Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766357

HYBRID HANDWRITTEN TEXT RECOGNITION SYSTEM USING CONVOLUTIONAL NEURAL NETWORK AND RECURRENT NEURAL NETWORK

J. Pradeep[1], M. Harikrishnan[2], K.N. Prasanth[3], R. Sriram[4] and M. Thanush[5]

1. Associate Professor, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College, Puducherry. pradeepj@smvec.ac.in
2. Assistant Professor, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College, Puducherry. harikrishnan@smvec.ac.in

3,4,5. UG Students, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College, Puducherry. prasanth241718@gmail.com , sriramradhakrishnan2002@gmail.com , thanushmagesh@gmail.com

Abstract:
The offline handwritten text recognition is critical tasks, which need to be accomplished to move towards a paperless environment. In this paper, a hybrid handwritten text recognition system is proposed using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). In the proposed system, IAM dataset is used for training and testing. Totally 87,292 images are used for training and 4,316 images are used for testing. In the proposed system, five distinct features are extracted from the database. In this system, two different classifiers are used for classification namely CNN and RNN. The results obtained are shown in the paper. From the result, RNN performs better than CNN.

Keywords: Hybrid handwritten text recognition, IAM dataset, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

1. INTRODUCTION

Handwritten text recognition is the ability of a computer system to recognize handwritten text and convert it into digital text. There are two handwritten recognition systems existed namely, online and offline handwritten text recognition. Offline handwritten text recognition is the process of converting handwritten text which has been scanned or photographed from a physical document, such as a handwritten letter or a historical manuscript, into digital text. At present, offline text recognition systems are mostly designed to recognize machine-printed and handwritten texts. Recognizing the text written in various styles, it requires special consideration when dealing with handwritten documents.

The general methodologies of handwritten text recognition are image acquisition, preprocessing, segmentation, feature extraction, classification, and postprocessing. In the image acquisition stage, an image in specific format like .jpeg, .jpg or .png is sent to the system. Various datasets such as CEDAR, IAM, and Chars74k are used in the image acquisition stage. Pre-processing techniques are then applied to enhance the image quality, followed by image segmentation, where the image is split into various segments based on text recognition requirements. The feature extraction stage involves identifying one or more features that can help to improve the system's text recognition. Finally, the classification stage is used to identify the input text from the image.

The remaining part of the paper is organized as follows. In section 2, related works are discussed. Section 3 presents the proposed system of hybrid handwritten text recognition. Section 4 evaluates the results and discussions in detail. The paper is concluded in section 5.

2. RELATED WORKS

In this section, several notable literatures have been quoted, which has been conducted an extensive analysis on handwritten text recognition using various datasets, classifiers, and feature extraction techniques.

Gauri Katiyar et.al [1] 2016, has developed a handwritten character recognition system with seven different features are mean, gradient functions, edges, centre of gravity, standard deviation, diagonal and box features. Genetic Algorithm (GA) has been used for feature selection and the author used Multi-Layer Perceptron (MLP) to recognize the characters. By training the system using the CEDAR dataset, the author achieved an accuracy of 94% and 91% for capital letters and small letters, respectively.

Hasan mahmud et.al [2] 2021, has developed a system with an architecture named as Dynamic Time Warping (DTW) and achieved an accuracy of 96.8%. The author created own dataset and extracted depth-based features to recognize and gain accuracy.

R. Geetha et.al, [3] 2021 has put forth a system with a CNN-RNN networks and achieved the accuracy of 98.7% using RIMES dataset. The author used a sequence-sequence (Seq2Seq) approach to gain the accuracy of 96.3% in IAM dataset.

Ragunath Dey et.al, [4] 2021 has trained the system using Chars74k dataset and extracted a sliding window feature. The feature is classified with Edit Distance (ED) classifier and achieved an accuracy of 90.84%.

Shrinivas R Zanwaret.al, [5] 2021 has proposed a system by extracting ICA (Independent Component Analysis) Feature and used a hybrid algorithm by combining Firefly Algorithm (FFA), Particle Swarm Optimization (PSO) and achieved accuracy rate of 98.3%. This system uses the Back Propagation Neural Network (BPNN) as classifier and dataset from the Modified National Institute of Standards and Technology (MNIST).

Manoj Kumar Sharma et.al, [6] 2015 developed a system using Feedforward Neural Network (FFNN) and used a unique feature called Pixel Plot Trace and Re-plot and Re-trace (PPTRPRT) which was extracted and achieved an accuracy of 97.3%. This model was trained using ICDAR2005 and CMATER dataset.

E Kavallieratou et.al, [7] 2002 has come up with many segmentation techniques like skew angle, text discrimination, slant removal, line, word and character segmentation to enhance the handwritten recognition. This system proposed NIST database to train by k-means algorithm to extract features like histogram and profile. This system gained accuracy that varies from 65.6% to 100%.

3. PROPOSED SYSTEM

The proposed system also consists of five stages and the general schematic representation diagram is shown in Fig.1. The system aims in reducing the computational time and extracting more features in improvising the recognition accuracy. The proposed system will offer a significant improvement in performance of text recognition, with potential applications in a wide range of fields, including document analysis, automated transcription, and image-to-text conversion.

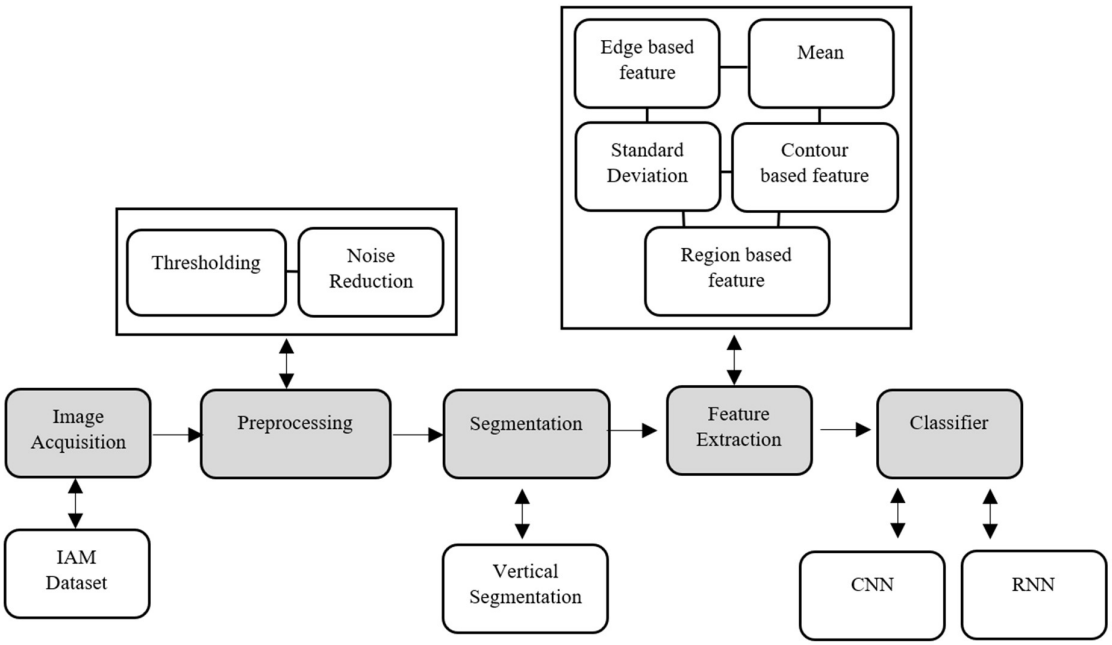


Fig.1 Schematic representation of the Proposed model

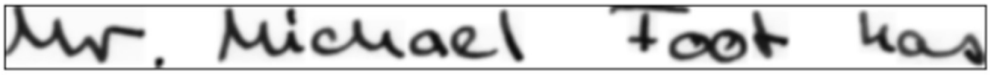
3.1 Image Acquisition

The first stage in recognizing and identifying text is image acquisition, which usually involves obtaining an image from its source. In the proposed model, the text recognition is sent to the model as an input in .png format from the IAM dataset.

The IAM dataset is utilized in the image acquisition stage for the proposed system, and it comprises 1,15,320 words and 13,353 lines from 657 writers. Out of these, 87,292 images were used for training, and 4,316 images were used for testing purpose. A sample input image from the IAM dataset is shown in Fig.2.

Fig.2 Sample input image from the IAM dataset

3.2



Preprocessing

The second stage in the analysis of an image is image pre-processing, the proposed system follows two preprocessing techniques which are noise reduction and thresholding. Gaussian filter is used for noise reduction in the system which helps in smoothing out noise in an image with preserving edges and other important features. The proposed system also uses Otsu thresholding method that is used to automatically determine an optimal threshold value. This threshold is then used to convert the grayscale image into a binary image, where pixels with values above the threshold are set to one (foreground), and those below are set to zero (background). The image after preprocessing is shown in Fig.3.

Fig.3 Image after Thresholding and Noise reduction

3.3 Segmentation

Segmentation helps in breaking a word into smaller fragments of letters. This process helps the system analyse and recognize each character in a more accurate way. Vertical segmentation technique is used in the proposed System for segmenting the words into individual characters. In Vertical segmentation input image is analysed vertically to identify the white space between the text lines or columns. Before Segmenting the characters, Skeletonization is followed to reduce the complexity of the individual characters in the image. This process involves thinning the character strokes to a single-pixel width while preserving the topological structure and connectivity of the character. Then the process of dividing an image into multiple columns using vertical strips is done. The image after Segmentation is shown in Fig.4.

Original text

Skeletonized text

Segmented Characters

Fig.4 Image after Vertical Segmentation

3.4 Feature Extraction

The Feature extraction step plays a chief role in retrieving the most relevant data from the input scanned image. In the proposed model, the default kernel value is set at (3, 0.8, 3.5) for grayscale to extract a set of hybrid features.

3.4.1 Edge based Detection and Edge based Binary Fill

Edge-based detection is used to identify the edges of individual characters. The proposed system uses canny operator for Detecting the edges of the character. Detection of edges is shown in Fig.5. Edge-based binary fill is a technique used to fill in the gaps between the detected edges of characters. This method involves converting the image into a binary format and filling in the gaps between the edges using a binary fill algorithm. Edge detection and Binary Fill is shown in Fig.6.

Fig.5 Image after Detection of Edges in the image

Fig.6 Image after Edge detection and Binary Fill

3.4.2 Mean

The mean of an image is a statistical measure that represents the average intensity of the image pixels. The mean is calculated using (1) for the IAM dataset. Mean is used in combination with other features, such as standard deviation, to provide a more comprehensive representation of the image.

$$Mean = \frac{Sum\ of\ pixel\ value}{total\ no\ of\ pixels} \quad \rightarrow (1)$$

3.4.3 Standard deviation (SD)

Standard deviation is used in conjunction with Mean feature, to create a more robust and informative feature for text recognition. Standard deviation (SD) is a statistical measure used to represent the variation in intensity of an image. Equation (2) is used to find the Standard deviation of the data in the Proposed System.

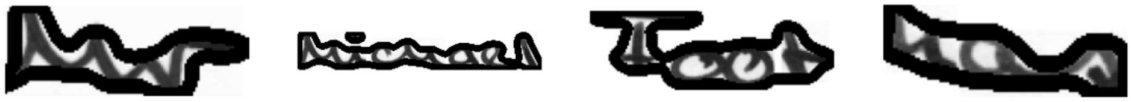
$$SD = \sqrt{(\left(\frac{1}{N}\right) * sum((x - mean)^2))} \rightarrow (2)$$

3.4.4 Region based feature

Region-based features can be used to capture a variety of characteristics of a specific region of interest, such as size, shape, texture, and stroke width. By focusing on specific regions, these features can be more targeted and provide a more accurate representation of the underlying text. In the proposed System, Zone Layout Descriptor (ZLD) is used, which captures the layout of a character or word by dividing the region of interest into sub-zones and calculating various statistical measures, such as mean and standard deviation, for each sub-zone. The image after Region based feature is used is shown in Fig.7.

Fig.7 Image after region based feature

3.4.5 Contour based feature



Contour-based features is used to capture a variety of characteristics of the shape of a specific region of interest, such as the number and position of corners, the length and curvature of the

contour, and the smoothness of the boundary. In the Proposed System, chain code type of Contour based feature is used which represents the boundary of a character or word as a sequence of directional codes. The Contour based feature is extracted and is shown in Fig.8.

Fig.8 Image with Contour features

3.5 Classification

The proposed system uses two algorithms for Classification, CNN and RNN. A four-layer CNN architecture is used in the proposed system which consists of input layer, convolution layer, max pooling layer and fully connected layer. Representing image width, height, length, batch size and maximum text length as parameters for CNN classifier. Rectified Linear Unit (RELU) acts as activation function, in which helps CNN to learn complex patterns of data. First layer receives the input image in a size of 128x64. Second layer is used to extract features from input image to produce feature maps. Third layer reduces the spatial resolution and dimensionality of feature maps. Fully connected layer is used to extract features to make a prediction about the input image, based on the learned relationship between the features during the training stage.

The Layers of Convolutional Neural Network and the Configured Layers of Convolutional Neural Network is shown in Fig.9 and Fig.10 respectively.

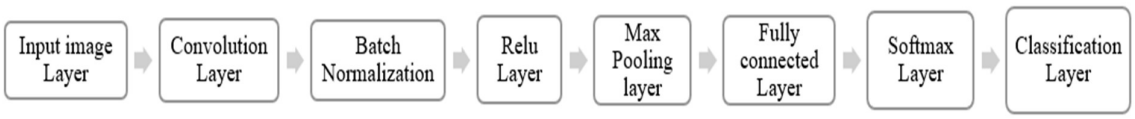


Fig. 9 Layers of Convolutional Neural Network

Convolution layer uses zero padding technique to avoid the problem of shrinking by adding a layer of zeros. In the proposed system, the input image is received with (128, 64, 1) size and with a filter size of 64. The stride size of kernel is (3,3) which specifies 3x3 matrix. The stride is used to control the size of the output feature maps produced by a convolutional layer. Stride determines the number of pixels by which the convolutional filter is shifted across the input image or feature map. Using a larger stride size, leads to reducing the spatial dimensions of the output feature map, resulting in fewer computations and faster processing.

The proposed System also uses RNN as another classifier and the Layers of Recurrent Neural Network is Shown in Fig 11. The input layer takes the input sequence and converts each character into a vector representation that can be processed by the network. This layer typically includes an embedding layer, which maps each character to a dense vector representation. The hidden layer consists of recurrent neurons and is implemented with Bi-Directional Long Short-Term Memory (Bi-LSTM) unit. The output layer takes the output of the hidden layer and generates a sequence of predicted characters in a text. This layer typically includes a fully connected layer with a SoftMax activation function, which recognizes the text.

Fig. 10 Configured Layers of Convolutional Neural Network

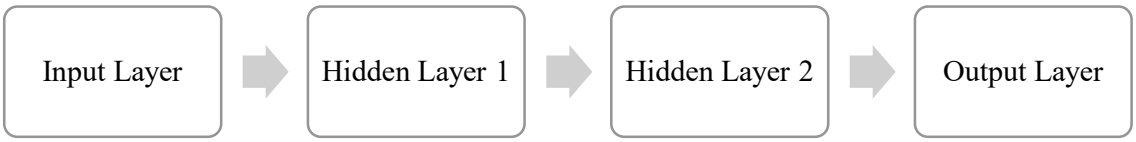


Fig. 11 Layers of Recurrent Neural Network

In the proposed system, the input image is received by input layer. The Text vectorization is a process of converting textual data into numerical vectors that can be processed by machine learning algorithms. Once the text has been vectorized, it can be fed into the RNN model as input. The Embedding layer is used to transform input data, which is typically represented as a sequence of discrete symbols such as words or characters, into a continuous vector space. The purpose of the embedding layer is to capture the semantic and contextual relationships between the symbols in the input sequence. The Bi-LSTM is capable of processing input sequences in both forward and backward directions. This means that the input sequence is processed from the beginning to the end and from the end to the beginning, and the outputs from both directions are combined to make a final prediction. Dense layer in the proposed system is used to generate the final output based on the Bi-LSTM. This allows RNN to generate output sequences such as predictions and texts are classified. The purpose of the SoftMax activation function in an RNN is to convert the output of the final dense layer into a probability distribution, which is useful for classification tasks. The Configured Layers of Recurrent Neural Network is shown in Fig. 12.

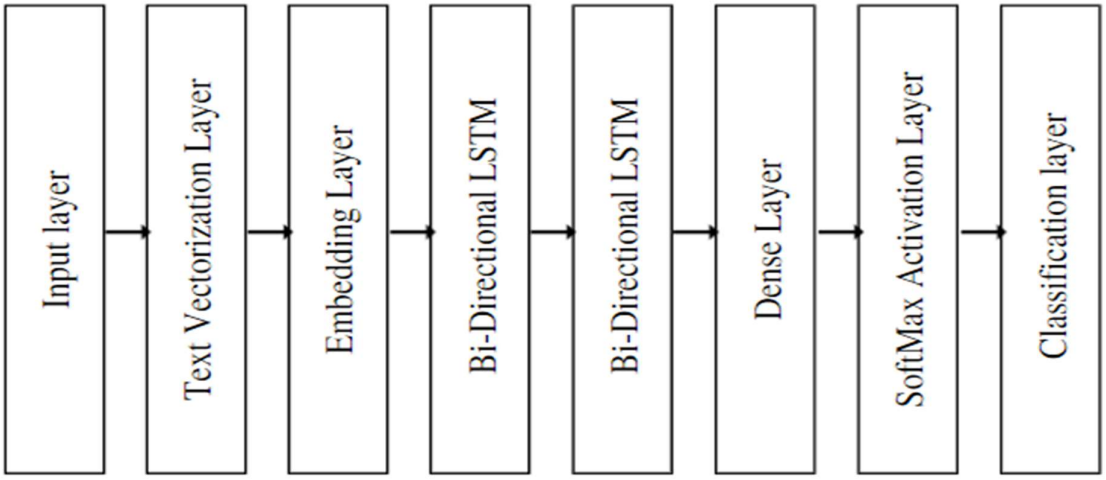


Fig. 12 Configured Layers of Recurrent Neural Network

4. Results and Discussion

In this proposed system, Python and Keras are used in building the proposed system and IAM dataset is used in training and testing the system. Totally 87,292 images are used for training and 4,316 images are used for testing. Five different features have been extracted in feature extraction stage and it is classified using two different neural networks namely CNN and RNN. Two various activation functions such as Relu and SoftMax are used in CNN and RNN respectively. The Comparison results of Architecture with accuracy and Comparison results of Features with accuracy are obtained and shown in Table 1 and Table 2 respectively.

Table 1 – Comparison of Architecture and Accuracy

Author	Architecture	Accuracy
Ragunathdey et.al	ED	90.45%
Gauri Katiyar et.al	MLP	94.65%
Geetha et.al	CNN-RNN	95.20%
Manoj kumarsharma et.al	FFNN	97.30%
Hasan muhmud et.al	DTW	96.84%
Proposed System	RNN	94.50%
	CNN	89.00%

Table 2 – Comparison of Features, dataset and Accuracy

Author	Features	Accuracy
Gauri Katiyar et.al	Hybrid features	92.5%
Geetha et.al	Seq-Seq	95%
Ragunath dey et.al	Sliding window	90.84%
Manoj kumar sharma et.al	PPTRPRT	97.3%
Hasan muhmud et.al	Depth	96.84%
Proposed System	Hybrid features	94.5%

The Training and Validation Loss of the RNN is shown in Fig.13 and Training and Validation Accuracy is shown in Fig.14. From the obtained results, the minimum loss is found at Epoch 30. At epoch level 30, the obtained loss is 0.0116. The maximum Accuracy of the system is found at Epoch 23. The Performance comparison of the system at different iterations are shown in Table 3.

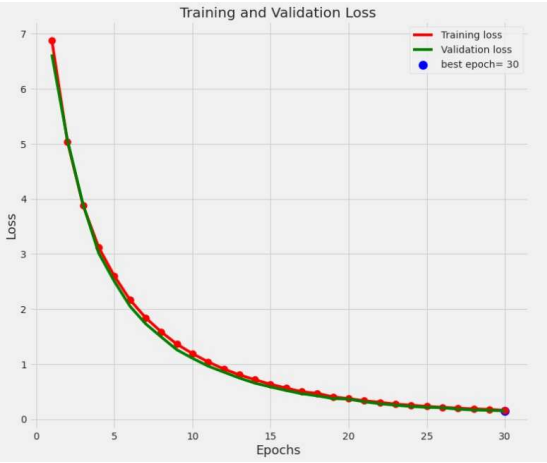


Fig.13 Training and Validation Loss

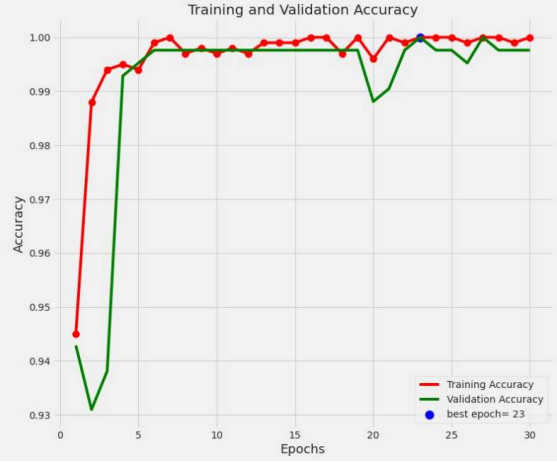


Fig.14 Training and Validation Accuracy

Table 3: Performance comparison at different iterations

Epoch	Accuracy	Loss	Val_Accuracy	Val_Loss
10	0.9946	0.0168	0.9321	0.2204
15	0.9918	0.0335	0.9042	0.3479
20	0.9891	0.0287	0.8709	0.4829

25	0.9918	0.0249	0.9069	0.3847
30	0.9980	0.0116	0.9171	0.3274

The Training and Validation Loss of the CNN is shown in Fig.15 and Training and Validation Accuracy is shown in Fig.16. From the obtained results, the minimum loss is found at Epoch 30. At epoch level 30, the obtained loss is 0.0545. The maximum Accuracy of the system is found at Epoch 24 and an average Accuracy rate is 89.00%. The Performance comparison of the system at different iterations are shown in Table 4.

Fig.15 Training and Validation Loss

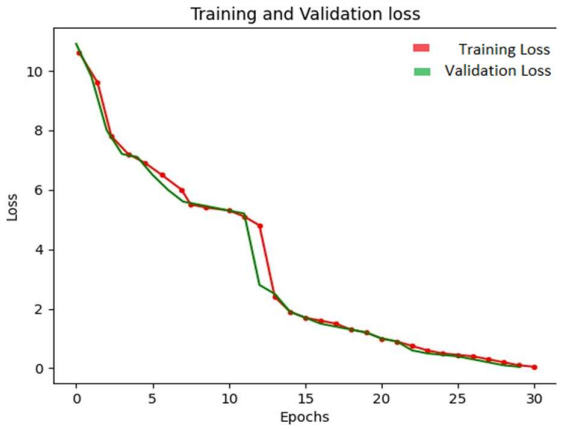


Fig.16 Training and Validation Accuracy

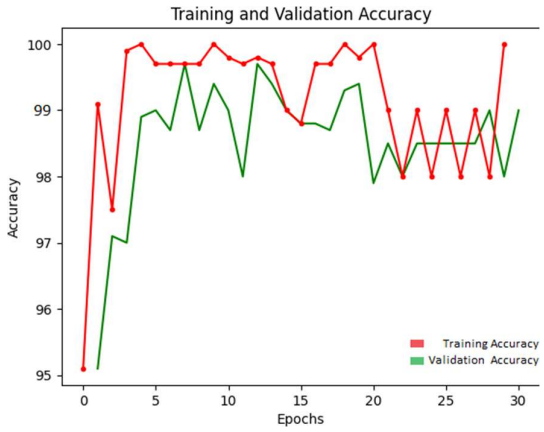


Table 4 Performance comparison at different iterations

Epoch	Accuracy	Loss	Val_Accuracy	Val_Loss
10	0.8989	0.2025	0.8045	0.3480
15	0.9413	0.0895	0.8607	0.5029
20	0.9283	0.0840	0.8466	0.3275
25	0.8911	0.1256	0,.7999	0.2204
30	0.9002	0.0545	0.8190	0.3845

As per the Experimental Results obtained, RNN gains an accuracy value of 94.50% while 89.00% for CNN.

5. Conclusion

The proposed system presents a Hybrid handwritten text recognition system by using CNN and RNN classifiers. There are five different approaches of features namely Edge Based Feature, Standard Deviation (SD), Region Based Feature, Contour based Feature have been used. The network is trained and tested on the IAM dataset. Two different Classification Algorithms are used and Experimental results are concluded that, by extracting the mentioned Features, RNN classifier Performs better than CNN as it provides better accuracy.

6. References

[1] Katiyar, G., Mehfuz, S. A hybrid recognition system for off-line handwritten characters. SpringerPlus 5, 357 (2016).

[2] Mahmud, H., Islam, R. & Hasan, M.K. On-air English Capital Alphabet (ECA) recognition using depth information. Vis Comput 38, 1015–1025 (2022).

[3] Geetha, R., Thilagam, T. &Padmavathy, T. Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks. Neural Comput&Applic 33, 10923–10934 (2021).
[4] Dey, R., Balabantaray, R.C. & Mohanty, S. Sliding window based off-line handwritten text recognition using edit distance. Multimed Tools Appl 81, 22761– 22788 (2022).
[5] Zanwar, S.R., Shinde, U.B., Narote, A.S. et al. Hybrid Optimization And Effectual Classification For High Recognitions In OCR Systems. J. Inst. Eng. India Ser. B 102, 969–977 (2021).
[6] Sharma, M.K., Dhaka, V.P. Segmentation of english Offline handwritten cursive scripts using a feedforward neural network. Neural Comput&Applic 27, 1369–1379 (2016).
[7] Kavallieratou, E., Fakotakis, N. & Kokkinakis, G. An unconstrained handwriting recognition system. IJDAR 4, 226–242 (2002).
[8] Sampath, A.K., Gomathi, N. Fuzzy-based multi-kernel spherical support vector machine for effective handwritten character recognition. Sādhanā 42, 1513– 1525 (2017).
[9] Yan, R., Peng, L., Xiao, S. et al. Dynamic temporal residual network for sequence modelling. IJDAR 22, 235–246 (2019).
[10] Hussain, R., Raza, A., Siddiqi, I. et al. A comprehensive survey of handwritten document benchmarks: structure, usage and evaluation. J Image Video Proc. 2015, 46 (2015).
[11] Nebti, S., Boukerram, A. Handwritten characters recognition based on natureinspired computing and neuro-evolution. ApplIntell 38, 146–159 (2013).
[12] Marti, UV., Bunke, H. The IAM-database: an English sentence database for offline handwriting recognition. IJDAR 5, 39–46 (2002).
[13] Morita, M., Sabourin, R., Bortolozzi, F. et al. Segmentation and recognition of handwritten dates: an HMM-MLP hybrid approach. IJDAR 6, 248–262 (2003).
[14] F. A. Kha, F. Khelifi, M. A. Tahir and A. Bouridane, "Dissimilarity Gaussian Mixture Models for Efficient Offline Handwritten Text-Independent Identification Using SIFT and RootSIFT Descriptors," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 2, pp. 289-303, Feb. (2019).
[15] M. Zimmermann, J. . -C. Chappelier and H. Bunke, "Offline grammar-based recognition of handwritten sentences," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 818-821, May (2006).
[16] Liu, CL., Sako, H. & Fujisawa, H. Performance evaluation of pattern classifiers for handwritten character recognition. IJDAR 4, 191–204 (2002).
[17] Gunter, S., Bunke, H. (2004). Ensembles of Classifiers for Handwritten Word Recognition Specialized on Individual Handwriting Style. In: Marinai, S., Dengel, A.R. (eds) Document Analysis Systems VI. DAS (2004).
[18] Koerich, A., Sabourin, R. & Suen, C. Large vocabulary off-line handwriting recognition: A survey. Patt. Analy. App. 6, 97–121 (2003).