

# Multi-Domain Deep Convolutional Neural Network for Ancient Urdu Text Recognition System

K. O. Mohammed Aarif<sup>1,\*</sup> and P. Sivakumar<sup>2</sup>

<sup>1</sup>Department of Electronics and Communication Engineering, C. Abdul Hakeem College of Engineering and Technology, Melvisharam, 632509, India

<sup>2</sup>Department of Electronics and Communication Engineering, Dr. NGP Institute of Technology, Coimbatore, 614048, India

\*Corresponding Author: K. O. Mohammed Aarif. Email: aarifko@gmail.com

Received: 19 August 2021; Accepted: 23 September 2021

**Abstract:** Deep learning has achieved magnificent success in the field of pattern recognition. In recent years Urdu character recognition system has significantly benefited from the effectiveness of the deep convolutional neural network. Majority of the research on Urdu text recognition are concentrated on formal handwritten and printed Urdu text document. In this paper, we experimented the Challenging issue of text recognition in Urdu ancient literature documents. Due to its cursiveness, complex word formation (ligatures), and context-sensitivity, and inadequate benchmark dataset, recognition of Urdu text from the literature document is very difficult to process compared to the formal Urdu text document. In this work, first, we generated a dataset by extracting the recurrent ligatures from an ancient Urdu fatawa book. Secondly, we categorized and augment the ligatures to generate batches of augmented images that improvise the training efficiency and classification accuracy. Finally, we proposed a multi-domain deep Convolutional Neural Network which integrates a spatial domain and a frequency domain CNN to learn the modular relations between features originating from the two different domain networks to train and improvise the classification accuracy. The experimental results show that the proposed network with the augmented dataset achieves an averaged accuracy of 97.8% which outperforms the other CNN models in this class. The experimental results also show that for the recognition of ancient Urdu literature, well-known benchmark datasets are not appropriate which is also verified with our prepared dataset.

**Keywords:** Text recognition; deep learning; multi-domain CNN; ligatures; pattern recognition

## 1 Introduction

Text recognition is regarded as one of the great advances in the field of natural language processing and has attracted many researchers to work in the discipline for translation of handwritten or printed documents to a computer editable format. Text recognition is also one of the biggest challenges in machine translation because it requires identifying the specific sequence of words that are the source of the sentence. Urdu is



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

one of the primeval and primary languages of the Indian sub-continent. Around the world, Urdu is spoken in more than 15 countries with an estimated over 280 million speakers. Urdu is the national language of Pakistan, and an official language of six states of India, and also one of the 22 official languages recognized in the Constitution of India. The Urdu language has the characteristics of Arabic, Farsi, and Pashto, so an Urdu speaker can without much of a stretch comprehend Hindi. Urdu scriptwriting is from right to left using alphabets that are derived from the Arabic alphabets character set. While speaking, Urdu is quite similar to Hindi but different in writing. Urdu is written in Arabic script with some additional characters, in different font styles of cursive script, such as Nasta'liq, Kofi, Thuluth, Diwani, Riq'a, and Naskh. From the last two decades, significant research has been conducted in languages like English, Chinese, French, Devanagari, Tamil, etc., to digitize existing printed and handwritten documents [1]. But the most challenging assignment is to develop a recognition system for cursive languages, like Arabic, Farsi, Pashtu, and Urdu. Deep learning has made a huge impact on computer vision and set the state-of-the-art in providing extremely definite classification results. With the advancement of deep learning, character recognition has become a widely used method for identifying and labeling natural languages. The most recent generation of deep learning is the convolutional neural network (CNN), which has shown to be very effective at detecting complex expressions in natural language words. CNN has become increasingly popular due to its ability to capture highly descriptive and well-ordered image features [1]. One of the prime factors of the supreme accuracy attainment of CNN is that it is trained with massive categorized samples. ImageNet is the most broadly perceived dataset used to train CNN models utilizing error prorogation [2]. But earning a dataset as extensively characterized as ImageNet remains a big challenge in the character recognition of under resource languages like Urdu. Training intense CNN utilizing a small size dataset frequently causes convergence issues.

Transfer learning is one of the most promising approaches for training deep CNN models with limited training data. Transfer learning makes the network learn quickly on new input data with fewer classes [3]. For under-resourced classification tasks like text recognition, transfer learning is the best option instead of reducing the network limit [4]. Various script identification systems have been effectively demonstrated using different network models recently [5], like Reference [6] has presented a handwriting recognition of historical documents by transfer learning CNN-BLSTM-CTC network, in which the authors have transfer learned a composite dataset with ground-truth and pooled the collective attributes with a new dataset. Similarly, Reference [7] exhibited a hybrid model for handwriting recognition using transfer learning a heterogeneous dataset. A character-level text ConvNets is presented in [8] for English and Chinese corpus by transfer learning to endeavor and reuse the important portrayals that are found out in the ConvNets from an enormous scope dataset. AlexNet the most robust CNN which is pre-trained with the ImageNet dataset has been exploited for character recognition of many scripts like Devanagari [9], Malayalam [10], Korean [11], and Tamil [12]. Manually written Devanagari character acknowledgment has been introduced in [13] using layer-wise preparation of Deep CNN and accomplished great results using six different adaptive gradient methods. A Latin and Chinese character acknowledgment has been presented in [14] where the author has transfer-learn a deep CNN on digits to recognize upper case letters. Transfer learning has successfully supported numeral recognition of several scripts where the network is trained with a small set of the numeral dataset [15]. One of the oldest languages of the world—Tamil, handwritten characters have been transfer learned using the Vgg16 and achieved promising results [16]. One of the cutting edge techniques for similar script family languages-Kanada and Telugu, character recognition by transfer learning has been exhibited in [17]. From the literature, we noticed that mostly the transfer learned networks are effectively smeared on the printed script rather than handwritten, and we also observe that CNN-based transfer learning is effectively applied to the non-cursive script like Chinese, Latin, Bangala, and Devanagari, etc. Relatively very few research have been concentrated on cursive scripts like Arabic [18], Urdu [19], and Farsi [20], where the experiment carried out on regular

documents, and the network models are trained using specific benchmark datasets like UNHD [21], UPTI [22], EMILLE [23] and WordNet [24]. These datasets consist of handwritten text lines and word images written by various writers. For Urdu literature document digitization these datasets and models cannot be applied due to the canonical complexity of Urdu literature documents like no proper baseline, diagonality, packed loops, incorrect loop, and hefty disparity of ligatures. To our knowledge currently, there is no such Urdu ligature image dataset available solely for recognition of Urdu text from literature documents. The key advantage of Urdu literature is that comes in multiple volumes, and nearly 80% of the text (words) in the first book are recurrent in the succeeding volumes. We took this advantage to develop a dataset, especially for Urdu literature document text recognition. We extracted the ligatures from the first volume of an Urdu ancient fatawa book (Fatawa Aziz) which has 4 volumes. Then we proposed a multi-domain deep CNN for handwritten Urdu ligature classification which exploits both spatial domain and frequency domain convolution to learn modular relating features from Urdu ligatures image.

The chief contributions of this work are (I). Developing a ligature-based image dataset for Urdu literature document text recognition, (II). Exploiting multi-domain CNN features for Urdu text recognition (III). Deep learning in Urdu ligatures classification, eliminates the need for segmenting individual characters. (IV). For ancient literature archive digitization, our proposed network is highly suitable for implementing real-time recognition.

## 2 Dataset Preparation

### 2.1 Characteristic of Urdu Literature Document Text

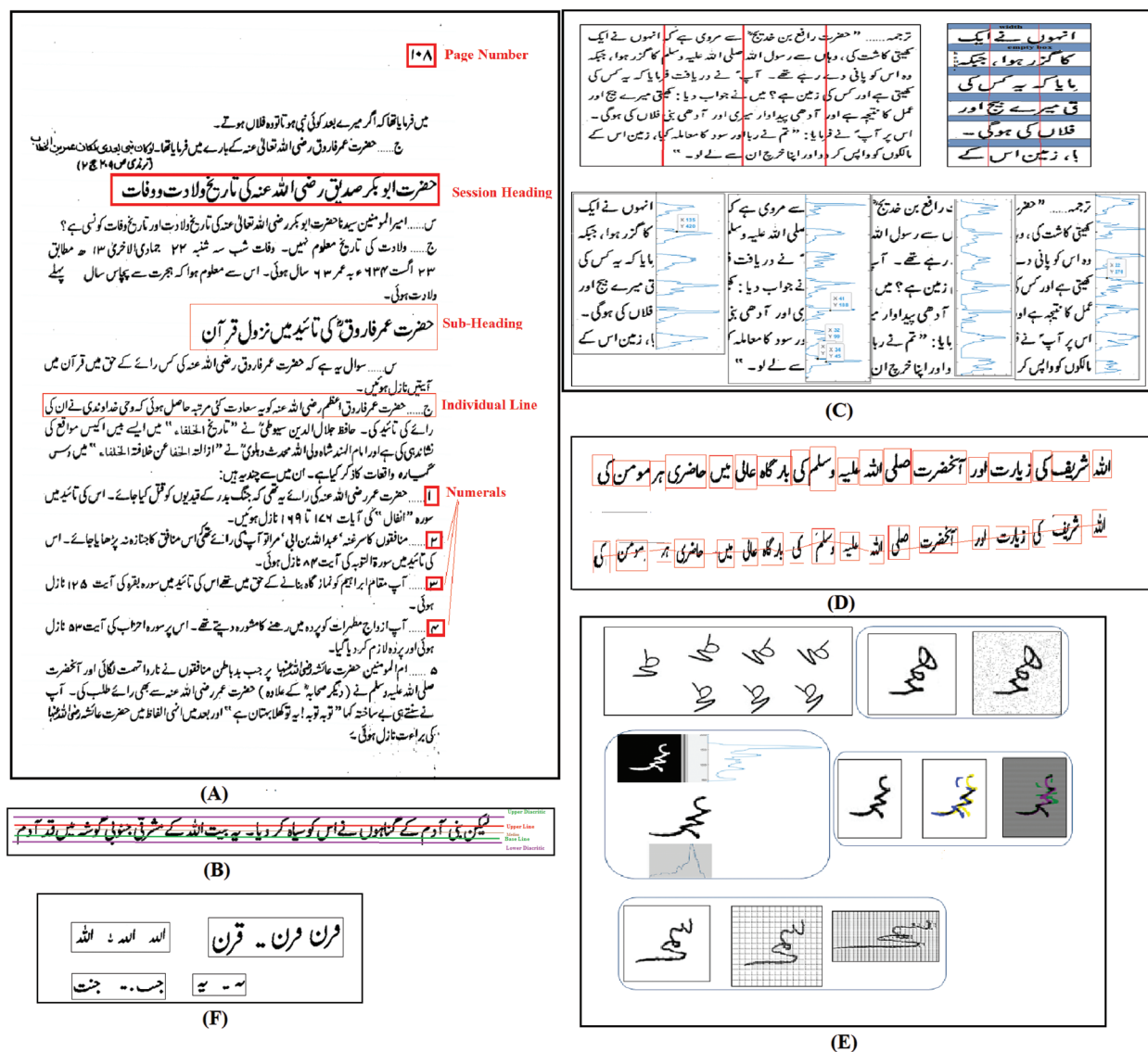
Nastalique is an artistic, curvaceous, and calligraphic style widely used in Urdu literature script writing. National Language Authority of Pakistan has defined 58 characters in Urdu, 28 of which are derived from Arabic, but just 40 essential characters and one dochashmi-hey is utilized to frame every composite letter set. Urdu Nastalique script is inherently cursive, has four unique shapes for each character, and also context-sensitive. A ligature is molded by fusing two or more characters cursively with diacritics in a free stream structure. A ligature is not a complete word, rather in the greater part of the cases a part of a word, or a subword. A Nastalique word is made out of ligatures and secluded characters.

### 2.2 Segmentation

We prepared the dataset by extracting all possible ligatures from the first volume of the Urdu fatwa book Fatawa Azizi authored by Shaykh Shah Abdul Aziz Muhaddith Dehlvi. r.a—consist of 4 Volumes. The scanned copy of this book was obtained from the Arabic college of our nearby town. The scanned documents had two common & conventional issues—noise and skew. As preprocessing, we performed document de-noising using median filtering scheme & k-fill method [25] and de-skewing using Ali's Algorithm [26]. To have a smart thought of the physical structure of the Urdu literature document, Fig. 1A shows a sample text page from the fatawa book.

Segmenting lines from a page or a paragraph in an Urdu literature document is a challenging and crucial task due to the following issues:

- (i) Overlapping line: The previous line may extend beyond their lower diacritic line and reaches the next line upper diacritic section.
- (ii) Connected components: Mostly the first and the last character in a word or ligature may extend to get connected with characters in the adjacent line.
- (iii) Merged diacritics: Multiple diacritics like (dots) may get merged as a small line.
- (iv) Skewed lines: line may be skewed and curved due to improper scanning.
- (v) Touching words within a line.



**Figure 1:** (A) Urdu fatawa document physical structure mentioning the different types of content. (B) Fictitious lines to perceive the Urdu literature text line. (C) A paragraph cropped with a straight down strip of width  $w$ , horizontal projection profile of each strip, identification of empty lines, non-empty lines, and the height of the text line in a strip. (D) Ligatures segmentation using connected components and DFS. (E) Ligature augmentation resultant images. (F) Separated primary, secondary, and complete ligature clusters

Several methods exist to take care of these problems and they can be categorized as one of two classifications, top-down and bottom-up approaches [27]. In a top-down methodology, a line division calculation utilizes enormous features of a line to decide its limits. The Bottom-up method begins from the smallest component of an archive picture i.e., the pixel. By gathering contacting pixels, associated segments are produced. Based on our document suitability we employed the block covering analysis method for text line segmentation as it is proven to be the best in the literature.

The ligature segmentation phase involves the separation of text lines into individual words/ligatures. Ligature segmentation in the case of Urdu literature documents relies basically upon the separation among words and sub-words. Due to non-consistently slanted sub-words in the literature script, the space among words is not considered. Literature writing is precisely compacted and free spaces are fully occupied. In the case of the Urdu literature text, there are no broad suppositions to be made. Still, for examination, we take 2 fundamental suppositions that maintain a significant degree for our chosen books. First, though the writing is cursive, it is legible and has an inclination under 30 degrees to the vertical. Second, the lines in the record are not even, yet at the same time, two back-to-back lines have a sufficient gap to isolate them, so we employed the depth-first search-based connected components method for our ligature extraction. Preparing a classifier to perceive requires labeled ligature classes. Grouping ligatures manually and labeling them as per the text or ligature is an exorbitant procedure as far as the time and exertion. Hence we complete a self-loader grouping of extricated ligatures. Miss-grouping in clustered classes is then revised through visual investigation in request to mark groups error-free to fill in as preparing information. We utilized the dynamic time wrapping sequential algorithm (DTW) [28] for our ligature clustering. The major advantage of this method is that it does not require the number of ligature classes initially. Fig. 1 shows the sequence of ligature extraction procedures from a document page. Tab. 1 list the total number of ligatures of various size extracted and Tab. 2 shows the number of High-Frequency Ligatures (HFL), Low-Frequency Ligatures (LFL) and Unique Ligatures (UL).

---

**Algorithm 1:** Algorithm for Ligature extraction

---

**Input:** Urdu Literature Document Image

**Output:** Individual ligatures

Step-1: Page to Line Segmentation:

(i) Line Segmentation: Block Covering Analysis-the paragraph image is cropped recurrently into straight down stripes of width 'w'.

(ii) 'd' = Dimension of horizontal projection profile of individual strips  
 Empty Line = Min.Histo Valu ; Non-Empty line = Max.Histo Valu;

Text Line = covering block dimensions are demarcated by measuring the intervals of nonempty lines

Step-2: Line to Ligature Segmentation:

(i) Ligature Segmentation: Depth First Search (DFS)-Here the ligature is taken as connected components

(ii) Calculate connectivity strength function 'C' to align components from the text line

(iii) Individual Ligature = Dense structure of given text image.

Step-3: Ligature Clustering:

Ligature Clustering: dynamic time wrapping method. A ligature is arbitrarily picked and is accepted as a means of the primary group, every one of the rest is then chosen individually by a mean distance of each group and processed using DTW

Step-4:

If (Separating distance = closest group is under a predefined edge), the current ligature is added to the separate bunch and the group means is updated.

Else (another group is made with the current ligature as its center) Text Line = covering block dimensions are demarcated by measuring the intervals of nonempty lines.

---

**Table 1:** Ligatures count of various sizes extracted from the fatawa books

| Number of characters per ligature | Count |
|-----------------------------------|-------|
| Individual characters             | 248   |
| 2-Character ligatures             | 1718  |
| 3-Character ligatures             | 921   |
| 4-Character ligatures             | 756   |
| 5-Character ligatures             | 541   |
| 6-Character ligatures             | 105   |
| 7-Character ligatures             | 22    |

**Table 2:** Our complete dataset details

| Dataset               | HFL  | LFL | UL  | Total ligature |
|-----------------------|------|-----|-----|----------------|
| Fatawa-Azizi volume 1 | 3283 | 731 | 297 | 4311           |

### 3 Image Augmentation

There are numerous approaches to address difficulties related to constrained resources in deep learning. Image augmentation is one valuable procedure in building convolutional neural systems that can expand the size of the preparation set without obtaining new images. The thought is straightforward to replicate pictures with verities so the model can gain from more samples. By augmentation, we can increase the picture in such a way that it protects the key highlights, yet revises the pixels enough that it includes some agitation. Text Image Augmentations come from simple basic changes, for example, even flipping, shading space enlargements, and irregular editing. etc. To emulate the data variant perceived in the Urdu ligatures we furnish the following five augmentation techniques: Geometric transformation, Addition of uniform noise, Image glitch, Distortion using a grid, and horizontal and vertical profile normalization. Our image augmentation sequence is described in Algorithm 2 and Fig. 1E shows the results of proposed augmentation methods for the Urdu text document.

---

#### Algorithm 2: Image Augmentation

---

Step 1. Geometric transformation:

Calculate the control points based on the spatial angle of the ligature.

Step 2. Pixels in the ligature image are spatially transformed by physical rearrangement.

Step 3. Assigns grey levels to the transformed image.

Step 4. Adding Uniform Noise:

Noise tests pseudo-arbitrarily from the standard ordinary circulation, with a customizable change and a mean of 0.

Step 5. The background noise is free and indistinguishably conveyed. A ligature that is enlarged with white Gaussian noise with a variance of 0.06 & 0.08.

Step 6. Ligature profile normalization:

Normalize the ligature Image to recompense the variations in the size using the distance between the upper BL & UDL as well as the lower BL and the LDL.

---

(Continued)



**Algorithm 2: (continued)**

Step 7. Grid-based image distortion:

Calculate the control point placement interval and standard deviation by which it randomly displaces the control points.

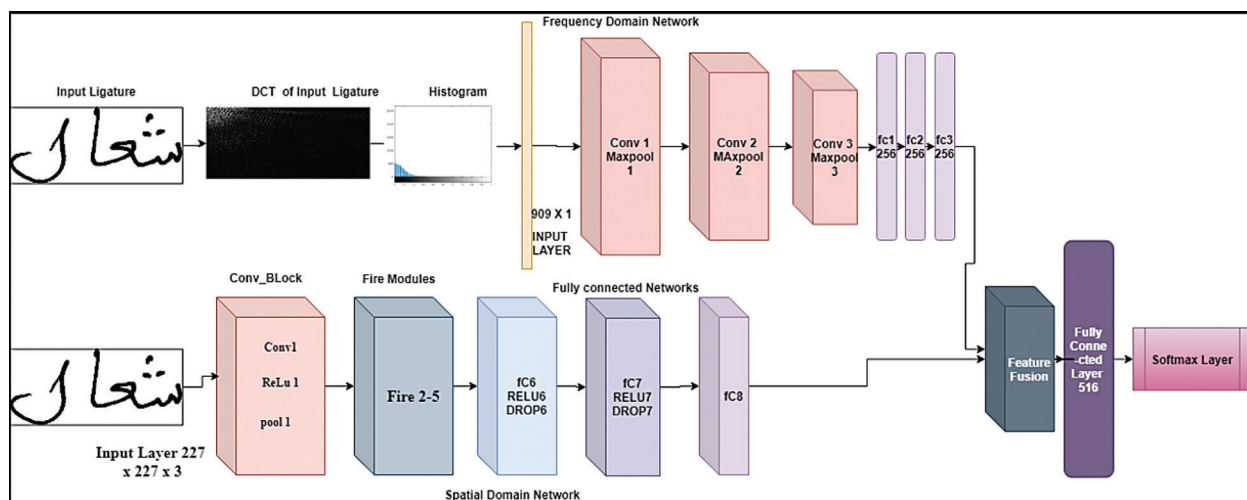
Step 8. Mapping is first characterized between any two proportionate normal frameworks with minor geometric confinements and is then ideally stretched out to the inside of the source matrix.

## 4 Proposed Network Architecture

In this work, we proposed a multi-domain Convolutional Neural Network for Urdu ligature recognition. Our network integrates a spatial domain CNN and a frequency domain CNN. The individual architecture of spatial and frequency domain CNN is discussed below.

### 4.1 Proposed Multi-Domain Network

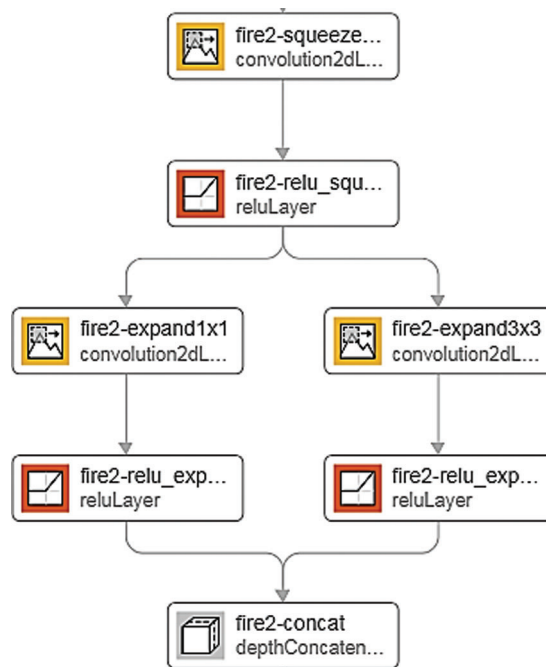
Traditional CNN models have huge parameter space because they are much denser and therefore require more time to parse. In our proposed network, the spatial domain network is constructed using the fire modules which makes the network lightweight and fast that can be easily deployed in the edge devices. The frequency and spatial domain networks are combined up to their individual fully connected layers. The frequency-domain network gets the input as a histogram of the DCT coefficients of the ligature image and the spatial network gets the input as a raw ligature image. The features originating from the last fully connected layers of both the networks are fused to learn the inter-modal relation of ligature images from two different domain networks. The fused robust feature vector is fed to the last fully connected layer which has 512 neurons followed by a softmax layer which yields the probability that individual text is classified into respective classes. Fig. 2 portrays our proposed multi-domain CNN. Algorithm 2 describes the complete process of our proposed system.



**Figure 2:** Proposed multi-domain network layer structure

### 4.2 Spatial Domain CNN

Our spatial domain network consists of a convolutional block (Conv-1, Relu-1 & max pool-1) and four fire modules similar to SqueezeNet. The network replaces some portion of  $3 \times 3$  convolution core in CNN with  $1 \times 1$  convolution parts, breaks down the first one convolution layer into two layers, and condenses them into a Fire module. Our fire module as shown in Fig. 3 has a Congestion layer ( $1 \times 1$  filter to decrease the input channel from  $3 \times 3$ ) which reduces the size of the feature map and an Expand layer (a combination of  $1 \times 1$  and  $3 \times 3$  filters to reduce filter size) which increases the gain. The input layer of the network is  $227 \times 227 \times 3$ , the initial convolutional comes with a RELU activation followed by 4 Fire modules (Fire2 to Fire5), finally tailed with three fully connected layers and a softmax layer.



**Figure 3:** Fire module of our spatial domain proposed network

### 4.3 Frequency Domain CNN

Our frequency-domain CNN has 3 convolutional blocks (Conv and Maxpool), and all the max pool layers have a kernel of the same size. The three fully connected layers have 256 neurons each. In the frequency domain network before feeding the input layer, we first compute the histogram of DCT coefficients from the input ligature image. From an image block of  $8 \times 8$ , DCT coefficients are extracted for each spatial frequency  $(i, j)$ , and the histogram  $h(i, j)$  is built on behalf of the events absolute quantized DCT values. The input layer takes a vector of 784 elements as input for each ligature image. This feature vector is used to train our spatial domain CNN.

After we characterize the layers of our network the following stage is to set up the training options for the organization. We utilize the trainingOptions capacity to characterize the global training parameters.

MiniBatchSize

A small cluster is a subset of the training set that is utilized to assess the slope of the loss function and update the weights, the measure of information remembered for each sub-age weight change is known as the batch size. While the utilization of huge smaller than expected batches expands the accessible computational



parallelism, little group training has been shown to give further developed speculation execution so we have chosen an intermediate batch size of 06.

Maxepochs:

An epoch is the full pass of the training algorithm over the whole training set. Datasets are generally gathered into groups (particularly when the measure of information is exceptionally huge). The overall relation where dataset size is  $S$ , number of epochs is  $O$ , number of iterations is  $I$ , and cluster size is  $C$  would be  $S \cdot O = i \cdot C$ .

InitialLearnRate

The volume that the weights are updated during training is alluded to as the “learning rate.” Specifically, the learning rate is a configurable hyper-parameter utilized in the training of neural networks that has a small positive value, regularly in the reach somewhere in the range of 0.0 and 1.0. Learning rate is utilized to scale the extent of parameter updates. The decision of the incentive for learning rate can affect two things: 1) how quick the calculation learns and 2) whether the cost function is minimized or not. The default esteem is 0.01 for the ‘sgdm’ solver and 0.001 for the ‘rmsprop’ and ‘adam’ solvers.

Shuffle

Shuffle the training information before each training epoch, and Shuffle the approval information before each network validation. On the off chance that the smaller than expected batch size doesn’t equitably isolate the number of training tests, then, at that point train-Network disposes of the training information that doesn’t squeeze into the last total scaled-down group of every epoch. To try not to dispose of similar information each epoch we set the ‘Shuffle’ value to ‘every epoch’.

Verbose: Indicator to display training progress information

---

### Algorithm 3: Proposed Multi-Domain Algorithm

---

**Input: Ligature Image**

**Output: Recognized Ligature**

Step-1: Load the image datastore and ‘splitEachLabel’ with the required partition

(i) Input to Frequency Domain Network: Histogram of DCT coefficients of  $8 \times 8$  block, spatial frequency ( $i, j$ ), and the histogram  $h(i, j)$ .

(ii) Input to Spatial Domain Network: Resize the ligature image to  $227 \times 227 \times 3$ .

Step-2: Load the proposed network into ‘NET’.

Step-3: Find the layers to transfer learn by ‘layer transfer.

fullyConnectedLayer; softmax layer; classification layer

Step-4:

Freeze the initial layer weights and bias and change the same for the new layers

WeightLearnRateFactor, 50, BiasLearnRateFactor, 50

Step-5:

Training Options: ‘MiniBatchSize’, 10, ‘MaxEpochs’, 02, ‘InitialLearnRate’,  $3e-4$ , ‘Shuffle’, ‘every-epoch’, ‘Verbose’, false.

Step-6:

Initialize training the NET with the provided dataset and training options

---

(Continued)

**Algorithm 3: (continued)**

Step-7:

The output of L7 Fc(spatial-4096) and the output of L6(frequency-256) are obtained and fed to the last fully connected layer which has 512 neurons.

Step-8:

Finally, the softmax layer classifies the ligatures in 4311 classes.

Accuracy = mean(YPred == YValidation).

## 5 Experimental Analysis

Our proposed network is trained using our generated dataset of Urdu literature documents. We have used A NVIDIA-1060 graphics Zotak (CUDA v10.0) with an Intel Core i7-operating at 3.60 GHz for faster response. For testing, we interface a smartphone to the system using the IP-Webcam app. IP Webcam app transforms the cell phone into a network camera with different review choices and transfers video over Wi-Fi without the internet. Our system reads the image through the smartphone IP gateway and resizes it appropriately to feed this to the input layers of our network for classification. We performed 4 rounds of tests, in TEST-1 we self-evaluate the generated ligatures in different ratios. In TEST-2, we performed testing & validating of ligatures on the remaining three-volume of the same book. In TEST-3 (Cross Validation) our trained network validated the ligatures of the Urdu ancient literature books which were not used in training. In TEST-4, we tested our trained network on formal Urdu handwritten documents and other benchmarks Urdu datasets.

TEST-1: The first test involves training our network with the prepared dataset and self-validating it. The training comprises learning above 4000 ligature classes with each around 150 images. [Tab. 3](#) shows our proposed method achieved an averaged recognition accuracy of 97.8% with the augmented dataset and 92% with the raw dataset. It proves that our proposed network and the augmentation techniques are best suited for Urdu ligature classification.

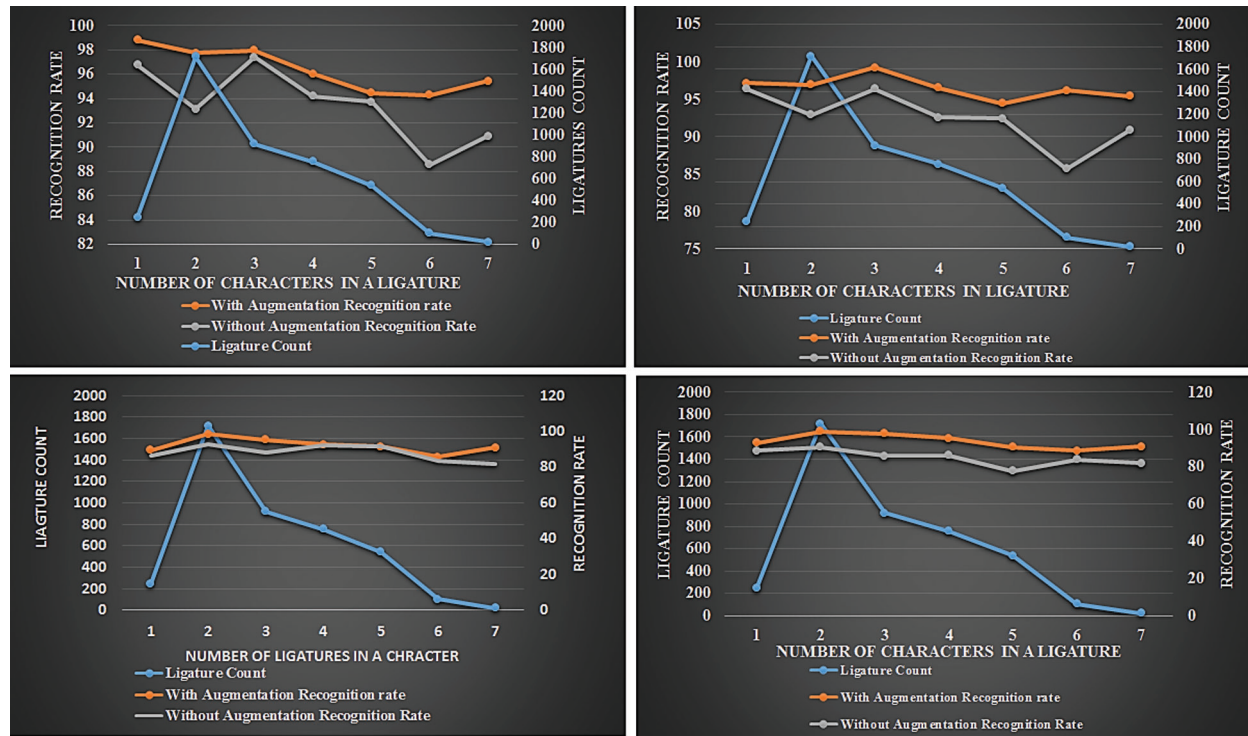
**Table 3:** Performance of self-validation in the raw and augmented dataset

| Ligatures               | Raw dataset |       |       | Augmented dataset |       |       |
|-------------------------|-------------|-------|-------|-------------------|-------|-------|
| Ratio                   | 7:3         | 8:2   | 9:1   | 7:3               | 8:2   | 9:1   |
| Validation-accuracy (%) | 92.94       | 93.20 | 93.87 | 96.16             | 96.94 | 97.61 |
| HFL-accuracy (%)        | 91.55       | 93.71 | 93.43 | 97.87             | 98.01 | 98.11 |
| LFL-accuracy (%)        | 89.64       | 90.16 | 90.27 | 93.33             | 94.09 | 94.17 |
| UL-accuracy (%)         | 88.72       | 88.87 | 90.11 | 92.10             | 92.91 | 93.75 |

In TEST-2 we evaluate our network by testing & validating the ligatures of the remaining volume of the same book. [Tab. 4](#) shows the result of averaged recognition accuracy associated with HFL, LFL, and UL. It is observed from [Tab. 4](#) that ligatures of volume 1 are recurrently available in the succeeding volumes. Nearly 95% of the ligatures are repeated in the remaining volume of the same literature book. [Fig. 4](#) shows the accuracy attainment of ligatures (based on the number of characters) in all four volumes with the raw and augmented dataset. It is witnessed that the minimum size ligatures (1 & 2 Characters) and maximum size ligatures (7-characters) are having good accuracy compared to other ligatures.

**Table 4:** The recognition accuracy of the first volume ligatures with the remaining volumes

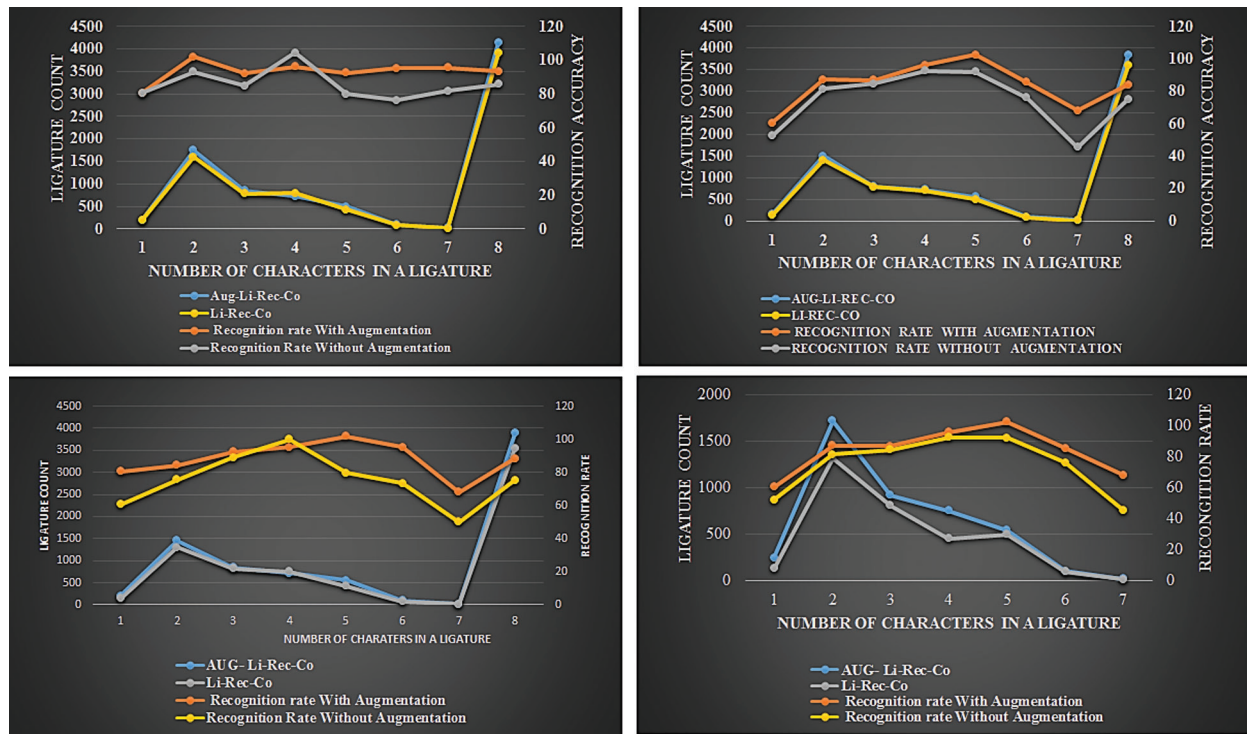
| Ligatures               | Volume 1 | Volume 2 | Volume 3 | Volume 4 |
|-------------------------|----------|----------|----------|----------|
| Validation-accuracy (%) | 97.61    | 96.44    | 97.07    | 96.54    |
| HFL-accuracy (%)        | 98.11    | 96.98    | 97.74    | 96.61    |
| LFL-accuracy (%)        | 94.17    | 93.81    | 94.41    | 93.31    |
| UL-accuracy (%)         | 93.75    | 93.71    | 94.70    | 93.50    |

**Figure 4:** Analysis of ligature recognition based on the number of characters with and without augmentation

In TEST-3, we cross-validate our trained network by experimenting it on the other Urdu ancient literature books (Umdat-ul-Fiqh-4-Volumes). This literature book also has 4 volumes with a nearly equal number of pages. Tab. 5 Shows the result of averaged accuracy associated with HFL, LFL, and UL. It is observed from Tab. 5 that ligatures of our original dataset match nearly 93% to the ligatures of the new book in all the volumes. Fig. 5 shows the accuracy attainment of ligatures (based on the number of characters) in all four volumes with the raw and augmented dataset.

**Table 5:** Recognition accuracy in the book *Umdat-ul-Fiqh-4-Volumes*

| Ligatures               | Volume 1 | Volume 2 | Volume 3 | Volume 4 |
|-------------------------|----------|----------|----------|----------|
| Validation-accuracy (%) | 93.17    | 93.87    | 92.96    | 93.19    |
| HFL-accuracy (%)        | 94.87    | 94.10    | 92.78    | 92.10    |
| LFL-accuracy (%)        | 90.31    | 91.87    | 89.72    | 89.56    |
| UL-accuracy (%)         | 92.54    | 93.55    | 95.70    | 94.44    |



**Figure 5:** Analysis of ligature recognition based on the number of characters with and without augmentation (Umdat-ul-Fiqh)

In TEST-4, we tested our trained network on formal Urdu handwritten documents obtained from 50 student's exam answer papers from local Arabic colleges. We also evaluated our trained network on the benchmark Urdu datasets like UNHD, UPTI, EMILLE, and WordNet. [Tab. 6](#) reveals that the ligatures of formal documents are nowhere compatible for experimenting on ancient Urdu literature documents. The training time in a DNN depends primarily on the network structure of the problem domain, computational resources, and model parameters. However, the trained models tend to improve significantly on the training error in the problem. As our proposed network has two independent different domain networks, the training of our selected datasets took quite a long time, but the recognition time of our system is extraordinarily fast compared to the other network in this domain as shown in [Tab. 7](#). [Tab. 8](#) shows that our proposed method performs comparably extremely well with a consistent averaged accuracy of 97.8% against the state-of-the-art in Urdu ligature classification.

**Table 6:** Performance of our trained model on formal documents and Urdu benchmark datasets

| Ligatures               | Handwritten formal documents | UNHD  | UPTI  | EMILLE | WordNet |
|-------------------------|------------------------------|-------|-------|--------|---------|
| Validation-accuracy (%) | 57.23                        | 42.74 | 31.89 | 30.51  | 28.34   |
| HFL-accuracy (%)        | 30.84                        | 38.21 | 27.84 | 22.54  | 27.33   |
| LFL-accuracy (%)        | 29.74                        | 32.74 | 24.23 | 36.98  | 30.12   |
| UL-accuracy (%)         | 30.87                        | 30.25 | 30.87 | 48.32  | 20.19   |

**Table 7:** Training and recognition time analysis with and without GPU

| Dataset              | Training time | Recognition time per ligature | System configuration   |
|----------------------|---------------|-------------------------------|--|
| Fatawa Azizi         | 960 min       | 0.7 s                         | Intel Core i7-operating at 3.60 GHz with A NVIDIA-1060 graphics Zotak (CUDA v10.0) |
| <i>Umdat-ul-Fiqh</i> | 873 min       | 0.7 s                         |  |
| Fatawa Azizi         | 1260 min      | 0.9 s                         | Intel Core i7-operating at 3.60 GHz System configuration                           |
| <i>Umdat-ul-Fiqh</i> | 1117 min      | 0.9 s                         |  |

**Table 8:** Performance comparison of proposed work against the state-of-the-art in ligature classification

| Reference        | Dataset                                | Methodology                      | Accuracy (%) |
|------------------|--|----------------------------------|--------------|
| [29]             | Customized synthetic Printed ligatures | AlexNet and VGG16                | 97           |
| [30]             | Customized printed ligatures           | Multi-level clustering           | 90(K-NN)     |
| [31]             | UPTI                                   | Stacked denoising autoencoder    | 96           |
| [32]             | Noori Nastaleeq ligatures              | Recognizers with maturation pass | 96           |
| [33]             | UPTI                                   | BLSTM                            | 96.71        |
| [34]             | CLE                                    | Neural network                   | 95.20        |
| [35]             | Customized ligatures                   | K-Nearest Neighbor, SVM          | 90.29        |
| [36]             | CLE                                    | Hidden Markov model              | 92.00        |
| Proposed network | UPTI, CLE, and customized ligatures    | Multi-domain CNN                 | 97.8         |

From the above correlations and significant evidence, we conclude that our proposed network with the information enlargement strategies is best appropriate for the recognition of ancient Urdu literature documents. It is also demonstrated that for the recognition of ancient Urdu literature, well-known available datasets are not appropriate which is verified with our prepared dataset.

## 6 Conclusion

In this paper, we experimented the challenging issue of text recognition of ancient Urdu literature documents. One of the key advantages of ancient Urdu literature is that it comes in multiple volumes and more than 80% of the text (words) in the first book are recurrent in the succeeding volumes. So we develop a dataset from the first volume of an Urdu literature book and trained to recognize the remaining volumes. To experiment this we proposed a multi-domain deep CNN which integrates a spatial domain CNN and a frequency domain CNN. Our network is lightweight and fast to learn the modular relations between features originating from the spatial domain and the frequency domain network. The experimental results show that for the recognition of ancient Urdu literature, well-known benchmark datasets are not appropriate which is verified with our prepared dataset and our proposed network is best appropriate for recognition of ancient Urdu literature documents. The results indicate that our approach can be effective at building a language model that could learn Urdu ligature more simply and compactly, we also conclude that the utilization of multi-domain network features which are robust and more



significant enables higher accuracy gain in the classification of Urdu ligatures. The integration of multi-domain hierarchical features and multi-domain handcrafted features to further improve the classification performance is our future consideration. The proposed system mainly goes in the direction of developing an ideal character recognition for the urdu script.

**Acknowledgement:** We are very much thankful to Mufti Riyaz Ahmed, Principal Miftha-ul-Uloom Arabic College Melvisharam, for providing us the ancient literature. We also thank Saad bin Ahmed, Valerie Mapelli/ELDA, Dr. Faisal shafait, and CLE-Lahore for providing us the UNHD, EMILLE/CIL, and UPTI 2.0 and WordNet1.0 datasets respectively.

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

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

## References

- [1] S. Naz, A. I. Umar, R. Ahmed, S. B. Ahmed, S. H. Shirazi *et al.*, “Offline cursive urdu-nastaliq script recognition using multidimensional recurrent neural networks,” *Neurocomputing*, vol. 177, pp. 228–241, 2016.
- [2] K. Simonyan and A. Zisserman, “A very deep convolutional networks for large-scale image recognition,” in *Int. Conf. on Learning Representations, (ICLR)*, San Diego, CA, 2015. <https://dblp.org/rec/journals/corr/SimonyanZ14a.bib>.
- [3] M. A. Kilvisharam and S. Poruran, “OCR-Nets: Variants of pre-trained CNN for urdu handwritten character recognition via transfer learning,” *Procedia Computer Science*, vol. 171, pp. 2294–2301, 2020.
- [4] S. Gutstein, O. Fuentes and E. Freudenthal, “Knowledge transfer in deep convolutional neural nets,” *International Journal on Artificial Intelligence Tools*, vol. 17, no. 3, pp. 555, 2008.
- [5] Y. Zhou, J. Liu, Y. Zie and Y. K. Wang, “Morphological feature aware multi-CNN model for multilingual text recognition,” *Intelligent Automation & Soft Computing*, vol. 30, no. 2, pp. 715–733, 2021.
- [6] A. Granet, E. Morin, H. Mouchère, S. Quiniou and C. V. Gaudin, “Transfer learning for handwriting recognition on historical documents,” in *Int. Conf. on Pattern Recognition Applications and Methods (ICPRAM)*, Madeira, Portugal, 2018.
- [7] F. Can and A. Yilmaz, “Hybrid handwriting character recognition with transfer deep learning,” in *27th Int. Conf on Signal Processing and Communications Applications Conf. (SIU)*, Turkey, pp. 1–4, 2019.
- [8] M. Sato, R. Orihara, Y. Sei, Y. Tahara and A. Osuhaga, “Text classification and transfer learning based on character-level deep convolutional neural networks,” in *Agents and Artificial Intelligence, Lecture Notes in Computer Science*, vol. 10839. Cham: Springer, 2018.
- [9] P. K. Sonawane and S. Shelke, “Handwritten devanagari character classification using deep learning,” in *Int. Conf. on Information, Communication, Engineering and Technology (ICICET)*, pp. 1–4, India, 2018.
- [10] A. James, J. Manjusha and C. Saravanan, “Malayalam handwritten character recognition using alexnet based architecture,” *Indonesian Journal of Electrical Engineering and Information*, vol. 6, no. 4, pp. 393–400, 2018.
- [11] S. G. Lee, Y. Sung, G. Kim, and E. Y. Cha, “Variations of alexnet and googlenet to improve korean character recognition performance,” *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 205–217, 2018.
- [12] B. R. Kavitha and S. Chandrasekaran, “Benchmarking on offline handwritten tamil character recognition using convolutional neural networks,” *Journal of King Saud University-Computer and Information Sciences*, 2019.
- [13] J. Mahesh and S. Srivastava, “Handwritten devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods,” *Journal of Imaging*, vol. 4, pp. 41, 2018.
- [14] D. C. Cireşan, U. Meier and J. Schmidhuber, “Transfer learning for latin and chinese characters with deep neural networks,” in *Int. Joint Conf. on Neural Networks IJCNN*, Australia, pp. 1–6, 2012.
- [15] A. k. Tushar, A. Ashiquzzaman, A. Afrin, and M. R. Islam, “A novel transfer learning approach upon hindi, arabic, and bangla numerals using convolutional neural networks,” in *Computational Vision and Bio Inspired Computing. Lecture Notes in Computational Vision and Biomechanics*, vol. 28. Cham: Springer, 2018.



- [16] M. A. Pragathi, K. Priyadarshini, S. Saveetha, A. S. Banu and K. O. Mohammed Aarif, "Handwritten tamil character recognition using deep learning," in *Int. Conf. on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, India, pp. 1–5, 2019.
- [17] S. Sahoo, P. Kumar B. and R. Lakshmi, "Online handwritten character classification of the same scriptural family languages by using transfer learning techniques," in *3rd Int. Conf. on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, (2020), India, pp. 1–4, 2020.
- [18] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten arabic character recognition," *Cognitive Systems Research*, vol. 50, pp. 180–195, ISSN 1389-0417, 2018.
- [19] M. A. Kilvisharam Oziuddeen, S. Poruran and M. Y. Caffiyar, "A novel deep convolutional neural network architecture based on transfer learning for handwritten urdu character recognition," *Tehniki Vjesnik, Technical Gazette*, vol. 27, no. 4, pp. 1160–1165, 2020.
- [20] Y. Nanekaran, D. Zhang, S. Salimi, J. Chen, Y. Tian *et al.*, "Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of farsi handwritten digits," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3193–3222, 2021.
- [21] S. Ahmed, S. Naz, S. Swati, M. I. Razzak, A. Khan *et al.*, "UCOM offline dataset-an urdu handwritten dataset generation," *International Arab Journal Information Technology*, vol. 14, pp. 239–245, 2014.
- [22] M. F. Naeem, N. u. S. Zia, A. A. Awan, F. Shafait and A. ul Hasan, "Impact of ligature coverage on training practical urdu ocr systems," in *14th Int. Conf. on Document Analysis and Recognition (ICDAR)*, Japan, pp. 131–136 2017.
- [23] R. Z. Xiao, A. M. McEnery, J. P. Baker and A. Hardie, "Developing Asian language corpora: standards and practice", In *The 4th Workshop on Asian Language Resources*, Chicago, 2004.
- [24] A. Zafar, A. Mahmood, F. Abdullah, S. Zahid, S. Hussain *et al.*, "Developing urdu wordnet using the merge approach," in *Int. Conf on Language and Technology (CLT12)*, Pakistan. 2012. ([URL:http://www.cle.org.pk/clt12](http://www.cle.org.pk/clt12)).
- [25] P. Nucharee, Y. Sukanya and P. Wichian, "A scheme for salt and pepper noise reduction and its application for OCR systems," *WSEAS Transactions on Computers Archives*, vol. 9, pp. 351360, 2010.
- [26] A. A. Ahmed and S. Mallaiah, "A novel approach to correction of a skew at document level using an arabic script," *International Journal of Computer Science and Information Technologies*, vol. 8, no. 5, pp. 569–573. 2017.
- [27] Y. Li, Y. Zheng, D. Doermann and S. Jaeger, "Script-independent text line segmentation in freestyle handwritten documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1313–1329, 2018.
- [28] I. U. 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. 62, pp. 1–8, 2017.
- [29] S. Y. Arafat and M. J. Iqbal, "Two stream deep neural network for sequence-based urdu ligature recognition," *IEEE Access*, vol. 7, pp. 159090–159099, 2019.
- [30] N. H. Khan, A. Adnan, and S. Basar, "Urdu ligature recognition using multi-level agglomerative hierarchical clustering," *Cluster Computing*, vol. 21, no. 1, pp. 503–514, 2018.
- [31] I. Ahmad, X. Wang, R. Li, and S. Rasheed, "Offline urdu nastaleeq optical character recognition based on stacked denoising autoencoder," *China Communications*, vol. 14, no. 1, pp. 146–157, 2017.
- [32] Q. U. A. Akram and S. Hussain, "Ligature-based font size independent OCR for noori nastalique writing style," in *Int. Workshop on Arabic Script Analysis and Recognition (ASAR)*, pp. 129–133, Nancy, 2017.
- [33] I. Ahmad, X. Wang, Y. H. Mao, G. Liu, H. Ahmad *et al.*, "Ligature-based urdu nastaleeq sentence recognition using gated bidirectional long short term memory," *Cluster Computing*, vol. 21, no. 1, pp. 703–714, 2018.
- [34] M. J. Rafeeq, Z. U. 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, 2019.
- [35] A. Rana and G. S. Lehal, "Online urdu ocr using ligature based segmentation for nastaliq script," *Indian Journal for Science and Technology*, vol. 8, no. 35, pp. 19, 2015.
- [36] S. Shabbir and S. Imran, "Optical character recognition system for urdu words in nastaliq font," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, pp. 567576, 2016.