

# CNN-RNN BASED HANDWRITTEN TEXT RECOGNITION

**G.R. Hemanth, M. Jayasree, S. Keerthi Venii, P. Akshaya, and R. Saranya**

*Department of Electrical and Electronics Engineering, PSG Institute of Technology and Applied Research, India*

## Abstract

*At present most of the scripts are handwritten due to the ease of using a pen tip in place of a keyboard, hence errors are common due to illegibility of the human handwriting. To avoid this problem handwriting recognition is essential. Offline handwritten Text recognition (OHTR) has become one of the major areas of research in recent times because of the need to eliminate errors due to misinterpretation of handwritten text and the need for automation to improve efficiency. The application of this system can be seen in fields like handwritten application interpretations, postal address recognition, signature verification, and various others. In this project, offline handwritten Text recognition is performed using Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) which uses the architecture of Recurrent Neural network (RNN) and Connectionist Temporal Classification (CTC). The neural network is trained and tested using the IAM database containing handwritten English text. The implementation of this work is done using image segmentation-based handwritten text recognition where OpenCV is used for performing image processing and TensorFlow is used for training and text recognition. This whole system is developed using python and the output is displayed in a word file.*

## Keywords:

*Offline Handwritten Text Recognition, Convolutional Neural Network, Recurrent Neural Network, Connectionist Temporal Classification, Long Short-Term Memory*

## 1. INTRODUCTION

Offline Text Recognition includes the science of recognizing both the human written font as well as the system-generated font. With the technology advancing, there arose an ardent need for combining conventional records into a digitized one, thereby, eliminating redundancies and diminishing the communication chain. With the world gravitating towards absolute digitization, there is a high demand for a Handwritten Text Recognition system. However, the challenge of implementing this system lies in the fact that handwritten words vary in characteristics such as slant and rounded letters, diacritic dots, crossbars, and humped letters. A good handwriting recognition system must accurately identify the distorted characters and hence find the most plausible words.

In this application, the IAM Dataset containing more than 100,000 images of unconstrained handwritten text is used for training, validating, and testing the system to achieve better efficiency. The Neural Network is trained using this dataset to eventually recognize handwriting. The recurrent neural network (RNN) can process larger input though it has lesser computational power. On the other hand, the Convolutional neural network (CNN) needs larger data for training. In this handwritten text recognition system, an adaptive method is proposed for offline HTR by integrating both. Here the dataset is trained consecutively with CNN and RNN. Connectionist temporal classification (CTC) network is fitted along with RNN through training to model the

probability of a label. In this system, the hand-written text of the user is taken in as images. These images are pre-processed, the purpose of this stage is to address the problems of data reduction, elimination of imperfections, and normalization and produce a set of data that is related to an image that is suitable for segmentation.

The implementation of this project is done in three steps namely, line segmentation, word segmentation, and text recognition. Line segmentation is an important part of handwritten text recognition where the initial stage involves converting the images into greyscale, then inverted binary images are obtained followed by dilation of those images, and finally, boundary boxes are drawn over each line. Word segmentation is done by the scale-space technique. This process segments text lines obtained as the output of line segmentation into separate words which is used by the recognition system to recognize the text.

The objective of our proposed research work is to analyse the complexities involved in recognizing handwritten text by using both CNN and RNN and to calculate the loss using the Connectionist temporal classification (CTC) network. The other objectives are to test the performance evaluation of the CNN-RNN method compared with the conventional ones and to arrive at a method that can save time, enable faster processing, and reduce the probability of errors. An improvement in the man-to-machine interactions in many applications occurs by utilizing the advanced automation processes involved in the handwritten text recognition systems.

In this paper, we develop a model that consists of CNN, RNN, and CTC layers. The input image is fed into the CNN layers. These layers are trained to extract relevant features from the image. Each layer consists of three operations. First, the convolution operation, secondly the non-linear RELU function is applied. Finally, a pooling layer summarizes image regions and outputs a downsized version of the input. The feature sequence contains 256 features per time-step. The RNN propagates relevant information through this sequence. The popular Long Short-Term Memory (LSTM) implementation RNN is used. Finally, the CTC is given the RNN output matrix to decode the output text.

The main contribution of the paper is given below:

- The author proposes a novel method by combining CNN and RNN networks to obtain the best results.
- The text paragraph images are processed into word images using OpenCV contour techniques and the word images are fed into the network model for recognition.
- The model is trained by using the IAM word image dataset.

The overall paper has been organized as follows: section 2 deals with the related works. Section 3 discusses the proposed methodology. Section 4 describes the results, and section 6 deals with the conclusion.

## 2. LITERATURE REVIEW

There are many methods to perform handwritten text recognition but due to the non-triviality in the hand-written text, each method has its disadvantages. For this reason, the method with simple methodology and utmost efficiency is used for implementation.

Shivakumara *et al.* [1] aim to recognize the license plate of the vehicles in Malaysia with a white background and dark background. To overcome this challenge, they use CNN-RNN based recognition system for feature extraction and BLSTM to extract context pieces of information (classification). Datasets used in their work are MIMO and UCSD. All the existing methods for license plate recognition are not effective for multiple adverse factors, so classification is a crucial step in this system. LSTM is used for the recognition of license plates. Therefore, their paper proves that classification is useful for improving recognition performance.

Sueiras *et al.* [2] use the method of combining deep neural networks with sequence to sequence networks, also called an encoder-decoder. The proposed architecture aims to identify characters and conceptualize them with their neighbours to recognize a given word. For training and testing IAM and RIMES, these datasets consist of handwritten texts on white background from many people. The error rate in the test set is 12.7% in IAM and 6.6% in RIMES. This method is more efficient in the case of language translation and speech to text conversion rather than handwriting recognition.

Sampath and Gomathi [3] used Firefly and Levenbreg-Marquardt (FLM) algorithm for optical character recognition is used where Firefly and Levenbreg-Marquardt algorithms are combined to form the hybrid neural network algorithm. This hybrid neural network algorithm combines the advantages of both the algorithm and increases the speed and accuracy of the system. FLM with feed-forward is compared with SVM-based technique to prove the efficiency of the hybrid algorithm as in gradient feature descriptors. The disadvantage of their system is that the architecture is more complicated to perform a simple operation.

As Vo *et al.* [4] discuss line detection in handwritten documents has been a serious problem for processing scanned documents. Most of the existing approaches apply the heuristic rules or hand-designed features for the estimation of the location of text lines. A novel approach has been proposed that first trains a fully convolutional network (FCN) to predict the structure of text lines in the scanned images of the document. The line adjacency graph has been used for the separation of the touching characters and assigning them to the different text lines to ensure the completion of the segmentation process. The robustness of the system has been validated using the data of ICDAR2013 Handwritten Segmentation which has shown high performance for the combination of different languages and multi-skewed text lines.

## 3. PROPOSED METHODOLOGY

The outline of the proposed work is represented using the block diagram as shown in Fig.1. The first step in the process is training the dataset. The dataset is trained with CNN and RNN layers. The obtained output and the ground truth text are passed

through the CTC layer to get the trained model. The obtained trained model is then used to recognize the text in the input image.

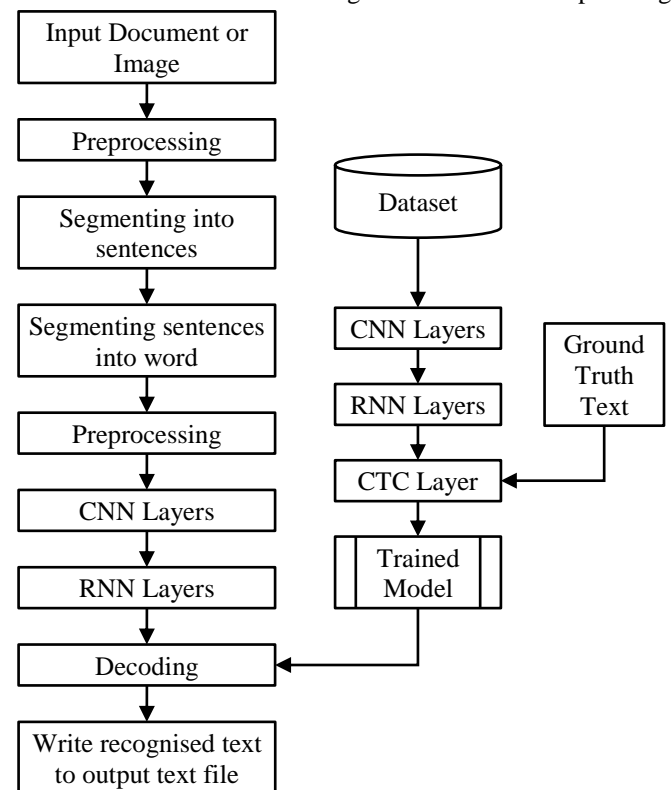


Fig.1. Block Diagram

The input handwritten image is pre-processed by adjusting the resolution. The first step in recognition is to break down the paragraph image into line images. Then line images are further segmented into word images. The word images are then pre-processed and passed through the same CNN and RNN layers that were used in training. The output of the RNN layers is given to the CTC layer (decoding level) to decode the output text with the help of the trained model.

The method of extracting word images from a paragraph and combining CNN, RNN, and CTC techniques to train the NN model has been effectively implemented. As a whole, an end-to-end handwritten text recognition system has been proposed and implemented.

### 3.1 CLASSIFIERS

#### 3.1.1 Convolutional Neural Network:

One of the most important algorithms used in deep learning is the convolutional neural network (CNN). It takes an image as the input, extracts feature from the image, and then distinguishes one feature from the other. It is inspired by the connections of neurons present in the human brain. The training data used in CNN itself drives the learning of the feature detection layer [7].

#### 3.1.2 Recurrent Neural Network:

Another most widely used algorithm in deep learning for the sequential processing of data in applications such as speech and text recognition are the recurrent neural network (RNN). The term recurrent indicates that the network repeatedly performs the same task for every element present in the data sequence and the

corresponding output produced depends on the values of its previous outputs [6].

### 3.1.3 Connectionist Temporal Classification:

Connectionist Temporal Classification (CTC) uses a new differentiable cost function that directly trains RNNs for the identification and labelling of the unsegmented sequences. To use this cost function, an additional blank symbol is introduced in the possible labels that the recurrent neural networks can output. The output layer of the RNN corresponds to probabilities over all possible labels.

## 3.2 PROPOSED CNN-RNN MODEL

The input image is given to the CNN layers. These CNN layers are trained to extract the necessary features from the input image. There are three operations performed in each layer, namely, convolution, activation, and downsized image generation. First, the convolution operation applies a  $5 \times 5$  size filter kernel for the first two layers and a  $3 \times 3$  filter kernel for the last three layers of the input image. Then the activation operation is done using a non-linear Rectified Linear Unit (ReLU). Finally, a downsized version of the input image is generated by a pooling layer that identifies the different regions of the image. In each layer, the image height is downsized by 2 and the channels are added by feature mapping to produce a  $32 \times 256$  output sequence.

Each time step consists of a feature sequence with 256 features that are applied to the RNN. The RNN is implemented using a Long Short-Term Memory (LSTM) network since it can transfer the data through a longer range and has more superior training characteristics compared to Vanilla RNN. The output sequence produced by RNN is mapped to a  $32 \times 80$  matrix. There are 79 different characters present in the IAM dataset along with an additional character required for creating CTC blank labels in the CTC operation. Thus, each time step has 80 elements.

While training the Neural Network (NN), the ground truth text and the RNN output matrix is fed to the CTC layer which first decodes the output matrix into a text, then compares the converted text and the ground truth text, and produces the loss value. The length of the recognized text is 32 characters. The average of the loss values is then used to train the NN. Then it is given to an RMSProp optimizer [6]. The trained model is then used to recognize the input image. The word error rate for the trained model is obtained as 10.62%.

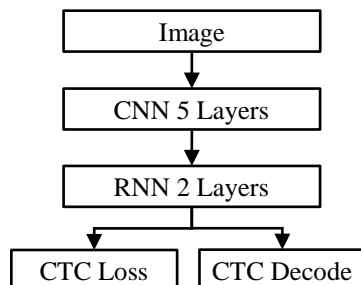


Fig.2. Overview of the model

## 3.3 IMPLEMENTATION

In this system, an adaptive method for offline HTR is proposed by integrating both RNN and CNN. The IAM dataset

containing approximately 100,000 British English words is trained consecutively with both neural networks.

### 3.3.1 Pre-processing

The pre-processing of handwriting involves a series of operations to be performed on the handwritten text data before the application of the necessary recognition algorithms. This stage addresses the problems of dimensionality reduction, removal of inconsistencies, and data normalization. It produces a set of data that is more suited for the segmentation of data present in the image format. Implementation of pre-processing is done by two major steps namely - Line segmentation [5], Word segmentation [5] using the OpenCV library.

The line segmentation involves pre-processing the input image by converting it into grayscale as shown in Fig.3. The obtained grayscale image is converted into an inverse binary image as shown in Fig.4, then the inverted binary image is dilated as shown in Fig.5. The contours of the binary image are found then applied with bounding boxes as shown in Fig.6 and are saved as separate line images as shown in Fig.7.

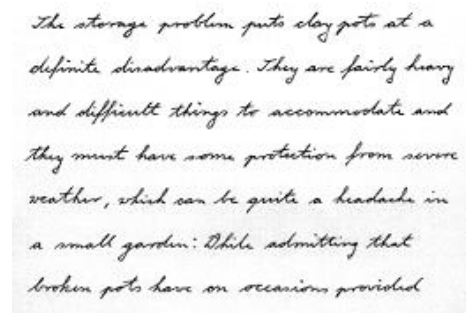


Fig.3. Gray scale image

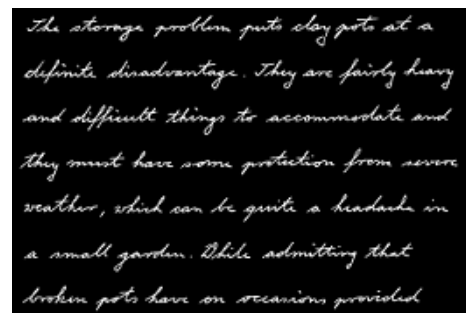


Fig.4. Inverted binary image



Fig.5. Dilated image

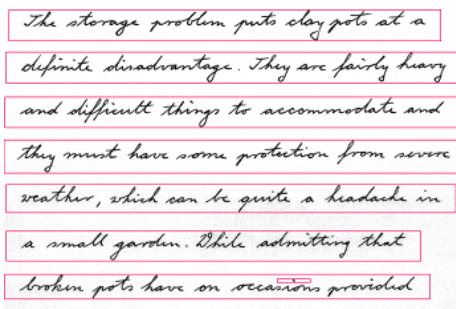


Fig.6. Bounding boxes over lines



Fig.7. Line images



Fig.8. Filtered line image

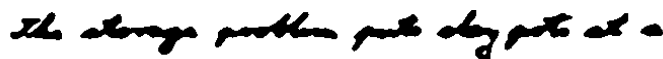


Fig.9. Dilated line image



Fig.10. Inverted line image

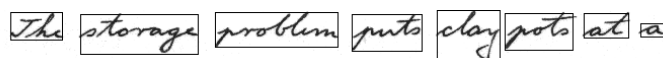


Fig.11. Bounding boxes over words

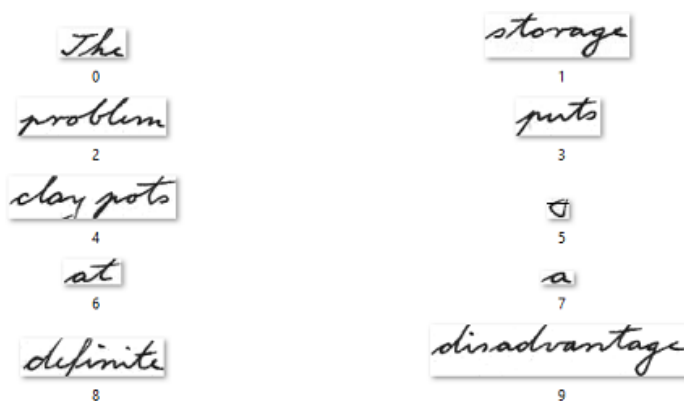


Fig.12. Word images

In the word segmentation process, the text lines obtained as the output of line segmentation are segmented into separate words. The line image is pre-processed and then the filter kernel is applied as shown in Fig.8. The filtered image is then dilated as shown in Fig.9 to obtain an inverted binary image as shown in Fig.10. The contours of the image are found and then applied with bounding boxes as shown in Fig.11. The words are then stored as separate word images as illustrated in Fig.12.

### 3.3.2 Dataset

The IAM dataset has been used for training the model. The IAM database consists of handwritten English sentences. It is based on the Lancaster-Oslo/Bergen (LOB) corpus. The database serves as a basis for a variety of recognition tasks, particularly useful in recognition tasks where linguistic knowledge beyond the lexicon level is used, as this knowledge is automatically derived from the underlying corpus. The IAM database also includes a few image-processing procedures for extracting the handwritten text from the forms and the segmentation of the text into lines and words. The training of the model is done using the IAM dataset along with the IAM dataset, custom handwritten paragraph images collected from random people were used for testing. These custom images were captured under normal lighting with a 5MP camera with a resolution of range 800 to 1000 dpi.

Table.1. Specification of IAM dataset

Description	Count
Number of writers contributed samples of their handwriting	657
Number of pages of scanned text	1539
Number of isolated and labelled sentences	5685
Number of isolated and labelled text lines	13353
Number of isolated and labelled words	115320

Table.2. Average probability for two sets of Training-Validation ratio

Image	50%-50%	65%-35%
1	74	72
2	54	45
3	70	68
4	55	57
5	80	75
6	63	66
7	49	51
8	46	45
9	70	68
10	72	70
11	78	75
12	76	75
13	71	71
14	79	75
15	71	69
<b>Average</b>	<b>67.2</b>	<b>65.47</b>

### 3.3.3 Training, Validation, and Testing

The dataset is split for training and testing with a ratio of 80:20. The training of the model is done using 80% of the IAM word image dataset, where it is further split into two for training and validation. The model is trained with two different Training-Validation ratios and further tested with the remaining 20% dataset. The handwritten paragraph images from the IAM dataset and handwritten custom images are taken randomly, pre-processed as discussed in Section.4.4.1, and tested for the Training-Validation ratios 50:50 and 65:35 to obtain the best word recognition probability. The average probabilities for the images are compared as shown in Table.2. It is found that the average probability for the Training-Validation ratio 50:50 is greater.

The aspect of boundary setting between training and testing datasets is crucial in determining the overall accuracy. More training data will not result in high accuracy. There lies a fine line in selecting signals and noise. Overfitting results when the training exceeds a limit and the system learns noise as a signal. This overfit model will then make predictions based on that noise. It will perform unusually well on its training data yet very poorly testing data.

The course of action commonly used to overcome the above disadvantage of overfitting is to split the dataset into 2 – Train and Test. The Test set is set aside and randomly chosen  $x\%$  of the Train dataset to be the actual Train set and the remaining  $(100-x)\%$  to be the Validation set, where  $x$  is a fixed number, the model is then iteratively trained and validated on these different sets. One common method used is Cross-Validation. In this, the training set is used to generate multiple splits of the Train and Validation sets. Cross-validation allows tuning hyperparameters with only the original training set and avoids overfitting. Thus cross-validating the training and validation set ratios, the efficient training – validation ratio is 50:50.

### 3.3.4 Recognition

The trained neural network model is then used for text recognition in the input image. The input images are first applied with the CNN layers. Each CNN layer generates an output of a 32-character sequence. There are 256 features present in each entry that is next processed by the RNN layers. The output of the RNN layers is given to the CTC to decode the output text.

## 3.4 HARDWARE AND SOFTWARE REQUIREMENT

The proposed work is done on a machine running on Windows 10 OS with 8 GB RAM and a 2.40 GHz processor. Python is a simple programming language with features like stability and flexibility for Artificial Intelligence projects. NumPy package in Python is used for scientific computing and data analysis which makes it efficient to store and manipulate data compared with built-in Python data structures.

OpenCV is an open-source, cross-platform library that facilitates computer vision applications. It is written in optimized C++ to provide high-level interfaces to capture, process, and present the image data. OpenCV's ability to bind with Python helps in providing data in a standardized format, which is compatible with scientific libraries like NumPy. Google's TensorFlow is an open-source library used for large-scale machine learning and complex numerical computations. The

advantage of TensorFlow is that it provides a level of abstraction in machine learning development that allows the developer to focus on the overall logic of the application.

## 4. RESULTS AND DISCUSSION

The offline handwritten character recognition system is implemented in an effective way to improve its working in each step of the process right from pre-processing, training, and finally recognition.

### 4.1 PRE-PROCESSING

In the pre-processing stage, the paragraphs are first converted to lines and finally to words. During this process, the image is converted to grayscale, and words are extracted. It is done effectively using OpenCV. This pre-processing stage helps in more efficient text recognition.

### 4.2 TRAINING AND VALIDATION

IAM dataset having more than 100,000 handwritten word images is used for training and validation which improved the algorithm efficiency. The training and validation of the model are done with ratios of 65:35 and 50:50 and it is inferred that the average probability for ratio 50:50 is greater. The word error rate for the trained model is obtained as 10.62%.

### 4.3 TEXT RECOGNITION

While evaluating the performance of the proposed system, the probability of each recognized word is found to be in the range 50%-98% in the IAM dataset image and 45%-90% in the custom handwritten image. The probability of the recognized paragraph is found to be in the range 70%-80% in the IAM dataset image and for custom images probability is in the range 50%-70% as shown in Table.3.

Table.3. Probability results of word and paragraph images

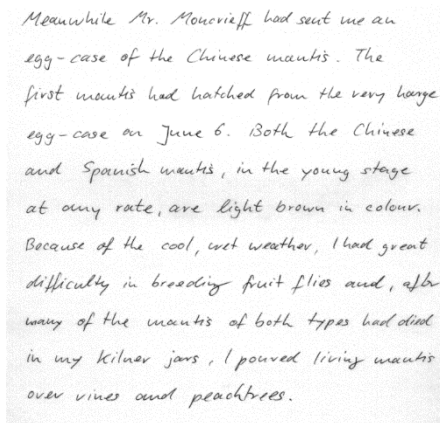
Probability	IAM Dataset Image	Custom Handwritten Image
Average Probability of all words in a paragraph	70% to 80%	50% to 70%
Individual probability of word image in a paragraph	50% to 98%	45% to 90%

### 4.4 OUTPUT

The input paragraph images from the IAM dataset and custom handwritten images are processed by the various steps as discussed in the model and finally, output text is obtained as a text file with an average probability of all words in the paragraph. The input image Fig.13 is taken from the IAM dataset and its recognized output with the total probability is shown in Fig.14.

The method in which images are generated is an essential factor in determining the accuracy as it affects the quality of input images to be fed to the system. Images produced by cameras are not as good compared with scanners. The issues affecting the quality of the image taken by the camera may be natural or device-oriented. Few factors are inadequate lighting, shadows, glares,

blurring, degradation in corners, skewness, tilting, aspect ratio, etc.



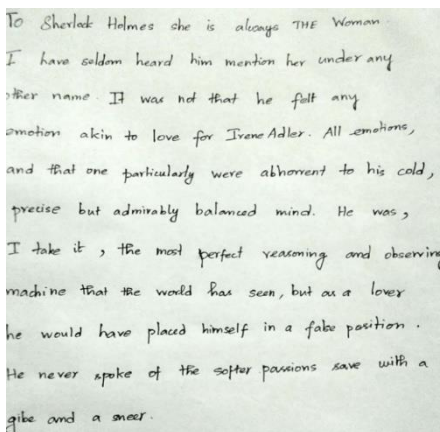
Meanwhile Mr. Moncrieff had sent me an egg-case of the Chinese mantis. The first mantis had hatched from the very large egg-case on June 6. Both the Chinese and Spanish mantis, in the young stage at any rate, are light brown in colour. Because of the cool, wet weather, I had great difficulty in breeding fruit flies and, after many of the mantis of both types had died in my Kilner jars, I poured living mantis over vines and peachtrees.

Fig.13. Input image (IAM dataset)



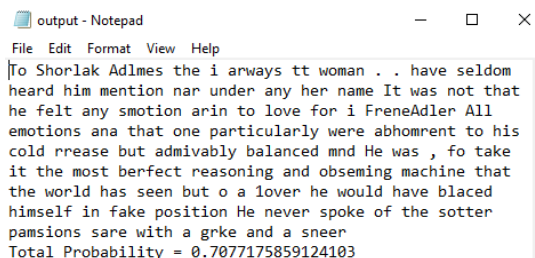
output - Notepad  
File Edit Format View Help  
Meanwhile Mr Moncrieff had sent me an 19y case of the Chinese mantis The hrst montis had hatched from the very harge egg- case on June , Both the Chinese and Spanish mantes in the young stage at ay rate are light brown in colour Because of the cool wet weather , had great difficulty in breeding frnit flies and after many of the mantis of both types had ldied in my Kilner yars poured living mantis over vies and peachtrees  
Total Probability = 0.8017153964366441

Fig.14. System Output



To Sherlock Holmes she is always THE Woman  
I have seldom heard him mention her under any other name. It was not that he felt any emotion akin to love for Irene Adler. All emotions, and that one particularly were abhorrent to his cold, precise but admirably balanced mind. He was, I take it, the most perfect reasoning and observing machine that the world has seen, but as a lover he would have placed himself in a false position. He never spoke of the softer passions save with a gibe and a sneer.

Fig.15. Input image (Custom)



output - Notepad  
File Edit Format View Help  
To Shorlak Adlmes the i arways tt woman . . have seldom heard him mention nar under any her name It was not that he felt any smotion arin to love for i FreneAdler All emotions ana that one particularly were abhormrent to his cold rrease but admivably balanced mnd He was , fo take it the most perfect reasoning and obseming machine that the world has seen but o a lover he would have blaced himself in fake position He never spoke of the sotter pamsions sare with a grke and a sneer  
Total Probability = 0.7077175859124103

Fig.16. System Output

A custom handwritten text image with 917 dpi resolution captured under normal home light setting as shown in Fig.15 was fed as input to the system. The IAM dataset paragraph input image

which was scanned using a scanner has a higher rate of word probability recognition as compared with the custom image taken with a 5MP camera. The corresponding recognized output with the total probability is shown in Fig.16.

Table.3. Comparison of the proposed model with conventional and other deep learning models

Title	Dataset	Method	Accuracy
[1]	MIMO and UCSD	CNN, BLSTM, LSTM	86.37-90.5%
[2]	IAM, RIMES	CNN, LSTM	95%
[3]	Chars74k	FLM	92-95%
[7]	Self-built Shui character	CNN	93.3%
Proposed Model	IAM and Custom Handwritten	CNN, RNN, CTC	98%

The proposed work has an accuracy of 98% for word recognition. When comparing the results with [1]-[3] and [7], the word recognition accuracy of our proposed work was found to be the highest. As [8] discusses, the HTR-FLOR model has the least word error rate of 10.92% for the IAM dataset. Whereas, our model has a word error rate of 10.62% which is comparatively less. The comparison of our work with the other models has been illustrated in Table.3.

## 5. CONCLUSION AND FUTURE WORK

In this system, an adaptive method is proposed for offline paragraph recognition by pre-processing and training the dataset consecutively with CNN and RNN. The input paragraph images are first pre-processed by using OpenCV contour techniques and are split into line images and further line images are processed into word images which are fed into the NN model layers during recognition. The output of the CNN layers is further processed by the RNN layers. The output of the RNN layers is given to the CTC to decode the output text. The results demonstrate the potential of consecutive use of CNN and RNN that improve the accuracy steadily.

In future work, we intend to improve the work by making use of hybrid datasets and experimenting with different activation functions, also increasing the number of neural network layers. Further, we aim to enhance the work by implementing online recognition and extend it to different languages, additionally we can promote the system to recognize degraded text or broken characters.

## REFERENCES

- [1] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal and M. Hossein Anisi, "CNN-RNN based Method for License Plate Recognition", *CAAI Transactions on Intelligence Technology*, Vol. 3, No. 3, pp. 169-175, 2018.
- [2] J. Sueiras, V. Ruiz, A. Sanchez and J.F. Velez, "Offline Continuous Handwriting Recognition using Sequence to Sequence Neural Networks", *Neurocomputing*, Vol. 289, pp. 119-128, 2018.

- [3] A. Sampath and N. Gomathi, "Handwritten Optical Character Recognition by Hybrid Neural Network Training Algorithm", *The Imaging Science Journal*, Vol. 67, No. 7, pp. 359-373, 2019.
- [4] Q. Vo, S. Kim, H. Yang and G. Lee, "Text Line Segmentation using a Fully Convolutional Network in Handwritten Document Images", *IET Image Processing*, Vol. 12, No. 3, pp. 438-446, 2018.
- [5] J. Chung and T. Delteil, "A Computationally Efficient Pipeline Approach to Full Page Offline Handwritten Text Recognition", *Proceedings of International Conference on Document Analysis and Recognition Workshops*, pp. 1-13, 2019.
- [6] Y. Chherawala, P. Roy and M. Cheriet, "Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Network Model", *IEEE Transactions on Cybernetics*, Vol. 46, No. 12, pp. 2825-2836, 2016.
- [7] Y. Weng and C. Xia, "A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices", *Mobile Networks and Applications*, Vol. 25, pp. 1-22, 2019.
- [8] A. De Sousa Neto, B. Bezerra, A. Toselli and E. Lima, "HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition", *Proceedings of International Conference on Graphics, Patterns and Images*, pp. 1-8, 2020.