier model is evaluated using the test dataset composed of 9500 instances.

Fig. 8 shows the confusion matrix, representing the results of the proposed UHAR model, taking 9500 test datasets which contain 38 classes, and each class contains 250 alphabets to evaluate the system performance. The actual and predictive classes are shown along x-axis and y-axis, respectively. The true positive value of each class is shown in a diagonal; see Fig. 8.

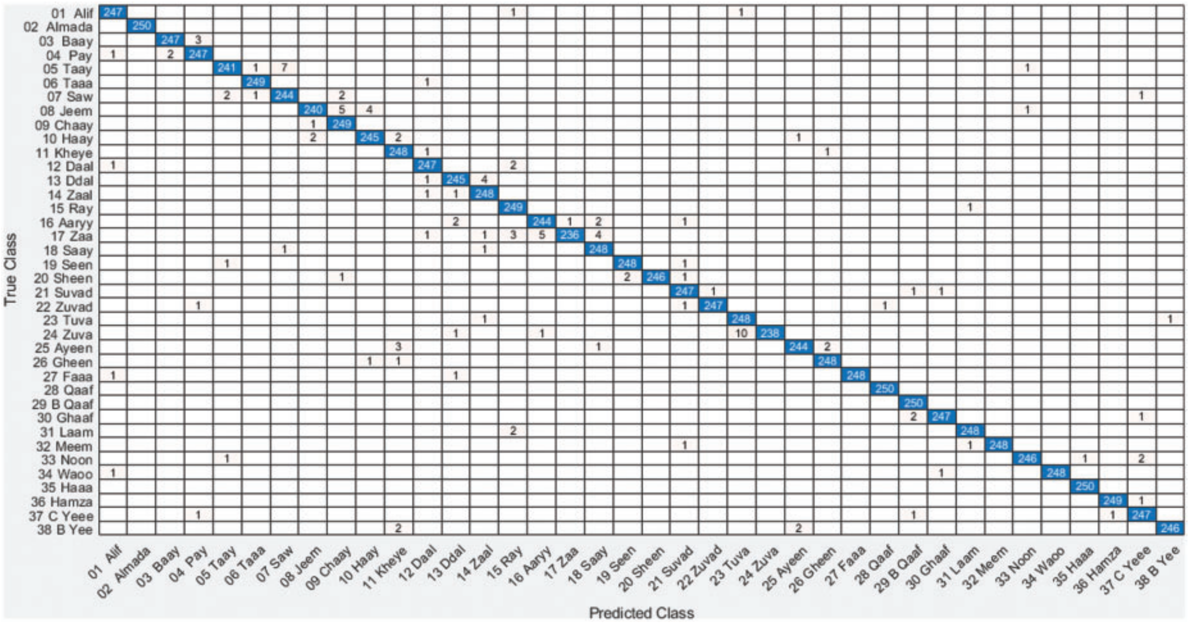


Figure 8: Confusion matrix

Fig. 9 shows the prediction results of the test dataset. The predicted results of the proposed UHAR model are shown in Fig. 10 by taking a random sample from the test dataset. Both actual and predicted alphabets are shown in Fig. 10.

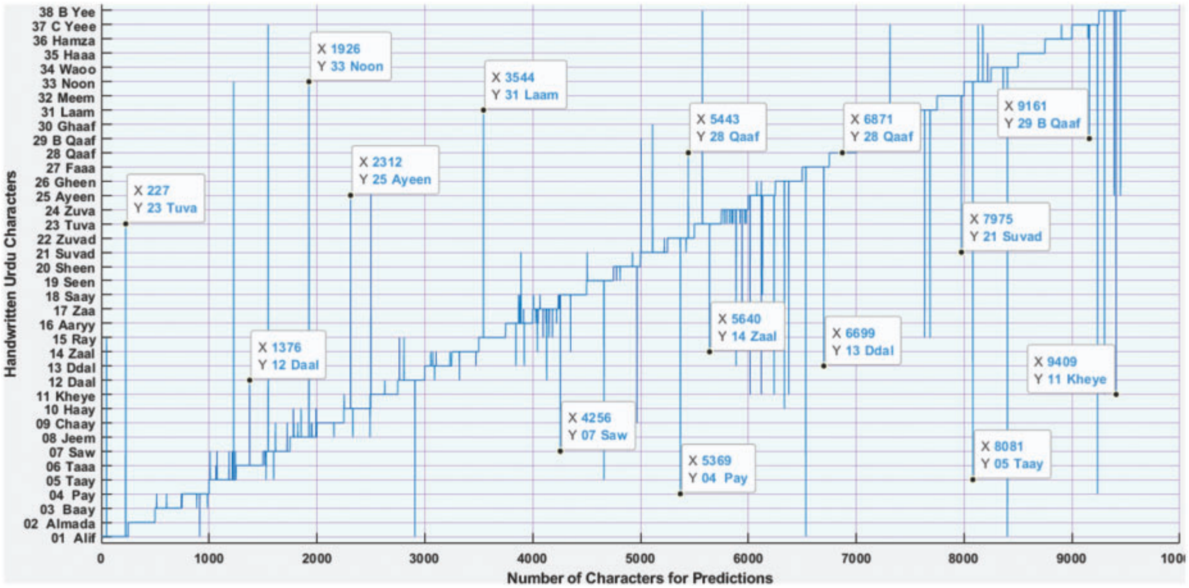


Figure 9: The graph of the prediction test

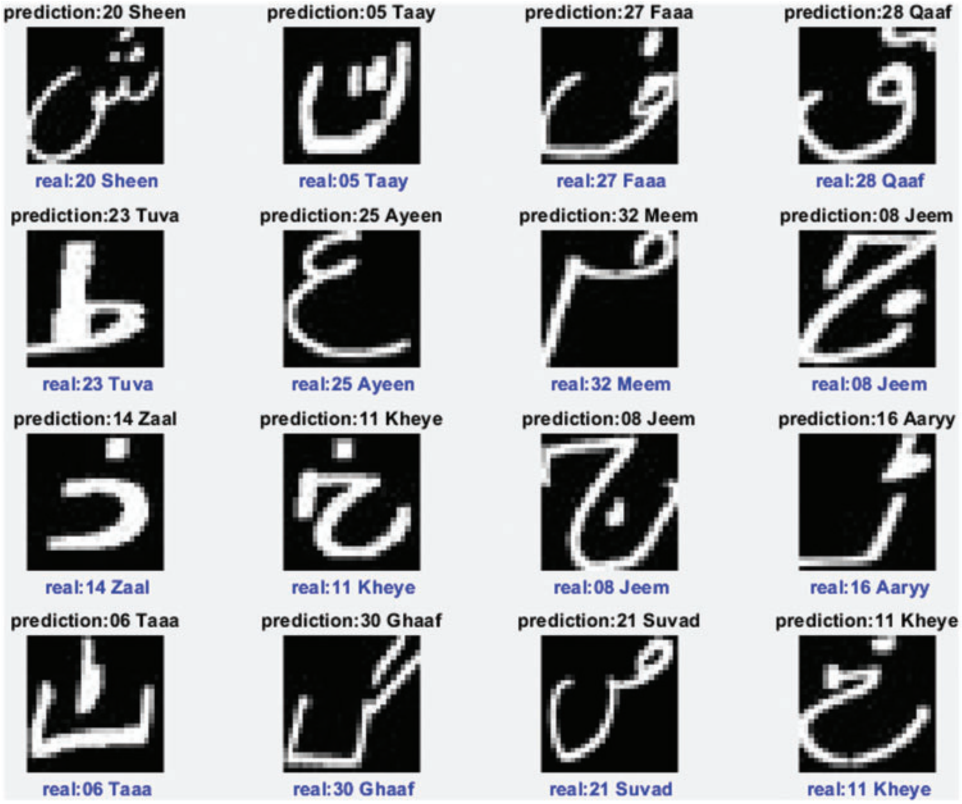


Figure 10: Actual and predicted results

The performance metrics play an essential role in the training and validation of the proposed UHAR model. The acceptance and rejection depend on the validation results from the testing dataset. Tab. 3 shows the accuracy, FNR, TPR, TNR, PPV, NPV, False predictive rate (FPR), FOR, and F1-score performance metrics results. Excellent prediction scores are demonstrated in Tab. 3.

Table 3: Performance metrics

Performance metric	Accuracy	FNR	TPR	TNR	PPV	NPV	FPR	FOR	F1-Score
Validation phase	0.9861 98.61%	0.0005 0.05%	0.9840 98.40%	0.9999 99.99%	0.9960 99.60%	0.9996 99.96%	0.0001 0.01%	0.0004 0.04%	0.9899 98.99%

The comparison of previously published articles with the proposed UHAR model is listed in Tab. 4. The accuracy of our model is 98.61%, higher than existing approaches, while the classification error rate is relatively low, i.e., 1.39%.

Table 4: The comparison of previously published approaches

Published approaches	Accuracy (Percent)	Error rate (Percent)
[25]	80.00%	20.00%
[26]	93.37%	6.63%
[27]	94.29%	5.71%
[28]	95.16%	4.84%
[29]	95.80%	4.20%
[30]	96.91%	3.09%
[31]	97.40%	2.60%
[32]	97.70%	2.30%
[33]	97.93%	2.07%
[34]	98.16%	1.84%
Proposed UHAR model	98.61%	1.39%

5 Conclusion

Handwritten Urdu Alphabet Recognition remains one of the challenging research problems despite the advancements of machine learning techniques. Our proposed UHAR model empowered with CNN provides better accuracy scores than previously published research works on the classification of handwritten Urdu characters. This research created a new Handwritten Urdu Alphabet dataset of 38 classes, where each class has 1000 handwritten images, from different age groups of students and teachers. We applied the CNN architecture for image classification. Four convolution layers were used in the proposed UHAR model, producing an excellent classification accuracy of 98.61% and an error rate of 1.39%. In the future work, we will apply transfer learning approaches and data augmentation to ameliorate the accuracy of our model.

Acknowledgement: The authors therefore, gratefully acknowledge DSR technical and financial support.

Funding Statement: This project was funded by the Deanship of Scientific Research (DSR), King Abdul-Aziz University, Jeddah, Saudi Arabia under Grant No. (RG-11-611-43).

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

References

[1] T. A. Rana, B. Bakht, M. Afzal, N. A. Mian, M. W. Iqbal *et al.,* "Extraction of opinion target using syntactic rules in Urdu text," *Intelligent Automation & Soft Computing (IASC)*, vol. 29, no. 3, pp. 839–853, 2021.

[2] T. Al-Taani and S. Al-Haj, "Recognition of on-line Arabic handwritten characters using structural features," *Pattern Recognition*, vol. 5, no. 1, pp. 23–37, 2010.

[3] A. Choudhary, R. Rishi and S. Ahlawat, "Off-line handwritten character recognition using features extracted from binarization technique," *AASRI Procedia*, vol. 4, no. 1, pp. 306–312, 2013.

[4] P. Anita and D. Singh, "Handwritten English character recognition using neural network," *International Journal of Computer Science & Communication*, vol. 1, no. 2, pp. 141–144, 2010.

[5] A. Amin, T. A. Rana, N. A. Mian, M. W. Iqbal, A. Khalid *et al.,* "Top-rank: A novel unsupervised approach for topic prediction using keyphrase extraction for Urdu documents," *IEEE Access*, vol. 8, no, 9, pp. 212675–212686, 2020.

[6] M. Alhawarat and M. Hegazi, "Revisiting k-means and topic modeling, a comparison study to cluster Arabic documents," *IEEE Access*, vol. 6, no. 1, pp. 42740–42749, 2018.

[7] R. Rizvi, A. Sagheer, K. Adnan and A. Muhammad, "Optical character recognition system for nastalique Urdu-like script languages using supervised learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 10, pp. 1953004, 2019.

[8] A. U. Rehman, A. H. Khan, M. Aftab, Z. Rehman, and M. A. Shah, "Hierarchical topic modeling for Urdu text articles," in *25th Int. Conf. on Automation and Computing (ICAC)*, Lancaster, UK, pp. 1–6, 2019.

[9] K. Mehmood, D. Essam and K. Sha, "Sentiment analysis system for Roman Urdu," in *Intelligent Computing*, Cham, Switzerland, Springer, pp. 29–42, 2019.

[10] C. Kang and C. He, "SAR image classification based on the multi-layer network and transfer learning ofmid level representations," in *IEEE International Geoscience and Remote Sensing Symp. (IGARSS)*, Beijing, China, pp. 1146–1149, 2016.

[11] S. Acharya, K. Pant and K. Gyawali, "Deep learning based large scale handwritten devanagari character recognition, in *9th Int. Conf. on Software, Knowledge, Information Management and Applications*, Kathmandu, Nepal, no. 9, pp. 36–44, 2016.

[12] N. H. Habib, A. Adnan and S. Basar, "An analysis of off-line and on-line approaches in Urdu character recognition," in *Artificial Intelligence, Knowledge Engineering Data Bases*, Venice, Italy, vol. 3, no. 2, pp. 234–245, 2016.

[13] A. Mezghani, S. Kanoun, M. Khemakhem and H. Abed, "A database for Arabic handwritten text image recognition and writer identification," in *Eighth Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Bari, Italy, pp. 399–402, 2012.

[14] A. Safdar and K. Khan, "Online Urdu handwritten character recognition: Initial half form single stroke characters," in *12th Int. Conf. on Computer Systems and Technologies (FIT)*, Islamabad, Pakistan, pp. 292–297, 2014.

[15] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: Present contributions and future directions," *IEEE Access*, vol. 6, no. 1, pp. 46019–46046, 2018.

[16] T. Zhang, J. Liang and B. Ding, "Acoustic scene classification using deep CNN with fine-resolution feature," *Expert System Application*, vol. 143, no. 1, pp. 113067, 2020.

[17] S. Xu, L. Liang and C. Ji, "Gesture recognition for human–machine interaction in table tennis video based on deep semantic understanding," *Signal Processing Image Communication*, vol. 81, no. 1, pp. 115688, 2020.

[18] E. Arif, S. K. Shahzad, R. Mustafa, M. A. Jaffar and M. W. Iqbal, "Deep neural networks for gun detection in public surveillance," *Intelligent Automation and Soft Computing (IASC)*, vol. 32, no. 1, pp. 909–922, 2022.

[19] O. Ali and A. Shaout, "Hybrid Arabic handwritten character recognition using PCA and ANFIS," in *Int. Arab Conf. on Information Technology United States*, Beni-Mellal, Morocco, pp. 23–31, 2016.

[20] H. Ali, A. Ullah, T. Iqbal and S. Khattak, "Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network," *SN Applied Sciences*, vol. 2, no. 2, pp. 1–12, 2020.

[21] S. Husain, A. Sajjad and F. Anwar, "Online Urdu character recognition system," in *Proc. of the 15th IAPR Int. Conf. on Machine Vision Applications*, Nagoya, Japan, pp. 98–101, 2017.

[22] S. Habib, M. K. Shukla and R. Kapoor, "Preprocessing for identification of degraded Urdu and Devanagari printed script," *Computational Intelligence in Pattern Recognition, Springer*, vol. 20, no. 1, pp. 519–528, 2020.

[23] N. Tabassum, T. Alyas, M. Hamid, M. Saleem and S. Malik, "Semantic analysis of Urdu English tweets empowered by machine learning." *Intelligent Automation & Soft Computing (IASC)*, vol. 30, no. 1, pp.175–186, 2021.

[24] V. Vehovar, V. Toepoel and S. Steinmetz, "Non-probability sampling," *Sage Handbook of Survey Methods*, no. 3, vol. 45, pp. 329–345, 2016.

[25] R. Laishram, P. Singh, T. Singh, S. Anilkumar and A. Singh, "A neural network based handwritten meitei mayek alphabet optical character recognition system," in *IEEE Int. Conf. on Computational Intelligence and Computing Research*, Coimbatore, India, pp. 1–5, 2014.

[26] M. Awni, M. I. Khalil and H. M. Abbas, "Deep-learning ensemble for offline Arabic handwritten words recognition," in *14th Int. Conf. on Computer Engineering and Systems (ICCES)*, Cario, Egypt, pp. 40–45, 2019.

[27] K. Nongmeikapam, W. Kumar and M. P. Singh, "Exploring an efficient handwritten manipuri meetei-mayek character recognition using gradient feature extractor and cosine distance based multiclass k-nearest neighbor classifier," in *Proc. of the 14th Int. Conf. on Natural Language Processing (ICON)*, Kolkata, India, pp. 328–337, 2017.

[28] C. Kumar and S. K. Kalita, "Recognition of handwritten numerals of manipuri script," *International Journal of Computer Application*, vol. 84, no. 17, pp. 1–5, 2013.

[29] N. Tabassum, A. Yaseen, M. Israr, N. Anwar, M. W. Iqbal *et al.,* "Data classification using decision trees J48 algorithm for text mining of business data," *Lahore Garrison University Research Journal of Computer Science and Information Technology*, vol. 5, no. 2, pp. 73–81, 2021.

[30] Y. Wen and L. He, "A classifier for bangla handwritten numeral recognition," *Expert System Application*, vol. 39, no. 1, pp. 948–953, 2012.

[31] A. Ashiquzzaman and K. Tushar, "Handwritten Arabic numeral recognition using deep learning neural networks," in *IEEE Int. Conf. on Imaging, Vision & Pattern Recognition (icIVPR)*, Dhaka, Bangladesh, pp. 1–4, 2017.

[32] U. Rashid, M. W. Iqbal, M. A. Skiandar, M. Q. Raiz, M. R. Naqvi *et al.,* "Emotion detection of contextual text using deep learning," in *4th Int. Symp. on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Istanbul, Turkey, pp. 1–5, 2020.

[33] N. Das, R. Sarkar, S. Basu and D. Basu, "A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application," *Application Soft Computing*, vol. 12, no. 5, pp. 1592–1606, 2012.

[34] M. Rahman, M. Akhand, S. Islam and M. Rahman, "Bangla handwritten character recognition using convolutional neural network," *International Journal Image, Graph. Signal Processing*, vol. 7, no. 8, pp. 42–49, 2015.

[35] S. Inunganbi, P. Choudhary and K. Singh, "Local texture descriptors and projection histogram based handwritten meitei mayek character recognition," *Multimedia Tools Applications*, vol. 79, no. 3, pp. 2813–2836, 2020.