

Convolutional Neural Network Based Intelligent Handwritten Document Recognition

Sagheer Abbas¹, Yousef Alhwaiti², Areej Fatima³, Muhammad A. Khan⁴, Muhammad Adnan Khan^{5,*},
Taher M. Ghazal^{6,7}, Asma Kanwal^{1,8}, Munir Ahmad¹ and Nouh Sabri Elmitwally^{2,9}

¹School of Computer Science, National College of Business administration and Economics, Lahore, 54000, Pakistan

²College of Computer and Information Sciences, Jouf University, Sakaka, 72341, Saudi Arabia

³Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan

⁴Riphah School of Computing & Innovation, Faculty of Computing, Riphah International University, Lahore Campus, Lahore, 54000, Pakistan

⁵Pattern Recognition and Machine Learning Lab, Department of Software, Gachon University, Seongnam, 13557, Korea

⁶Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Selangor, Malaysia

⁷School of Information Technology, Skyline University College, University City Sharjah, 1797, Sharjah, UAE

⁸Department of Computer Science, GC University, Lahore, Pakistan

⁹Department of Computer Science, Faculty of Computers and Artificial Intelligence, Cairo University, 12613, Egypt

*Corresponding Author: Muhammad Adnan Khan. Email: adnan@gachon.ac.kr

Received: 22 June 2021; Accepted: 23 July 2021

Abstract: This paper presents a handwritten document recognition system based on the convolutional neural network technique. In today's world, handwritten document recognition is rapidly attaining the attention of researchers due to its promising behavior as assisting technology for visually impaired users. This technology is also helpful for the automatic data entry system. In the proposed system prepared a dataset of English language handwritten character images. The proposed system has been trained for the large set of sample data and tested on the sample images of user-defined handwritten documents. In this research, multiple experiments get very worthy recognition results. The proposed system will first perform image pre-processing stages to prepare data for training using a convolutional neural network. After this processing, the input document is segmented using line, word and character segmentation. The proposed system get the accuracy during the character segmentation up to 86%. Then these segmented characters are sent to a convolutional neural network for their recognition. The recognition and segmentation technique proposed in this paper is providing the most acceptable accurate results on a given dataset. The proposed work approaches to the accuracy of the result during convolutional neural network training up to 93%, and for validation that accuracy slightly decreases with 90.42%.

Keywords: Convolutional neural network; segmentation; skew; cursive characters; recognition



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.

1 Introduction

Character recognition is the field where many machine learning techniques are widely applied. The world is advancing towards paperless communication, but there are many fields where handwritten document sharing still exists in daily communication. A significant challenge in handwritten document recognition: the processing of distorted shapes of character and various writing styles. Secondly, proper segmentation techniques are required for this character-by-character processing. These handwritten words may generate many challenging tasks for researchers. The dataset consists of handwritten characters that may not necessarily be sharp enough and write perfectly in a straight line. Another issue is the curve of characters may not consistently smooth enough like printed characters. Different orientations and sizes of handwritten characters can also generate problems in processing. Finally, characters may not always be in their complete shape, which can fall into different categories and generate improper recognition in the recognition process.

Since the last few years, deep learning techniques have successfully performed their role in many fields, like, speech recognition, image and textual classification, face and facial expressions recognition, semantic-based video searching and many other areas. Many of experimented problems are re-experimented using deep learning to acquire significant results. The proposed method addresses the previously defined issues by developing an intelligent and efficient handwritten scripts recognition system. The development of such kinds of systems always demands extensive image processing and pattern recognition techniques to be part of these systems. This study reports the deep convolutional neural network technique to recognize handwritten scripts. The deep convolutional neural network consists of many hidden layers, and these layers comprise many neurons. An extensive dataset of 65000 handwritten characters' images is first prepared to train and test the proposed approach. Then, a deep convolutional neural network is properly trained on the defined dataset, and a separate testing phase applies to these handwritten scripts to check the accuracy of recognition. This handwritten script is given to the recognition system; it demands the proper segmentation of written lines, words and characters. The proposed approach develops three kinds of segmentation algorithms: line-based segmentation, word-based segmentation, and character-based segmentation of handwritten words. These segmentation techniques will separate each character of the script that will be recognized by the deep convolutional neural network later. This recognized handwritten script is immediately converted into an electronic text document. This proposed technique will generate a valuable contribution to the field of handwritten document recognition systems.

2 Related Work

Since the 1950s, handwriting recognition has been under investigation. For this, the application for new digital computer technology became the subject of interest. In 1968, Eden suggested the technique known as analysis-by-synthesis. In this proposed method, the author formally proved that all characters are consist of an infinite number of schematic features. As a result, many researchers have added their valuable contributions later in this field.

The work done by Grimsdale and Bullingham describes how the process of handwriting recognition can be simplified and speeded up. In this paper, the technology of the flying-spot uses a high-resolution scanner as a spot of light to read or scan an image [1].

In [2], the authors proposed a recognition system; its feature extraction phase aims to illustrate the pattern with the help of a minimum number of features used to discriminate different pattern classes. This paper use gradient representation o measure the direction and magnitude of

the enormous change in intensity in a minor neighborhood of every pixel. Through the Sobel operator, gradients are computed. The Sobel operator technique is mainly used for edge detection, where it creates an image emphasizing edges. In this paper, the obtained recognition accuracy of English Characters is 94%. The logical simplicity and easy use of the gradient features technique become the reason for this technique's popularity for recognition purposes [3].

In research [4], authors proposed handwriting recognition using fuzzy theory. The proposed method consists of two main phases, pattern recognition and feature extraction, respectively. First, the fuzzy technique is used for pattern recognition to fetch the fuzzy patterns. The problem in this method with handwritten characters is that every character has a different shape, size, and position because of different writing styles [5].

The proposed approach describes the effects of changing the models of Artificial Neural Networks to recognize the characters in the input document [6]. The paper explains the behavior of different Neural Network models that are used in Optical Character Recognition. Different parameters are considered in their proposed work, such as hidden layers, no. of neurons used in each layer and epochs, etc. They use Multilayer Feed Forward and Backward network for the recognition of characters. Their proposed work consists of phases like pre-processing, segmentation of characters, normalizing and de-skewing [7].

In [8], the authors propose a new technique, 'diagonal based feature extraction, to recognize handwritten alphabets. A multilayer feed-forward neural network is used for this purpose. The system performs a high level of accuracy compared to other conventional horizontal or vertical based feature extraction [8].

In [9], the authors develop a system using MATLAB. It acquires the image and converts it into a greyscale image. This preprocessed image then uses for recognition purposes. Multilayer perceptron (MLP) neural network is used for the recognition of characters. An MLP uses the backpropagation technique of a neural network in which each neuron of every layer is fully connected to the neuron of the next layer. Every node of the layer is work as an individual neuron except the input nodes. It decreases the training time and cost [9].

3 Methodology

Fig. 1 is defining the proposed method, which consists of multiple sub-phases. The first stage will get the written script from the environment in the image acquisition phase. Then this image will transfer to the pre-processing stage for the grey scaling, binarization and skew correction of the input image. Then this handwritten script will send to the next stages for line, word and character segmentation. Finally, the adequately segmented characters will send to the feature extraction phase. Finally, CNN configured layers will perform the training phase; after this, the character is classified in any of 26 characters' classes.

3.1 Image Acquisition

Image acquisition is the primary phase of a handwritten character recognition system. The proposed recognition system takes a scanned handwritten character image as an input. To start the process, the user should upload the image of the handwriting

3.2 Pre-Processing

Pre-processing is used to increase the quality of the image. It involves the following operations.

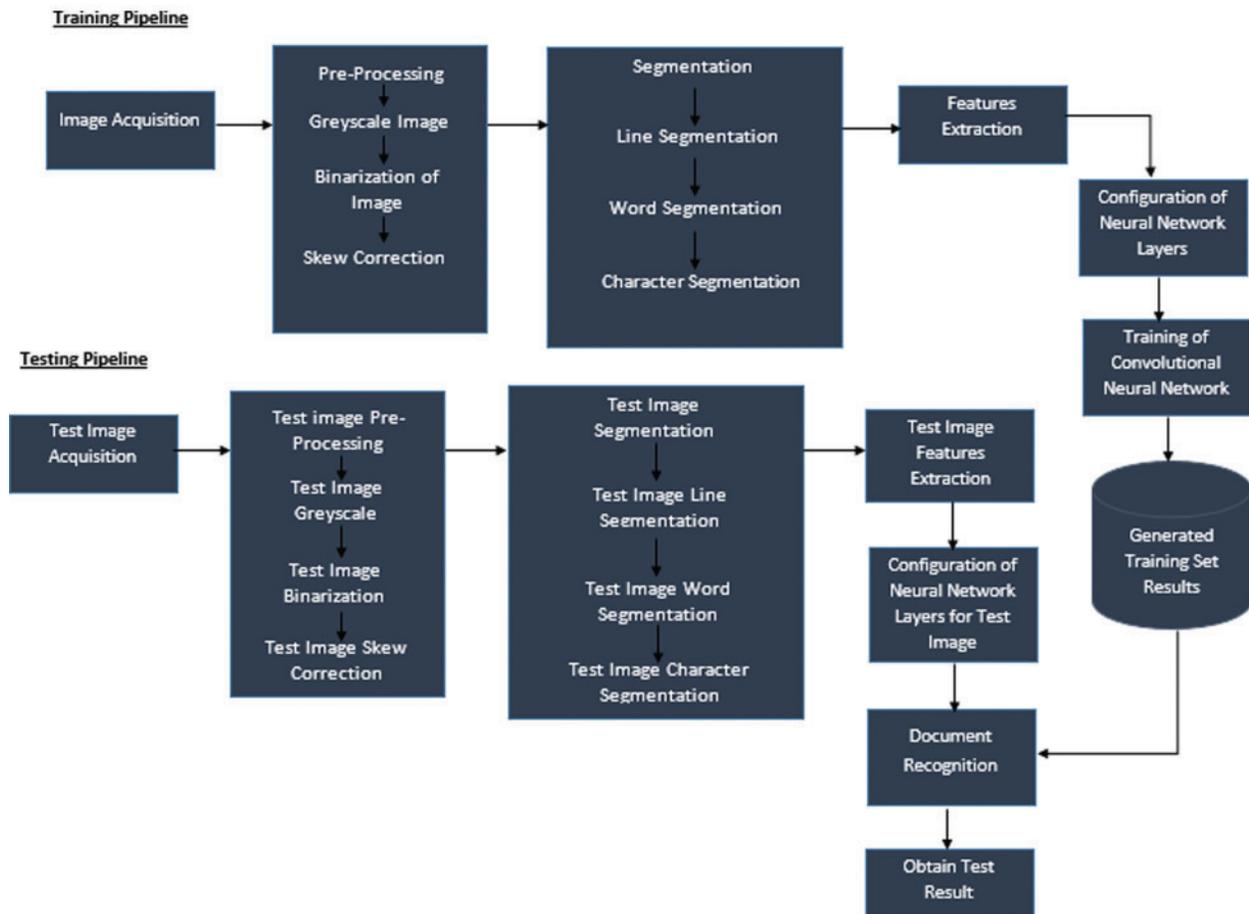


Figure 1: Proposed model for training and testing of the handwritten document

3.2.1 Grayscale Conversion

The original image is firstly converted into a grayscale image [10]. This is converting the RGB values (24 bit) into grayscale values (8 bit).

3.2.2 Noise Removal

Noise means unwanted information which disturbs the quality of the image. It means pixels in the image have different intensity values than the actual pixel values. Filters are the way in image processing to eliminate noise from the input image. For example, using a median filter is an efficient technique used to remove salt and pepper noise.

This proposed method is using a median filter to remove salt and pepper noise [10]. The median filter will substitute the current pixel value of the image with its median value. In median filtering, it first performs numerical order sorting on the pixel values from adjacent pixels and then exchanges the pixel under consideration with the middle pixel value.

3.2.3 Binarization

It is a part of the pre-processing process. This will convert a gray-scale image into a black and white image [10]. Binarizing the image will invert the pixel values. In this, black is represented by 0 bit, and white is represented by 1 bit.

3.3 Segmentation

Segmentation of characters is a crucial step in handwriting recognition because it directly affects the accuracy of the system [11]. Therefore, the accuracy of this recognition process will become better when the characters are correctly segmented. In the proposed work, the segmentation process is divided into three phases Line Segmentation, Word Segmentation, and Character Segmentation.

3.3.1 Line Segmentation

In the line segmentation, the input image has dark background pixels and white foreground pixels. Therefore, there is a possibility that text can touch the top and bottom lines. Removing such errors requires first pad the image to make a black space at the top and bottom in the image. Then, it will help to calculate the dark centroids in the image. The method proposed for line segmentation is based on the idea of projection problem taken from the Algorithm 2.1 “Projection Profile-Based Algorithm” presented in [11] with some amendments. In that study, the horizontal projection of the image is found, and then through that projection, they find maxima and minima of the image for further processing. In this proposed method, after finding the vertical projection, the non-text regions are marked. Then, the centroids of those dark regions are found, and the segmentation points for lines will consider as dark centroids. This algorithm does not work well for the images with skew angles, as shown in Fig. 2a. A skew angle is the direction of the text baseline. It is the clockwise or anti-clockwise orientation of text baseline concerning horizontal frame [12]. Skew can be of the following types:

- (1) *Negative Skew*: In it, the direction of the text baseline goes from the bottom left to the top right, in an upward direction; it is called “negative skew.”
- (2) *Zero-Skew*: The text baseline is parallel to the horizontal frame; it is zero skewed.
- (3) *Positive Skew*: The direction of the text baseline comes from the top left to bottom right in the downward direction; it is positive skew.

In the skew correction, the algorithm “Skew Detection using Center of Gravity” presented in [13] is used with some modifications. The applied algorithm can correct a slight skew angle, not corrected as shown in Fig. 2b. Wherever in the proposed method, whenever the line’s slope is calculated, it is checked that the line must be a straight line or an approximate straight line. The result of this method is shown in Fig. 2c. The proposed Modified Skew Detection using the Center of Gravity algorithm is given in Algorithm #1.

Algorithm #1: Modified skew detection using center of gravity

- 1 Divide the text image into two equal halves vertically.
 - 2 For both left and right halves, compute the center of gravity.
 - 3 Join the two centers of gravity by a line.
 - 4 Compute slope of the line.
 - 5 If the value of the slope is not between -0.1 and 0.1 , then skew is corrected
-

(Continued)

-
- 6 Else
- (a) By taking the inverse of the slope, calculate the skew angle
 - (b) Rotate the signature by detected skew angle.
 - (c) Repeat steps 1 to 6 until the condition at Step 5 becomes true
-

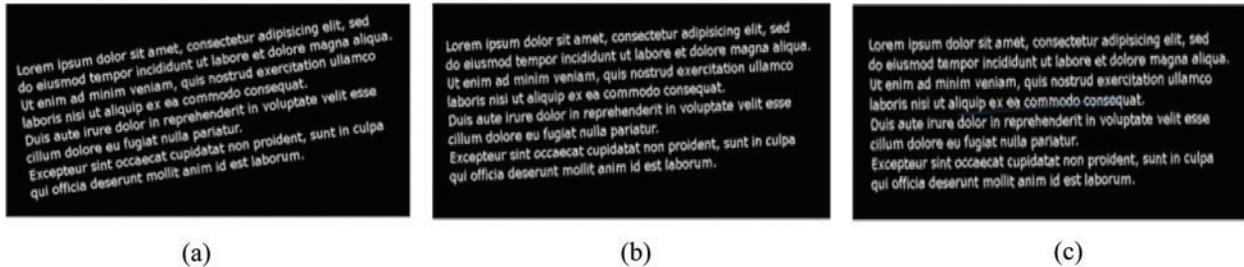


Figure 2: Skew correction (a) Image with negative skew angle (b) Skew corrected with referenced algorithm (c) Skew corrected with proposed method

Algorithm #2: Line segmentation:

- a.1 Pad the image.
 - a.2 Find Horizontal projection by summing the image horizontally. i.e., Find the sum of each row. For example, as shown in Fig. 3a.
 - a.3 Find the dark lanes between the lines with letters on them by setting a threshold value. i.e., 0 for a binary image.
 - a.4 Label each dark region using the bwlabel function of MATLAB.
 - a.5 Find Number of dark regions.
 - a.6 Find centroids of those dark regions using the regionprops function of MATLAB
 - a.7 Store all centroids into an array.
 - a.8 Find all x-centroids and y-centroids
 - a.9 Now crop the region of texts by assuming a line of text between 2 consecutive y-centroids.
 - a.10 Store all the cropped regions into a separate array. Now each line is segmented correctly, as shown in Fig. 3b.
-

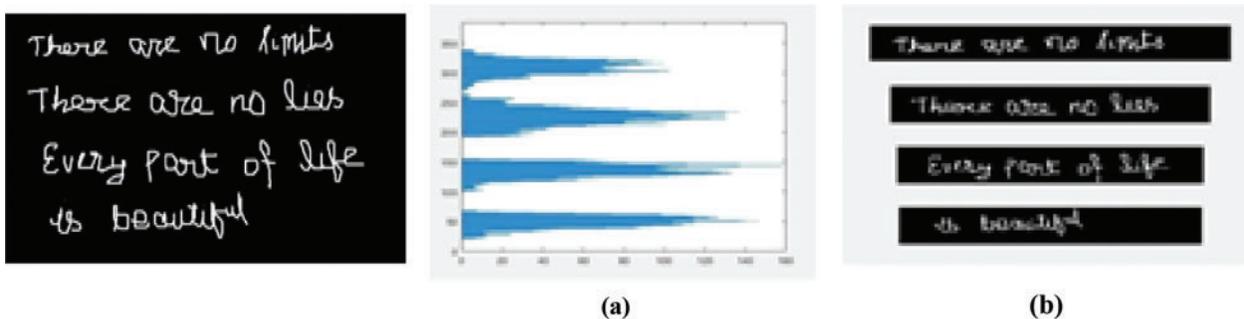


Figure 3: Line segmentation. (a) Histogram (b) Line segmentation correctly

3.3.2 Word Segmentation

The Line segmentation further demands the segmentation of words. In the word segmentation, morphology is applied to the image [11]. This process is done by dilating the image; in the result,

it will connect the near parts of characters [14]. Dilation is a primitive morphological process that raises or condenses items into a binary image. Fig. 4a, referred to as a shaping component, is used to regulate the precise method and magnitude of this thickening. Shaping components are small cliques or sub-images used to review an image under consideration for features of interest. The dilation process is mainly used with line segmentation, but in the proposed work author is using it for word segmentation. In the dilation process of the image, the characters in words are connected, and those connected components can be easily extracted from the image. The extraction of those connected components demands proper labeling of these words, and this proposed work uses MATLAB bwalabel labeling function. Those labels are cropped from the line image, and as a result, the words are extracted. The word segmentation is done through Algorithm #3.

Algorithm #3: Word segmentation

1. Dilate the line image using strel and imdilate function of MATLAB. As shown in Fig. 4b.
 2. Label the connected components using bwlabel function of MATLAB. As shown in Fig. 4c.
 3. Separate each label from the image. The separated images will be the words extracted from the line. As shown in Fig. 4d.
-

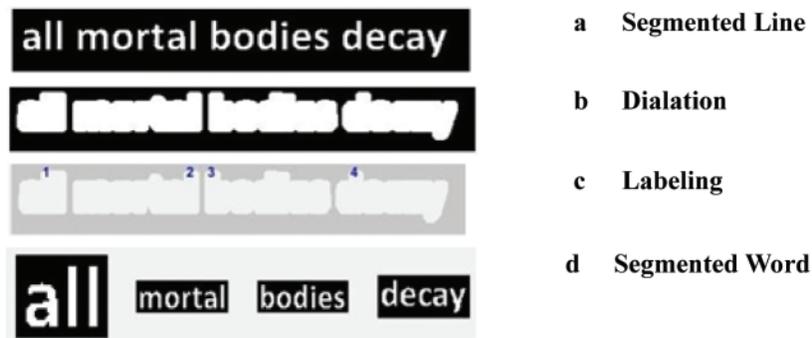


Figure 4: Word segmentation

3.3.3 Character Segmentation

In handwritten document generation, sometimes the negative or positive slant occur in the written words, which demands the slant correction [15]. Slant correction will help to perform character segmentation. Handwritten characters could be cursive and untouched characters.

- (1) Cursive character segmentation
- (2) Untouched character segmentation

Cursive Character Segmentation In cursive segmentation, the characters are segmented from the word image of cursive handwriting. In this segmentation, the main challenge is to avoid mis-segmentation and over-segmentation. Mis-segmentation means the characters that had to be segmented are not segmented properly. Over-segmentation means that a single character is segmented as two characters, just like 'm' can be segmented as two n's, and we can be segmented as two v's. This problem can be avoided through the combination of algorithms presented in [16].

In the first step, the word image is skeletonized, and the vertical projection of the word is calculated by summing all the columns. The vertical projection can determine ligatures between characters as they will have only one foreground pixel in the perspective column. Sometimes, over-segmentation can occur in these characters 'm', 'n', 'u', 'v', 'w', etc. The over-segmentation of 'm' and 'n' can be avoided through the midpoint of the height of the image is calculated. The image is scanned vertically for each column; if the white pixel finds, its position is determined. In that position, if the pixel is below the midpoint of height, then that column may be a segmentation column. In the end, if the sum of that column is more significant than one, then this column is discarded. In another case, it may not be the joining point, and it is stored as a potential segmentation column. The over-segmentation problem for these characters ('u', 'v', 'w') can be avoided through the distance-based approach. If the distance between this column and the previous segmentation column is less than the given threshold value, then it is discarded, and this process is repeated for the next columns. Fig. 5a present the slant words and Fig. 5b define slant correction. Whereas Fig. 6. represent the working of Algorithm #4.



Figure 5: (a) Slant words (b) Slant correction

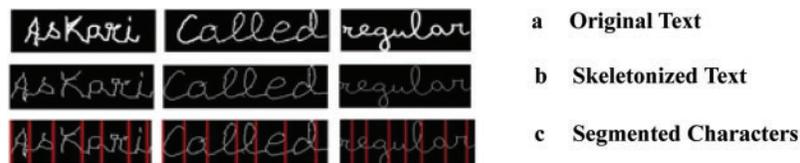


Figure 6: Cursive character segmentation

Algorithm #4: Cursive character segmentation

- a.1 Skeletonize the image using bwmorph function of MATLAB.
 - a.2 Find Vertical projection (vp) of the image by summing all the columns.
 - a.3 Create an array for segmentation points SP
 - a.4 Store first Segmentation Point SP (1) at the start of the first character.
 - a.5 Find the midpoint of the height of the image by finding the midpoint of the number of rows present in the image.
 - a.6 Repeat for each column i starting from 1st SP
 - 6.1. Repeat for each row j starting from row 1
 - 6.1.1. Check If image(j,i) is 1 AND row number (j) > midpoint AND the vertical projection of that column vp (i) <= 1
 - 6.1.1.1. If the distance of column i from the previous SP is greater than the threshold value
-

(Continued)

-
- 6.1.1.1.1 Store this column in SP array then Break
 - 6.1.1.2. Else Break
 - 6.1.2. Else Break
 - 6.2. Else Check if there is no white pixel in that column, then Store this column in SP array
If its distance with previous SP is greater than the threshold
 - 6.2.1 Else break
 - a.7 For each SP draw a red line on the word image for visualizing segmentation points.
 - a.8 Store 0 in all rows of all Segmentation Points SP
 - a.9 Now all the touched characters are disconnected.
 - a.10 For segmentation of these disconnected characters algorithm is given in the following section.
-

Untouched Character Segmentation: Segmentation of untouched characters is much easier than touched characters. In that process, the line segmentation method is used. It will find black spaces between characters and make a separation between characters. In this type of character segmentation, vertical projection is calculated instead of horizontal projection.

Algorithm #5: Untouched character segmentation

- 1 Pad the left and right sides of the image.
 - 2 Find Vertical projection by summing the image vertically. i.e., Find the sum of each column, as shown in [Fig. 7b](#).
 - 3 Find the dark lanes that define the zones between the letters by setting a threshold value. i.e., 0 for a binary image.
 - 4 Label each dark region using bwlabel function of MATLAB.
 - 5 Find Number of dark regions.
 - 6 Find centroids of those dark regions using regionprops function of MATLAB
 - 7 Store all centroids into an array.
 - 8 Find all x-centroids and y-centroids
 - 9 Now crop the region of texts by assuming a line of text between 2 consecutive x-centroids.
 - 10 Store all the cropped regions into a separate array. Now each character is segmented properly. As shown in [Fig. 7c](#).
-

3.4 Feature Extraction

Feature extraction is the process of detecting the features of interest from an image and storing them for further processing. In image processing, feature extraction is a critical step that allows moving from pictorial to data representation. The proposed work is using a Convolutional Neural Network for feature extraction [17].

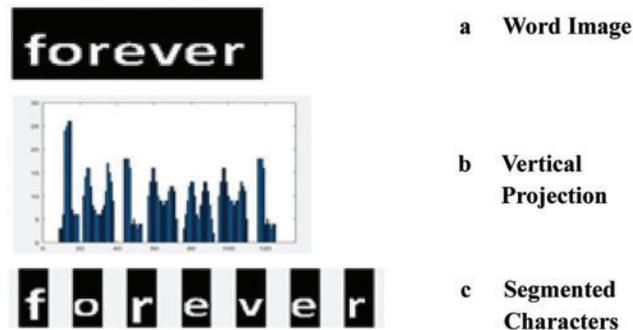


Figure 7: Untouched character segmentation

3.4.1 Creating a Convolutional Neural Network

The proposed method is using a Convolutional neural network (CNN). CNN contain neurons with learnable weights and biases. CNN can contain multiple layers, which are also known as deep learning. CNN is a feed-forward neural network that can contain one or more convolutional layers. One or more fully connected layers follow CNN, just like in a simple multiple layer neural network. The architecture of CNN is modeled so that it can use the 2D structure of the image as an input of a Neural Network. The proposed configuration of the neural network is using the back-propagation technique for the training of a CNN. Traditional CNN consist of layers defines in Fig. 8. Whereas Fig. 9. define in detail the layering scheme of configured CNN.

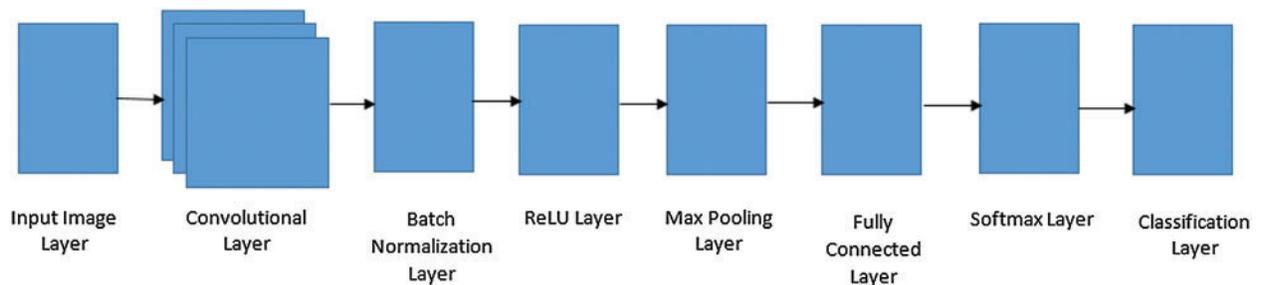


Figure 8: Layers of convolutional neural network

3.4.2 Training of Convolutional Neural Network

The configuration of the neural network then demands the training of this neural network. The overall dataset set is split into two sections, training data, and validation data to achieve this step. The sample data for training consist of handwritten images of different writing styles. The dataset of 65000 handwritten characters is prepared for the training and validation of the neural network. There are 2500 sample images of each alphabet, and the total sample images are $2500 * 26 = 65000$. Sample image is a gray-scale image that has a white foreground and black background. The neural network is trained by the initial learning rate of 0.005. The maximum training accuracy of CNN is recorded as 97%. The trained neural network is saved, and then it is used for the recognition process.

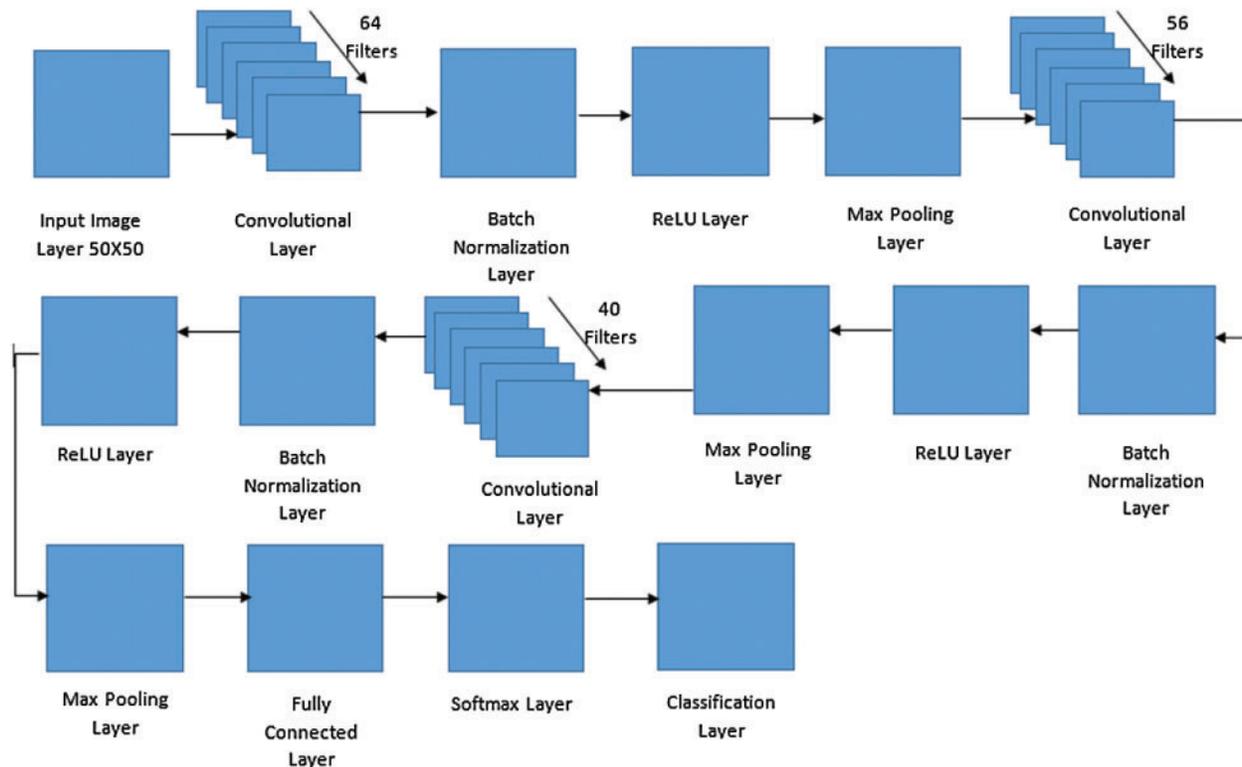


Figure 9: Configured layers of convolutional neural network

3.4.3 Document Recognition

For image recognition, the neural network comprises four primary operations defined in the next sections.

Convolution: The CNN use a convolutional operator to extract features from an input image. The spatial association between pixels Convolution is preserved by learning image features using the sub-matrix of the input image. Every image can be expressed as a matrix contains pixel values. For example, let's suppose the image matrix of $5 * 5$ size consists of binary values and a filter matrix of $3 * 3$ size. The convolution operator works as the filter matrix is placed over the image matrix. Every value of the filter matrix is multiplied by the image matrix's corresponding value, and then the filter matrix is moved by one pixel, and the same step is repeated. The step moved or jumped is called stride in the proposed work stride one. The resulting matrix is the convolution matrix.

CNN depends on the following things:

- (1) *Depth:* Number of filters
- (2) *Stride:* Filter matrix slides over the image by some number of pixels; those number of pixels are called Stride.
- (3) *Zero-padding:* Sometimes, zeros are padded around the border; it is used that the filter can be applied to border elements of the input image matrix. It is also known as wide convolution.

In the convolutional layer, the convolution (\bullet) is the dot product between inputs image M and the filter matrix N . The output of the whole process is a convoluted features matrix represented as $icon$. The $icon$ can be calculated using Eqs. (1) and (2).

$$f_{con} = convolution(k, l) \quad (1)$$

$$(M \otimes N)(k, l) = \sum_m \sum_n M(m, n) N(k - m, l - n) \quad (2)$$

Non-Linearity (ReLU): ReLU stands for Rectified Linear Unit; it is used to perform a non-linear process. ReLU performs an element-wise operation. It is applied to each pixel, and it substitutes all negative pixel values by zero in the extracted feature maps. It retains only nonnegative value pixels in a feature map.

The mathematical model of the ReLU function consists of the piecewise nonlinearity operator that defines the maximum output indication. Thus, the ReLU function is represented as $ReLU(\bullet)$. The output of the ReLU function will be the rectified feature map, $irec$, which can be calculated using Eq. (3).

$$f_{rec} = ReLU(x_k) = max(O, x_k) \quad (3)$$

Pooling Step: Pooling is the process of defining sub-sampling or downsampling. This feature reduces the dimensionality of the feature maps and also preserves the essential information. This study is using the Max pooling technique for spatial pooling. In Max pooling, a spatial matrix is defined as $2 * 2$, and the maximum portion from that area is selected.

Mathematically the max-pooling function can be Pool (\bullet) defined as shown in Eq. (4).

$$f_{pl} = pool(k, l) = max(x_{k,l}) \quad (4)$$

Fully Connected Layer: The fully connected layer is used for classification. A fully connected layer means that every neuron of a layer is connected to every neuron of another layer. The output of the convolution and pooling layer is fed to full connected layers, and it performs classification.

Softmax Layer: A softmax layer trained on the handwritten alphabets will output a separate probability for each of the 26 alphabets, and the probabilities will add up to one. Thus, the softmax activation function in the output layer of a deep neural network is to express a categorical dispersal over class labels. Thus, the probabilities of each input element are obtained that belong to a label.

The softmax function $Softmax(\bullet)$, is a multiclass classifier. The i th probabilistic output of that function can be calculated using Eq. (5).

$$p(O_i) = softmax(x_k) = \frac{exp(x_k)}{\sum_{g=1}^y exp(x_k)} \quad (5)$$

Classification Layer: The classification layer classifies the output obtained from the softmax layer.

4 Obtained Results

Matlab 2017b is being used to find the overall accuracy of the proposed work and dataset. In this study, the accuracy of the proposed method and dataset is demonstrated by using the CNN algorithm.

4.1 Application View

The application view is defined in Fig. 10, wherein the first stage needs to upload the image for the image acquisition process. In the preprocessing stage, it will perform all preprocessing phases defined previously in Section 3.2. After preprocessing according to the input image, the user will choose the segmentation type (cursive or untouched). Next, the system will perform all types of segmentation phases defined in Section 3.3. Then CNN will configure automatically for the recognition of segmented characters. On completing the recognition phase, the recognized characters will be stored in a simple notepad file for verification of the system. The simulated system consists of seven layers. Input Layer, Convolutional Layer, batch normalization layer, ReLU layer, max-pooling layer, fully connected layer, softmax layer and classification layer. The system consists of three convolutional layers in which first convolutional layer comprises 64 filters, second convolutional layer contains 56 filters, and the last convolutional layer consists of 40 filters. Thus, systems have three batch normalization layers, three ReLU layers, three max-pooling layers, one fully connected layer, one softmax layer and one classification layer. Max pooling layer have kernel size of $2 * 2$ matrix.

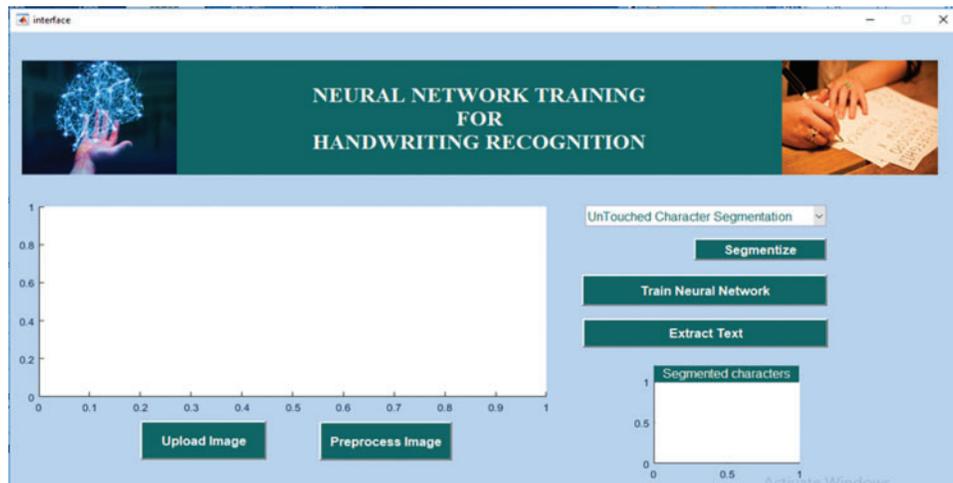


Figure 10: Application view

Fig. 11 defines the accuracy of the system during the segmentation phase. Fig. 11 shows that the system's accuracy will get better during the cursive handwriting each character skew corrected and is equally distanced aligned. Tab. 1 shows the overall accuracy of the segmentation phase of a system for each character.



Figure 11: Cursive segmentation examples

Table 1: Cursive segmentation accuracy

| Character | Test parameters | | | |
|-----------|---------------------|-------------------------------|--------------------------|--------------------------|
| | Number of instances | Correctly segmented instances | Over segmented instances | Miss segmented instances |
| A | 50 | 45 (90%) | 3 (6%) | 2 (4%) |
| B | 50 | 48 (96%) | 1 (2%) | 1 (2%) |
| C | 50 | 47 (94%) | 2 (4%) | 1 (2%) |
| D | 50 | 48 (96%) | 1 (2%) | 1 (2%) |
| E | 50 | 46 (92%) | 1 (2%) | 3 (6%) |
| F | 50 | 47 (94%) | 1 (2%) | 2 (4%) |
| G | 50 | 45 (90%) | 3 (6%) | 2 (4%) |
| H | 50 | 42 (84%) | 6 (12%) | 2 (4%) |
| I | 50 | 35 (70%) | 2 (4%) | 13 (26%) |
| J | 50 | 46 (92%) | 1 (2%) | 3 (6%) |
| K | 50 | 43 (86%) | 5 (10%) | 2 (4%) |
| L | 50 | 40 (80%) | 1 (2%) | 9 (18%) |
| M | 50 | 35 (70%) | 13 (26%) | 2 (4%) |
| N | 50 | 40 (80%) | 8 (16%) | 2 (4%) |
| O | 50 | 49 (98%) | 0 (0%) | 1 (2%) |
| P | 50 | 48 (96%) | 2 (4%) | 0 (0%) |

(Continued)

Table 1: Continued

| Character | Test parameters | | | |
|-----------|---------------------|-------------------------------|------------------------------|------------------------------|
| | Number of instances | Correctly segmented instances | Over segmented instances | Miss segmented instances |
| Q | 50 | 42 (84%) | 6 (12%) | 2 (4%) |
| R | 50 | 41 (82%) | 7 (14%) | 2 (4%) |
| S | 50 | 44 (88%) | 4 (8%) | 2 (4%) |
| T | 50 | 40 (80%) | 3 (6%) | 7 (14%) |
| U | 50 | 41 (82%) | 8 (16%) | 1 (2%) |
| V | 50 | 40 (80%) | 8 (16%) | 2 (4%) |
| W | 50 | 38 (76%) | 10 (20%) | 2 (4%) |
| X | 50 | 42 (84%) | 6 (12%) | 2 (4%) |
| Y | 50 | 42 (84%) | 6 (12%) | 2 (4%) |
| Z | 50 | 45 (90%) | 4 (8%) | 1 (2%) |
| | | Total average accuracy | Total average over segmented | Total average miss segmented |
| | | 86% | 8.615% | 5.308% |
| | | Maximum accuracy achieve | Maximum over segmented | Maximum miss segmented |
| | | 98% | 26% | 26% |

Fig. 12 shows the system performance during the uncorrected cursive segmentation. Again, the system will incorrectly segment the characters if they are too skewed and have an equal distance.

**Figure 12:** Incorrect cursive segmentation

4.2 Training/Testing of CNN

Fig. 13 is showing the overall system performance during the Training/Testing phase of CNN. The first graph shows the accuracy of the CNN in the training and validation phase. Where the second graph shows the loss ratio in the training and validation phase. The learning rate of the CNN was 0.01 with a piecewise threshold function. The training cycle runs for 1680 of 2225 iterations for convergence. Where the iterations per epoch are 89. For the number of an epoch, the training cycle run is 19 of 25. It takes 107 min and 50 s for the training of the system on

given samples. Training finished on the criteria meet. The validation frequency of CNN is 120 iterations, and the validation accuracy of the system is 90.89%.

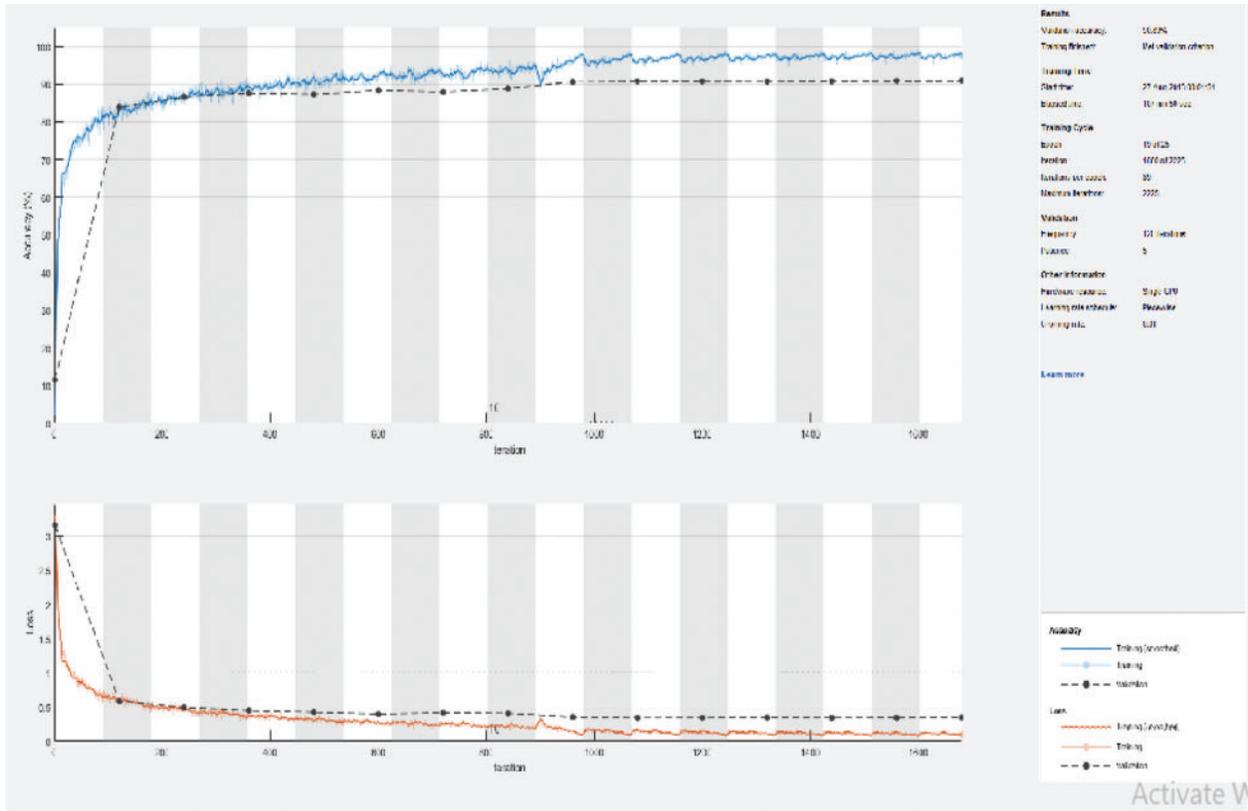


Figure 13: CNN training/testing phase performance

Tab. 2, defines the average accuracy of the system in the perspective of recognition of each character in the validation step. The accuracy of the system depends on the writing style of the writer. If the writing style is untouched, the system will perform much better than the performance defined in Tab. 2. The English script gives the system for validation purposes. The script can be in cursive and untouched writing style form. Then given image is firstly pass through the pre-processing process. Then after segmentation, each character is separately sent to the system for verification purposes.

5 Discussion

The solution's average accuracy reached 90% in the proposed solution, which is an excellent result obtained in the handwritten document recognition on the given dataset. This study is helpful to the design and development of handwritten Optical Character Recognition Systems in future. The comparison between the recognition accuracy of previous studies and the proposed method is demonstrated in Tab. 3.

In this study, the essential contribution by the authors are:

- Designing a valid dataset that is used to train the systems efficiently can be trained for both printed and handwritten documents.
- Designing of new algorithms for line, word and character segmentation of cursive and non-cursive handwriting.
- Find every possible writing style for every alphabet, joining style with other alphabets in English language alphabets.

Table 2: Parameters

| Characters | Test parameters | | | |
|------------|-----------------|-----------------------|--------------------------------|----------------------------------|
| | Train instances | Validation instances | Correctly recognized instances | Incorrectly recognized instances |
| A | 2400 | 100 | 90 (90%) | 10 (10%) |
| B | 2400 | 100 | 92 (92%) | 8 (8%) |
| C | 2400 | 100 | 94 (94%) | 6 (6%) |
| D | 2400 | 100 | 92 (92%) | 8 (8%) |
| E | 2400 | 100 | 91 (91%) | 9 (9%) |
| F | 2400 | 100 | 88 (88%) | 12 (12%) |
| G | 2400 | 100 | 90 (90%) | 10 (10%) |
| H | 2400 | 100 | 92 (92%) | 8 (8%) |
| I | 2400 | 100 | 72 (72%) | 28 (28%) |
| J | 2400 | 100 | 90 (90%) | 10 (10%) |
| K | 2400 | 100 | 85 (85%) | 15 (15%) |
| L | 2400 | 100 | 83 (83%) | 17 (17%) |
| M | 2400 | 100 | 97 (97%) | 3 (3%) |
| N | 2400 | 100 | 88 (88%) | 12 (12%) |
| O | 2400 | 100 | 94 (94%) | 6 (6%) |
| P | 2400 | 100 | 92 (92%) | 8 (8%) |
| Q | 2400 | 100 | 92 (92%) | 8 (8%) |
| R | 2400 | 100 | 87 (87%) | 13 (13%) |
| S | 2400 | 100 | 94 (94%) | 6 (6%) |
| T | 2400 | 100 | 97 (97%) | 3 (3%) |
| U | 2400 | 100 | 92 (92%) | 8 (8%) |
| V | 2400 | 100 | 83 (83%) | 17 (17%) |
| W | 2400 | 100 | 88 (88%) | 12 (12%) |
| X | 2400 | 100 | 91 (91%) | 9 (9%) |
| Y | 2400 | 100 | 86 (86%) | 14 (14%) |
| Z | 2400 | 100 | 97 (97%) | 3 (3%) |
| | | Total accuracy | 90.423% | 9.577% |

Table 3: Comparison with existing solutions

| Algorithms | Training | | Validation | |
|------------------------|-----------------------|----------------|-----------------------|----------------|
| | Average accuracy in % | Miss rate in % | Average accuracy in % | Miss rate in % |
| Acharjya et al. [1] | 92.4% | 7.6% | 94% | 16% |
| Simayi et al. [18] | 89.67% | 10.33% | 86.04% | 13.96% |
| Sasipriyaa et al. [19] | 93.4% | 6.6% | 95.36% | 4.64% |
| Konkimalla et al. [20] | 91.6% | 8.4% | 89.7% | 10.3% |
| Honey et al. [21] | 88.67% | 11.33% | 86.55% | 13.45% |
| Proposed technique | 93% | 7% | 90.423% | 9.577% |

6 Conclusion

This paper proposed a Convolutional Neural Network and different segmentation approaches for Recognition of Handwritten English documents. The proposed technique is to train and test on the standard user-defined dataset prepared for the proposed system. From experimental results, it is observed that the proposed technique provides the best accuracy rate. The proposed system is currently achieving 90.42% accuracy in the validation phase. This decrease in accuracy due to many factors like a distorted stroke in writing, multiple sizes and thickness of characters, different writing styles, illumination of writing and many others. In future, the accuracy level can be further improved by modifying segmentation techniques in line, word and character segmentation and indulging more intermediate layers and filters in convolution neural networks.

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] P. P. Acharjya, S. Mukherjee and D. Ghoshal, "Digital image segmentation using median filtering and morphological approach," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 1, pp. 552–557, 2014.
- [2] A. Aggarwal, R. Rani and R. Dhir, "Handwritten devanagari character recognition using gradient features," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 5, pp. 15–23, 2021.
- [3] A. E. Mouatasim, "Fast gradient descent algorithm for image classification with neural networks," *Signal Image and Video Processing*, vol. 14, no. 8, pp. 1565–1572, 2020.
- [4] P. N. Ekhande, S. A. Rumane and M. A. Ahire, "Document image binarization using image segmentation algorithm in a parallel environment," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 1, pp. 360–366, 2015.
- [5] H. Akouaydi, S. Njah, W. Ouarda, A. Samet and T. Dhieb, "Neural architecture based on fuzzy perceptual representation for online multilingual handwriting recognition," *Journal of Latex Class Files*, vol. 5, no. 3, pp. 1–12, 2019.
- [6] W. Liu, Y. Wen, Z. Yu and M. Yang, "Large-margin softmax loss for convolutional neural networks," *International Conference on Machine Learning*, vol. 2, no. 3, pp. 7–19, 2016.

- [7] L. B. Mahanta and A. Deka, "Skew and slant angles of a handwritten signature," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 9, pp. 2030–2034, 2013.
- [8] J. Pradeep, E. Srinivasan and S. Himavathi, "Diagonal based feature extraction for handwritten character recognition system using neural network," in *Int. Conf. on Electronics Computer Technology*, Kanyakumari, India, pp. 364–368, 2011.
- [9] C. I. Patel, R. Patel and P. Patel, "Handwritten character recognition using neural network," *International Journal of Scientific & Engineering Research*, vol. 2, no. 5, pp. 123–132, 2015.
- [10] Puneet and N. K. Garg, "Binarization techniques used for greyscale images," *International Journal of Computer Applications*, vol. 71, no. 1, pp. 8–11, 2013.
- [11] H. R. Mamatha and K. Srikantamurthy, "Morphological operations and projection profiles-based segmentation of handwritten Kannada document," *International Journal of Applied Information Systems*, vol. 4, no. 5, pp. 12–18, 2012.
- [12] M. Manomathi and S. Chitrakala, "Skew angle estimation and correction for noisy document images," *International Conference on Advances in Computing and Communications*, vol. 4, no. 7, pp. 415–424, 2011.
- [13] R. Ptak, B. Żygadło and O. Unold, "Projection-based text line segmentation with a variable threshold," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 1, pp. 195–206, 2017.
- [14] J. Ryu, H. I. Koo and N. I. Cho, "Word segmentation method for handwritten documents based on structured learning," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1161–1165, 2015.
- [15] H. Mehta, S. Singla and A. Mahajan, "Optical character recognition system for roman script and english language using artificial neural network classifier," *Int. Conf. on Research Advances in Integrated Navigation Systems*, vol. 7, no. 3, pp. 110–124, 2016.
- [16] K. C. Prakash, Y. M. Srikar, G. Trishal, S. Mandal and S. S. Channappayya, "Optical character recognition for telugu: database, algorithm and application," in *IEEE Int. Conf. on Image Processing*, Athens, Greece, pp. 40–47, 2018.
- [17] H. Cevikalp, G. G. Dordinejad and M. Elmas, "Feature extraction with convolutional neural networks for aerial image retrieval," in *Signal Processing and Communications Applications Conf.*, Turkey, pp. 1–10, 2017.
- [18] W. Simayi, M. Ibrayim and A. Hamdulla, Character type based online handwritten Uyghur word recognition using recurrent neural network. in: *Wireless Networks*. Berlin, Germany: Springer, 2021.
- [19] N. Sasipriyaa, P. Natesan, R. S. Mohana, E. Gothai, K. Venu *et al.*, "Design and simulation of handwritten detection via generative adversarial networks and convolutional neural network," *Materials Today: Proceedings*, In press, pp. 1–4, 2021. <https://doi.org/10.1016/j.matpr.2021.05.024>.
- [20] K. C. Prakash, Y. M. Srikar, G. Trishal, S. Mandal and S. S. Channappayya, "Optical character recognition for Telugu: Database Algorithm and Application," in *25th IEEE Int. Conf. on Image Processing*, Athens, Greece, pp. 10–17, 2018.
- [21] H. Mehta, S. Singla and A. Mahajan, "Mahajan optical character recognition system for roman script and english language using artificial neural network classifier," in *Int. Conf. on Research Advances in Integrated Navigation Systems*, Bangalore, India, pp. 1–8, 2016.