Handwritten Text Recognition using Deep Learning for Automated Paper Checking

Prof. Anisha M Lal
Associate Professor Grade
School of Computer Science (SCOPE)
Vellore Institute of Technology
Vellore, India
anishamlal@vit.ac.in

Prakhar Dungarwal
Student Dept. School of
Computer Science (SCOPE)
Vellore Institute of Technology
Vellore, India
prakhar.dungarwal2018@vitstudent.ac.in
(18BCE2243)

Madhav Narayan Singh
Student Dept. School of
Computer Science (SCOPE)
Vellore Institute of Technology
Vellore, India
madhav.narayansingh2018@vitstudent.ac.in (18BCE2236)

Ritiwik Sharan
Student Dept. School of
Computer Science (SCOPE)
Vellore Institute of Technology
Vellore, India
ritwik.sharan2018@vitstudent.ac.in
(18BCE2232)

Abstract -

Amidst these treacherous times of the COVID-19 pandemic everyone wants to develop an fully-automated online examination system. Due to increasing number of courses and appearing students many hours of examiner and a lot of efforts are required for effective evaluation. Computer and technologies can be used to solve such complex problem.

The goal is to evaluate and assign scores to descriptive answer which are comparable to those of human assigned score by coupling deep learning technologies and Image Processing Techniques. Our proposed system includes an algorithm that is capable of evaluating an answer script based on your own handwriting and comparing it with the initially entered answer keywords both key word wise and also comparing each of these keywords.

Objective of our project is to reduce the time for manual paper checking by our proposed HTR Model.We in this paper have implemented Convolutional Recurrent Neural Networks (CRNN) which are better than tradition Hidden Markov Model (HMM) both for performance efficiency and better text recognition.

So the input given to the system is keywords of the answer, processing in the system is *through Handwritten Text Recognition (HTR) models and image processing techniques output are* the percentage of marks out of total marks to be awarded to the system. So it benefits both the teacher and student, teachers save their paper checking time as there is no need to check the whole answer they can award marks based on percentage evaluated by the system and student benefits as even if his answers do not contain the exact keywords the Deep Learning and Image Processing Systems are intelligent enough to give marks based on matching keywords and gets a fair amount of marks.

Keywords -

Handwitten Text Recognition (HTR), fully-automated, Convolutional Recurrent Nueral Networks (CRNN), Hidden Markov Model (HMM), Deep Learning

I. Introduction

Handwriting recognition has been one of the most fascinating and challenging research areas in field of image processing and pattern recognition in the recent years. It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications. Several research works have been focusing on new techniques and methods that would reduce the processing time while providing higher recognition accuracy.

The first important step in any handwritten recognition system is pre-processing followed by segmentation and feature extraction. Pre-processing includes the steps that are required to shape the input image into a form suitable for segmentation. In the segmentation, the input image is segmented into individual characters and then, each character is resized into m x n pixels towards the training network.

The manual system for evaluation of Subjective Answers for technical subjects involves a lot of time and effort of the evaluator. Subjective answers have various parameters upon which they can be evaluated such as the question specific content and writing style. Evaluating subjective answers is a critical task to Perform. When human being evaluates anything, the quality of evaluation may vary along with the emotions of the person. Performing evaluation through computers using intelligent techniques ensures uniformity in marking as the same inference mechanism is used for all the students. In Machine Learning, all result is only based on the input data provided by the user.

- Our Proposed System uses Deep learning and Image Processing to solve this problem.

- Our Algorithm performs a task like Tokenizing words and sentences, Part of Speech tagging, Chunking, chinking, Lemmatizing words and Wordnetting to evaluate the subjective answer.

- Along with it, our proposed algorithm provides the semantic meaning of the context.

Our System is divided into **two modules** :

1) Extracting the data from the scanned images and organizing it in the proper manner and
The software will take a scanned copy of the answer as an input and then after the preprocessing step, it will extract the test of the answer.

2) This text will again go through processing to build a model of keywords and feature sets. Model answer sets and keywords categorized as mentioned will be the input as well. The classifier will then, based on the training will give marks to the answers. Marks to the answer will be the final output.

The need for online examination aroused mainly to overcome the drawbacks of the existing system as well as the global COVID-19 crisis. The main aim of the project is to ensure user-friendly and more interactive software to the user. The online evaluation is a much faster and clear method to define all the relevant marking schemes. It brings much transparency to the present method of

answer checking The answers to all the questions after the extraction would be stored in a database. The database is designed as such that it is very easily accessible. Automating repetitive tasks has been the main aim of the industrial and technological revolution.

The paper is organized as follows, in Section II we go through the related work and compare our model with other models. In Section III, we look at the CRNN and deep learning techniques combined with image processing techniques to perform Handwritten Text recognition. Section IV Describes what overall framework are we implementing and architectural diagrams for the same.Section V presents the experimental results such as the predicted and the ground truth , probability of matching, no. of matched keywords in case of our own handwritten text. Section VI throws light on the performance evaluations on our technique and compares it to other existing techniques, the paper is concluded in Section VII.

II. **Literature Survey/ Related Work :**

[1] PAQUET, M. T. (2017). Structuring of Hidden Information in Markov Modeling with Application to Handwriting Recognition (Doctoral dissertation, telecom-paristech).

In the paper, the author presents the different pre-processing approaches coping with major types of variabilities in handwriting, namely: size, skew, slant, guidelines and noise. Some approaches were based on handcrafted heuristics while others were based on learned methods via neural networks. The latter ones could achieve better performance, but they require manual labeling of the training data. They have demonstrated that guidelines have large effect on the performance and proposed approaches to detect and remove them. This has led to a close performance for images with and without guidelines, which means that the effect of the guidelines was successfully reduced.At the end, we showed the improvement brought by all the preprocessing techniques adopted on the performance of our baseline systems, HMM and BLSTM.

[2] Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C. V. (2018, August). Improving cnn-rnn hybrid networks for handwriting recognition. In 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 80-85). IEEE.

In this paper, the author presented effective ways to train a CNNRNN hybrid architecture using synthetic data and domain specific image normalization and augmentation. He has also showed the individual contributions of each of these modules for improving the recognition rates at both line and word levels. In future, he would also like to integrate our line-level recognition model with language model based decoding so as to further enhance our recognition performance.

[3] Balci, B., Saadati, D., & Shiferaw, D. (2017). Handwritten text recognition using deep learning. CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring, 752-759.

In this paper, the proposed work employed a deep network model to recognize the handwritten texts. Preprocessing of dataset is also considered as a major factor behind the accuracy of the models. Training of network requires a supportive hardware to implement large dataset efficiently. It is necessary to train the network with large amount of dataset so that the network can easily recognize the character. Hence there is a requirement of high memory and high processing speed to achieve efficient network.the model consists of 5 layers of Convolutional Neural Network (CNN), 2 layers of Recurrent Neural Network (RNN) and outputs a character-probability matrix. This matrix is either used for CTC loss calculation or for CTC decoding.

[4]Dongare, S. A., Kshirsagar, D. B., & Waghchaure, M. S. V. (2014). Handwritten devanagari character recognition using neural network. IOSR J Comput Eng (IOSR-JCE) Ume, 16(2), 74-79.

This paper deals with development of grid based method which is combination of image centroid zone and zone centroid zone of individual character or numerical image. In feature extraction using grid or zone based approach individual character or numerical image is divided into n equal sized grids or zones then average distance of all pixels with respect to image centroid or grid centroid is computed. In combination of image centroid and zone centroid approach it computes average distance of all pixels present in each grid with respect to image centroid as well as zone centroid which gives feature vector of size 2xn features. This feature vector is presented to feed forward neural network for recognition. Complete process of Devanagari character recognition works in stages as document preprocessing, segmentation, feature extraction using grid based approach followed by recognition using feed forward neural network.

[5] Miroslav NOHAJ, Rudolf JAKA, "Image preprocessing for optical character recognition using neural networks"Journal of Patter Recognition Research, 2011.

In this paper,. primary task of this master's thesis is to create a theoretical and practical basis of preprocessing of printed text for optical character recognition using forward-feed neuralnetworks. Demonstration application was created and its parameters were set according to results of realized experiments.

[6]]Nisha Sharma et al, "Recognition for handwritten English letters: A Review"International Journal of Engineering and Innovative Technology (IJEIT) Volume 2

This paper gives an overview of research work carried out for recognition of hand written English letters. In Hand written text there is no constraint on the writing style. Hand written letters are difficult to recognize due to diverse human handwriting style, variation in angle, size and shape of letters. Various approaches of hand written character recognition are discussed here along with their performance.

[7] J.Pradeep et al,, "Diagonal based feature extraction for handwritten alphabets recognition System using neural network"International Journal of Computer Science and Information Technology (IJCSIT), Vol 3, No 1, Feb 2011

An off-line handwritten alphabetical character recognition system using multi layer feed forward neural network is described in the paper. A new method, called, diagonal based feature extraction is introduced for extracting the features of the handwritten alphabets. The proposed recognition system performs quite well yielding higher levels of recognition 6 accuracy compared to the systems employing the conventional horizontal and vertical methods of feature extraction.

[8] Agarwal, M., Shalika, V. T., & Gupta, P. (2019). Handwritten Character Recognition using Neural Network and Tensor Flow. International Journal of Innovative Technology and Exploring Engineering (IJITEE).

After going through the paper, it was concluded that feature extraction method like diagonal and direction techniques are way better in generating high accuracy results compared to many of the traditional vertical and horizontal methods. Also using a Neural network with best tried layers gives the plus feature of having a higher tolerance to noise thus giving accurate results.

[9] Karthikeyan, U., & Vanitha, M. (2019). A Study on Text Recognition using Image Processing with Datamining Techniques.

In this paper, an overview of various text recognition techniques, methods and recognition algorithms has been presented. Based on the literature review various text recognition algorithms accuracy are discussed. The detailed steps and flow of the text recognition techniques by surveying that image acquisition, preprocessing, feature extraction, classification, and post-processing from many research articles. Merits and demerits of text recognition algorithms are discussed.

[10]Chen, L., & Li, S. (2018, November). Improvement research and application of text recognition algorithm based on CRNN. In Proceedings of the 2018 International Conference on Signal Processing and Machine Learning (pp. 166-170).

The objective of the paper is to develop a web application which simulates an optical character recognition system by providing the functionality of text recognition is satisfied. This functionality is successfully implemented using deep learning by applying a CRNN architecture which is a combination of CNN and RNN. The development of this project using this architecture allowed the exploration of different neural network architectures leading to a hybrid structure of using CNNs and RNNs along with CTC loss.

[11] Stricker, D. (2019, March). Multi-font Printed Amharic Character Image Recognition: Deep Learning Techniques. In Advances of Science and Technology: 6th EAI International Conference, ICAST 2018, Bahir Dar, Ethiopia, October 5-7, 2018, Proceedings (Vol. 274, p. 322). Springer.

The objective of there proposed method, to optimize Convolution Neural Networks (CNN) using Simulated Annealing (SA), has been achieved for optical character recognition. Theclassification accuracy from the proposed method is lower than the original of CNN for variation of fonts. This

proposed method could potentially be employed and tried for benchmark dataset RETAS.The proposed method proves the efficiency of character recognition.

[12]Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR). IEEE Access, 8, 142642-142668.

It was observed that researchers are increasingly using Convolutional Neural Networks(CNN) for the recognition of handwritten and machine printed characters. This is due to the fact that CNN based architectures are well suited for recognition tasks where input is image. CNN were initially used for object recognition tasks in images

[13]Panwar, N. N. S. (2012). Handwritten text recognition system based on neural network. Journal of Computer and Information Technology, 2(2), 95-103.

A proposed handwritten character recognition system has been designed and tested. A comparison with related work has been presented. ANNs have been trained for this purpose with various types of input samples and that's why the developed program has an ability to test and classify the input character into 52 different classes with an accuracy of more than 95%.

[14]Singh, S. (2013). Optical character recognition techniques: a survey. Journal of emerging Trends in Computing and information Sciences, 4(6), 545-550.

This paper is a detailed discussion about handwritten character recognize and include various concepts involved, and boost further advances in the area. The accurate recognition is directly depending on the nature of the material to be read and by its quality. Current research is not directly concern to cursive handwriting and to recognize the child handwriting which require high supervised system.

[15] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012, November). End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st international conference on pattern recognition (ICPR2012) (pp. 3304-3308). IEEE.

The work described in this thesis presents an alternative means of approaching the problem of end-to-end text recognition using techniques from unsupervised feature learning and large scale convolutional architectures. In particular, our system integrates the specificity of learned features for capturing the underlying data with the vast representational power of a two-layer convolutional neural network.

III. **METHODOLOGY :**

Proposed system of different types of neural networks to predict sequences of characters or words. For many years, handwritten text recognition systems have used the Hidden Markov Models (HMM for the transcription task, but recently, through Deep Learning, the Convolutional Recurrent Neural Networks (CRNN) approach has been used to overcome some limitations of HMM. To exemplify a CRNN model.

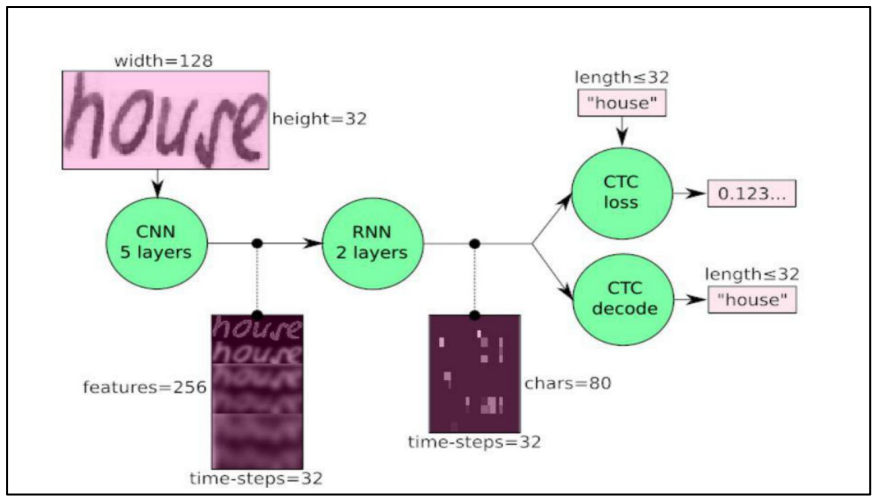


Fig. Overview of CRNN Network

The workflow can be divided into 3 Modules :

Module 1: the input image is fed into the CNN layers to extract features. The output is a feature map.

Module 2: through the implementation of Long Short-Term Memory (LSTM), the RNN is able to propagate information over longer distances and provide more robust features to training.

Module 3: with RNN output matrix, the Connectionist Temporal Classification (CTC) calculates loss value and also decodes into the final text.

Explaining module 3 (CTC) is the same for all architectures presented, then the Vanilla Beam Search method is used, since it doesn't require a dictionary for its application, unlike other know n methods such as Token Passing and Word Beam Search. Therefore, the architectures presen ted in the following sections only act in Module 1 and 2.

In addition, the charset for encoding text is also the same for all datasets. So, the list used consists of 95 printable characters from ASCII table (Figure) by default and doesn't contain accented letters.

```
 !"#$%&\'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}
```

CNN: 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 operation. First, the convolution operation, which applies a filter kernel of size 5×5 in the first two layers and 3×3 in the last three layers to the input. Then, the non-linear RELU function is applied. Finally, a pooling layersummarizes image regions and outputs a downsized version of the input. While the image height is downsized by 2 in each layer, feature maps (channels) are added, so that the output feature map (or sequence) has a size of 32×256.

RNN: 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 of RNNs is used, as it is able to propagate information through longer distances and provides more robust training-characteristics than vanilla RNN. The RNN output sequence is mapped to a matrix of size 32×80. The IAM dataset consists of 79 different characters, further one additional character is needed for the CTC operation (CTC blank label), therefore there are 80 entries for each of the 32 time-steps.

CTC (Connectionist Temporal Classification) : while training the NN, the CTC is given the RNN output matrix and the ground truth text and it computes the **loss value**. While inferring, the CTC is only given the matrix and it decodes it into the **final text**. Both the ground truth text and the recognized text can be at most 32 characters long.

CNN output: Fig. 4 shows the output of the CNN layers which is a sequence of length 32. Each entry contains 256 features. Of course, these fea tures are further processed by the RNN layers, however, some features already show a high correlation with certain high-level properties of the input image: there are features which have a high correlation with characters (e.g. “e”), or with duplicate characters (e.g. “tt”), or with character-properties such as loops (as contained in handwritten “l”s or “e”s).

RNN output: Fig. 5 shows a visualization of the RNN output matrix for an image containing the text “little”. The matrix shown in the top-most graph contains the scores for the characters including the CTC blank label as its last (80th) entry. The other matrix-entries, from top to bottom, correspond to the following characters:

!#&’()*+,-./0123456789:;?ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz.

It can be seen that most of the time, the characters are predicted exactly at the position they appear in the image (e.g. compare the position of the “i” in the image and in the graph). Only the last character “e” is not aligned. But this is OK, as the CTC operation is segmentation-free and does not care about absolute positions. From the bottom-most graph showing the scores for the characters “l”, “i”, “t”, “e” and the CTC blank label, the text can easily be decoded: we just take the most probable character from each time-step, this forms the so called best path, then we throw away repeated characters and finally all blanks: “l---ii--t-t--l-…-e” → “l---i--t-t--l-…-e” → “little”.

IV. **ARCHITECTURAL FRAMEWORK :**

CRNN involves a convolutional feature extractor that encodes visual details into latent vectors, followed by a recurrent sequence decoder that turns the latent vectors into human-understandable characters. The whole architecture is trained end-to-end via Connectionist Temporal Classification (CTC) loss function or attention mechanism.

Inside CRNN, the role of the sequence decoder i.e. LSTM has been reported to serve as a language model. It is observed that OCR model attains higher accuracy for meaningful text line than for random text line.

CRNN firstly combined CNN and RNN to extract sequential visual features of a given text image, and then directly fed them into a CTC decoder to predict the best character category of each time step, where CTC only maximized the probability of all the paths that can reach the ground truth according to the visual classification of each position.

Improvement over RNN:

LSTM (Long Short-Term Memory) Networks :

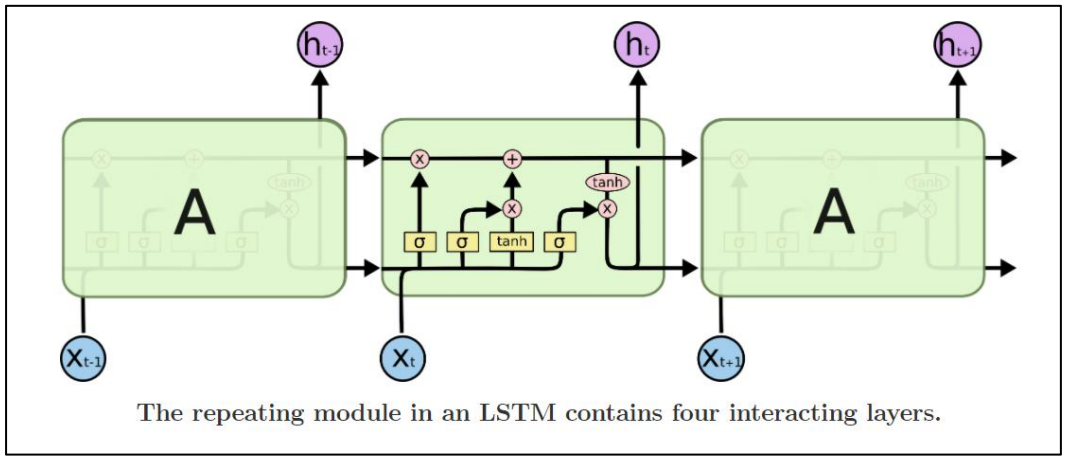
In order to add a new information, RNN transforms the existing information completely by applying a function. To overcome this limitation, we use LSTM algorithm. LSTMs make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a particular cell state has three different dependencies.

These dependencies can be generalized as:

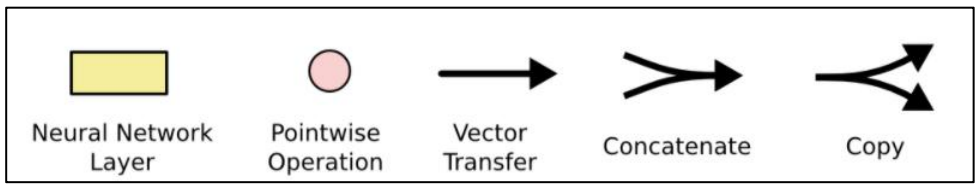
- The previous cell state i.e. the information that was present in the memory after the previous time step.
- The previous hidden state i.e., this is the same as the output of the previous cell.
- The input at the current time step i.e. the new information that is being fed in at that moment.

ARCHITECTURE OF LSTM :

LSTMs have a chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



The repeating module in an LSTM contains four interacting layers.



In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through.

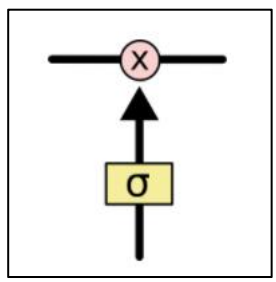


Figure representing gates

Gates are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means don't let anything through, while a value of one means let everything through.

Working of LSTM is divided into 3 steps:

1) The first step in LSTM is to decide what information we're going to throw away from the cell state. The decision is made by the gates.

2) The next step is to decide what new information we are going to store in the cell state. This has

two parts. First, a sigmoid layer called the or the input gate layer decides which values is to be updated. Next, the tanh layer creates a vector of new candidate values, that could be added to the state. After this we update the old cell state with the new one.

3) The final step is to decide the output. The output will be based on our cell state. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh and multiply it by the output of the sigmoid gate, so that we output the parts that we have decided to.

Thus, the proposed Gated-CNN-BLSTM architecture preserves the low number of parameters (around 820 thousand) and high recognition rate. It shows in detail the distribution of parameters and hyperparameter through 11 convolutional layers (5 gated included) and 2 BLSTM.
(Bidirectional Long Short Term Memory) .

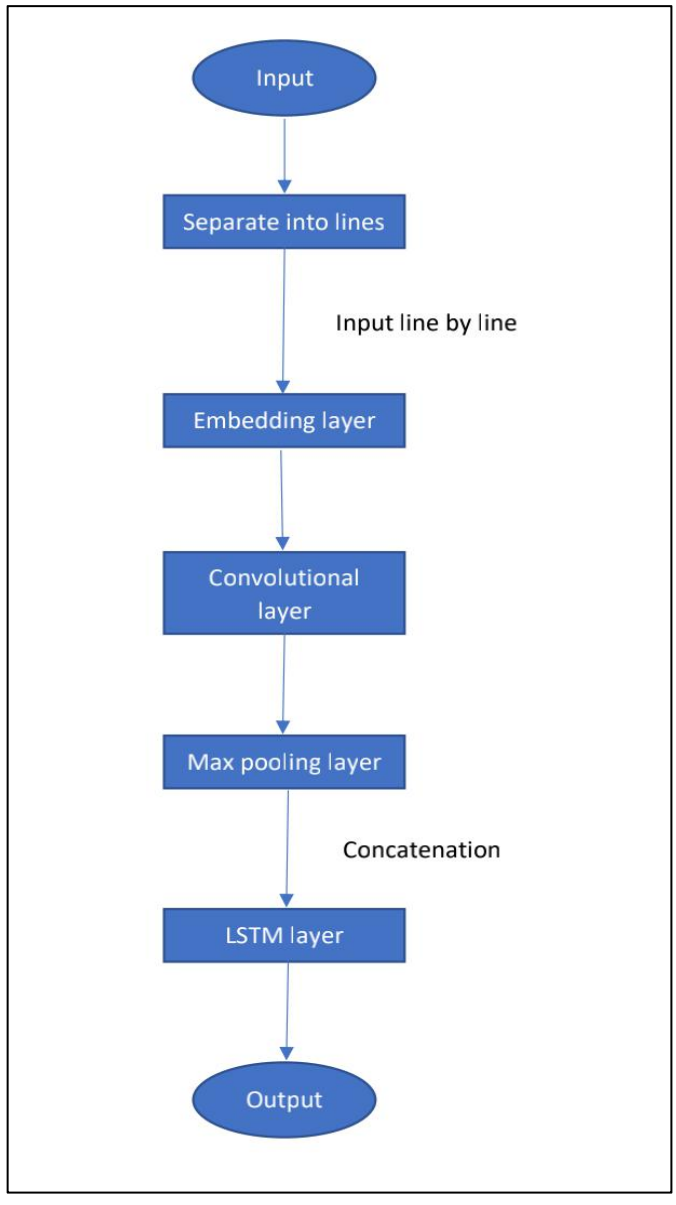


Fig. Flor Architecture Structure

ALGORITHM :

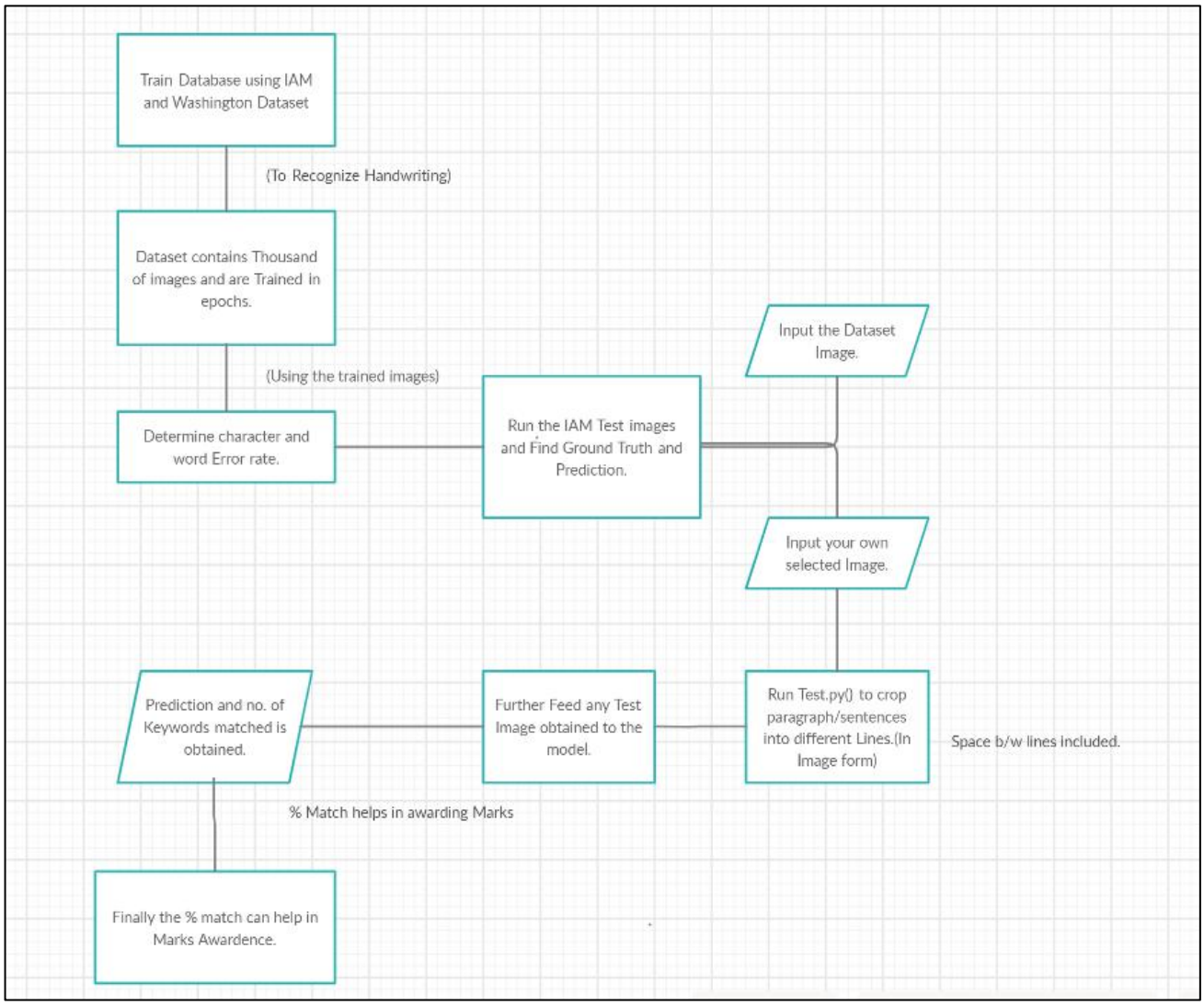


Fig. An algorithmic overview of how text recognition is happening in the proposed model

PSUEDOCODE :

```
INPUT the sample image
    FOR each image select a top starting width and bottom starting width
        FOR each images in the range(i,n) :
            Set area = (right,top,left,botton);
            Set image as cropped image
            IF image is cropped :
                top = top + x; bottom = bottom + y;
            END IF
        END FOR
    END FOR
Save the cropped image
```

V. **EXPERIMENTAL RESULTS :**

Command to train the dataset :

prakhardungarwal@Prakhars-MacBook-Air source_main %
python3 main.py --source iam --arch flor --train --epochs 5

Output :

```
----------------------------------------------------------------
Train for 336 steps, validate for 47 steps
Epoch 1/5
 13/336 [>.............................] - ETA: 28:27 - loss: 12.9712[1]  - terminated  python3 main.py --source iam --image test2.jpg
335/336 [============================>.] - ETA: 5s - loss: 15.5624
336/336 [==============================] - 1741s 5s/step - loss: 15.5449 - val_loss: 15.8678kpoint_weights.hdf5
Epoch 2/5
335/336 [============================>.] - ETA: 4s - loss: 14.97792020-10-15 10:14:59.864214: W tensorflow/core/kernels/data/generator_dataset_op.cc:103] Error occurred whe
n finalizing GeneratorDataset iterator: Cancelled: Operation was cancelled

336/336 [==============================] - 1580s 5s/step - loss: 14.9515 - val_loss: 15.6529/checkpoint_weights.hdf5
Epoch 3/5
335/336 [============================>.] - ETA: 4s - loss: 14.71432020-10-15 10:42:18.571406: W tensorflow/core/kernels/data/generator_dataset_op.cc:103] Error occurred whe
n finalizing GeneratorDataset iterator: Cancelled: Operation was cancelled

336/336 [==============================] - 1635s 5s/step - loss: 14.6878 - val_loss: 15.0551/checkpoint_weights.hdf5
Epoch 4/5
335/336 [============================>.] - ETA: 9s - loss: 14.5611 2020-10-15 11:36:42.186772: W tensorflow/core/kernels/data/generator_dataset_op.cc:103] Error occurred wh
en finalizing GeneratorDataset iterator: Cancelled: Operation was cancelled

336/336 [==============================] - 3260s 10s/step - loss: 14.5402 - val_loss: 14.5543checkpoint_weights.hdf5
Epoch 5/5
335/336 [============================>.] - ETA: 10s - loss: 14.32692020-10-15 12:37:55.864524: W tensorflow/core/kernels/data/generator_dataset_op.cc:103] Error occurred wh
en finalizing GeneratorDataset iterator: Cancelled: Operation was cancelled

336/336 [==============================] - 3671s 11s/step - loss: 14.3012 - val_loss: 14.4193checkpoint_weights.hdf5
2020-10-15 12:38:35.832422: W tensorflow/core/kernels/data/generator_dataset_op.cc:103] Error occurred when finalizing GeneratorDataset iterator: Cancelled: Operation was c
ancelled
2020-10-15 12:38:35.850347: W tensorflow/core/kernels/data/generator_dataset_op.cc:103] Error occurred when finalizing GeneratorDataset iterator: Cancelled: Operation was c
ancelled
```

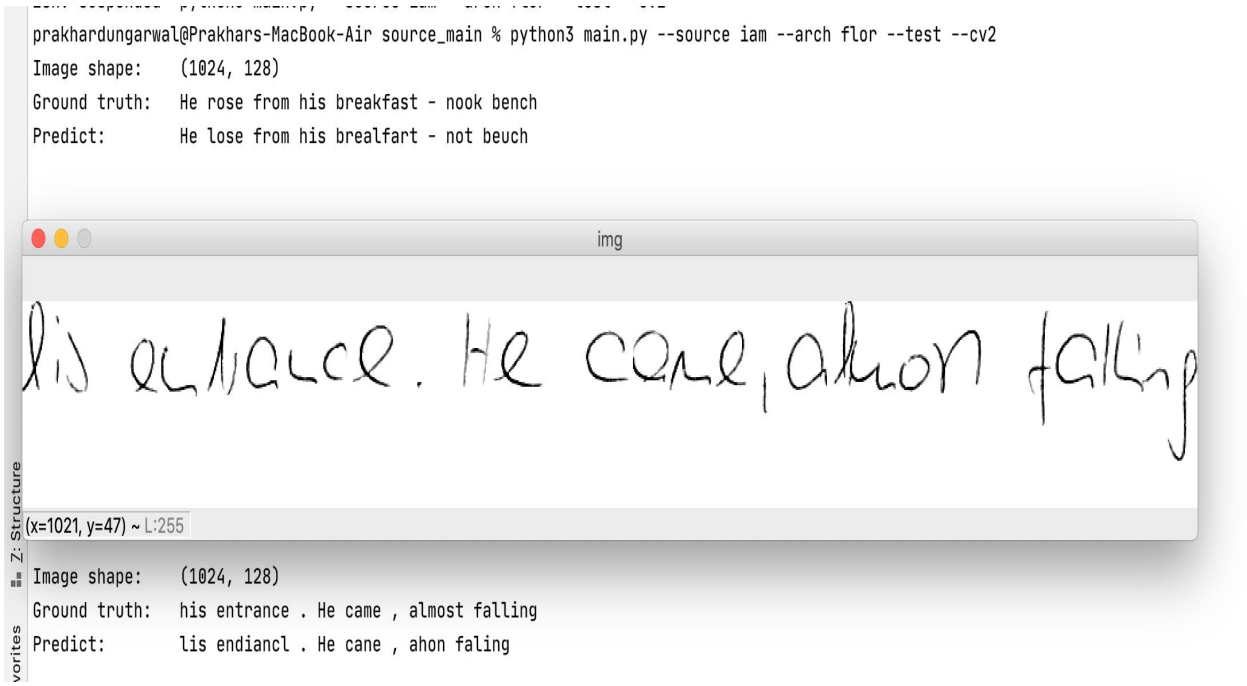
```
Total train images:      5369
Total validation images: 744
Batch:                   16

Total time:              3:18:07.999923
Time per epoch:          0:39:37.599985
Time per item:           0:00:00.388942

Total epochs:            5
Best epoch               5

Training loss:           14.31240539
Validation loss:         14.41931232
```

Fig. Training Results for the Dataset when trained for 5 epochs

Command to Show trained Dataset Output :

prakhardungarwal@Prakhars-MacBook-Air source_main % python3 main.py --source iam --arch flor --test --cv2

OUTPUT :



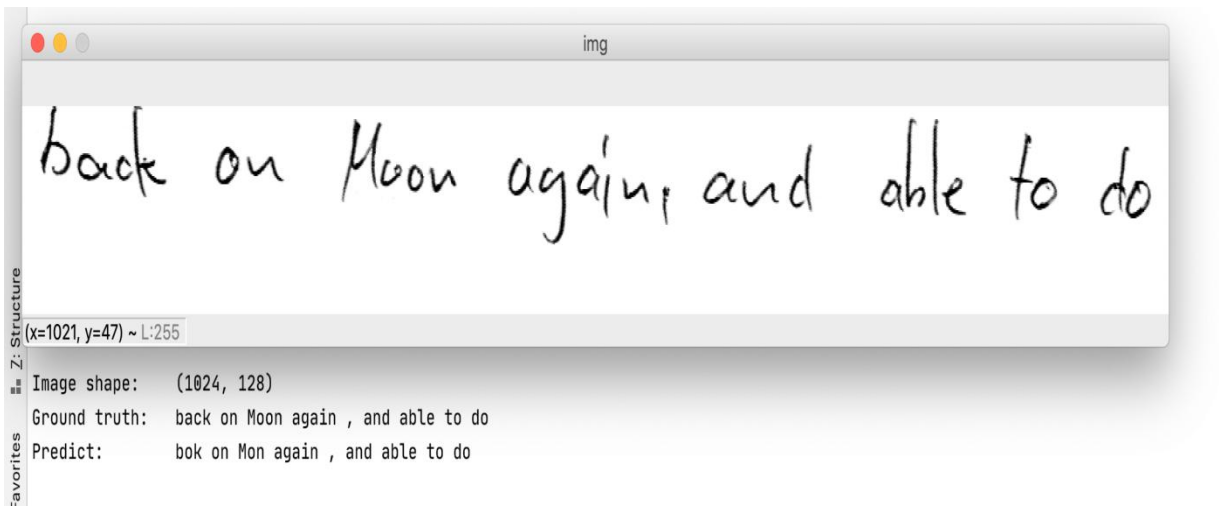


Fig. Output of trained IAM Dataset images

Command to split your image into sub-images :

prakhardungarwal@Prakhars-MacBook-Air source_main % python3 test.py

```
main.py  test.py  TestImage.jpg

1  from PIL import Image
2  img = Image.open(r"/Users/prakhardungarwal/Downloads/IMAGE PROJECT/Code Files/source_main/image/TestImage.jpg")
3  top=0
4  bottom=300
5  for i in range(0,5):
6      area = (20, top,3000, bottom)
7      cropped_img = img.crop(area)
8      top=top+150
9      bottom=bottom+150
10     im1 = cropped_img.save(r"/Users/prakhardungarwal/Downloads/IMAGE PROJECT/Code Files/source_main/image/"+str(i)+".jpg")
```

Fig. Test Image path and code for cropping test image

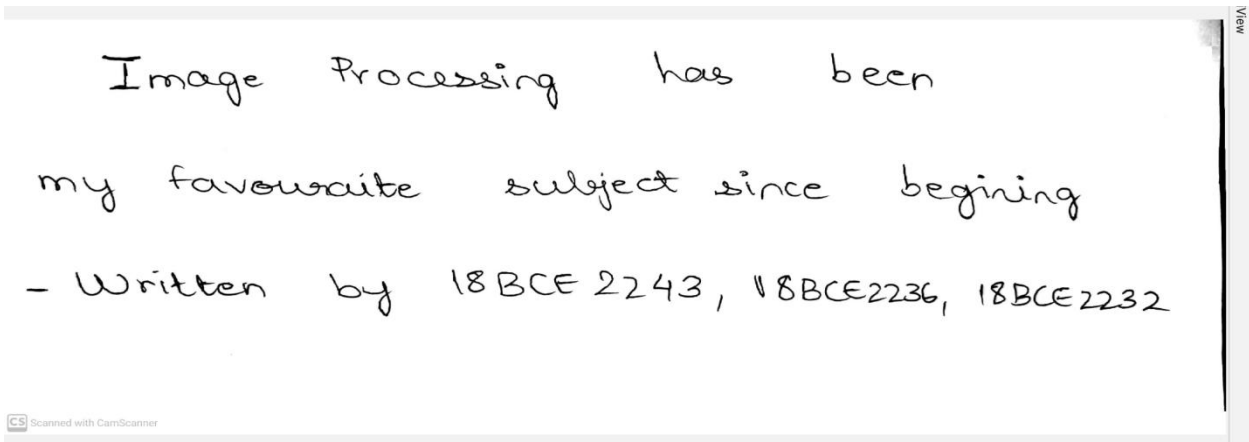


Fig. Test Image

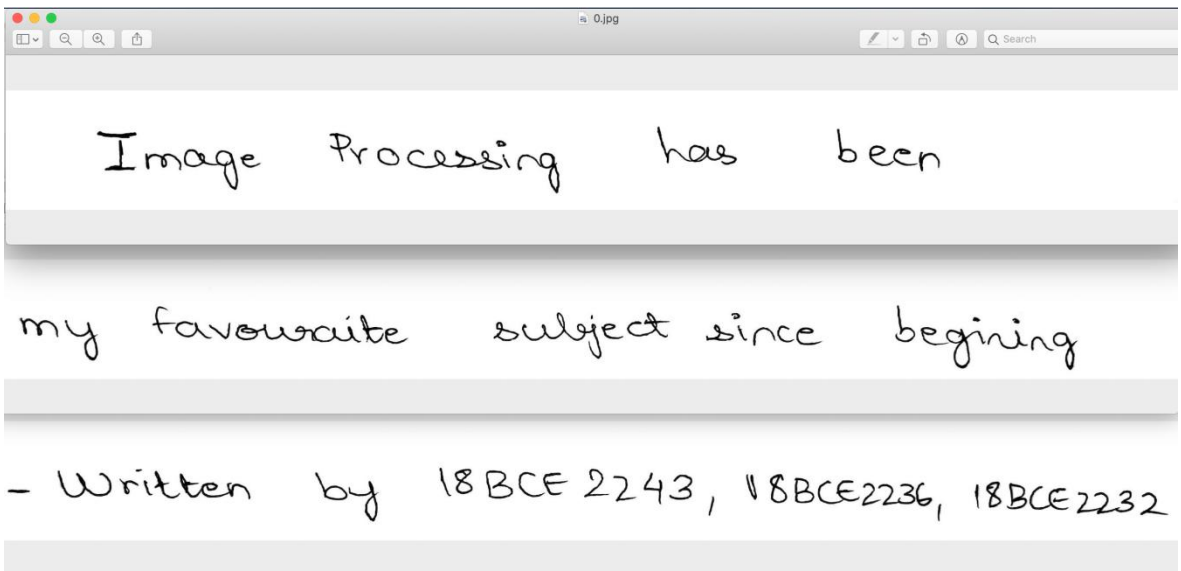


Fig. Cropped Images

Command to run model for self image :

Input Image :

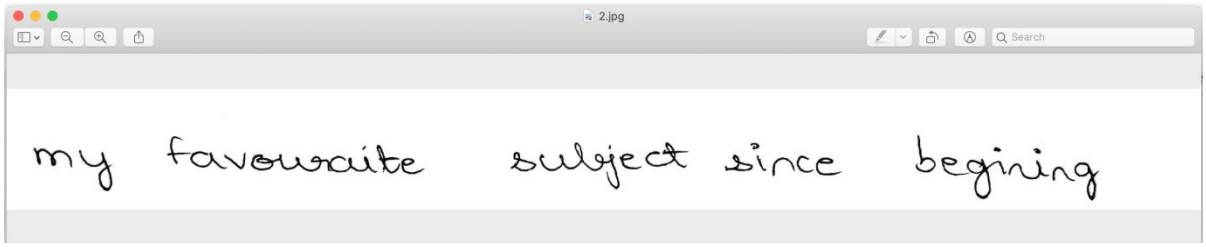


Fig. Input Image

Command : python3 main.py --source iam --image 2.jpg

Output :

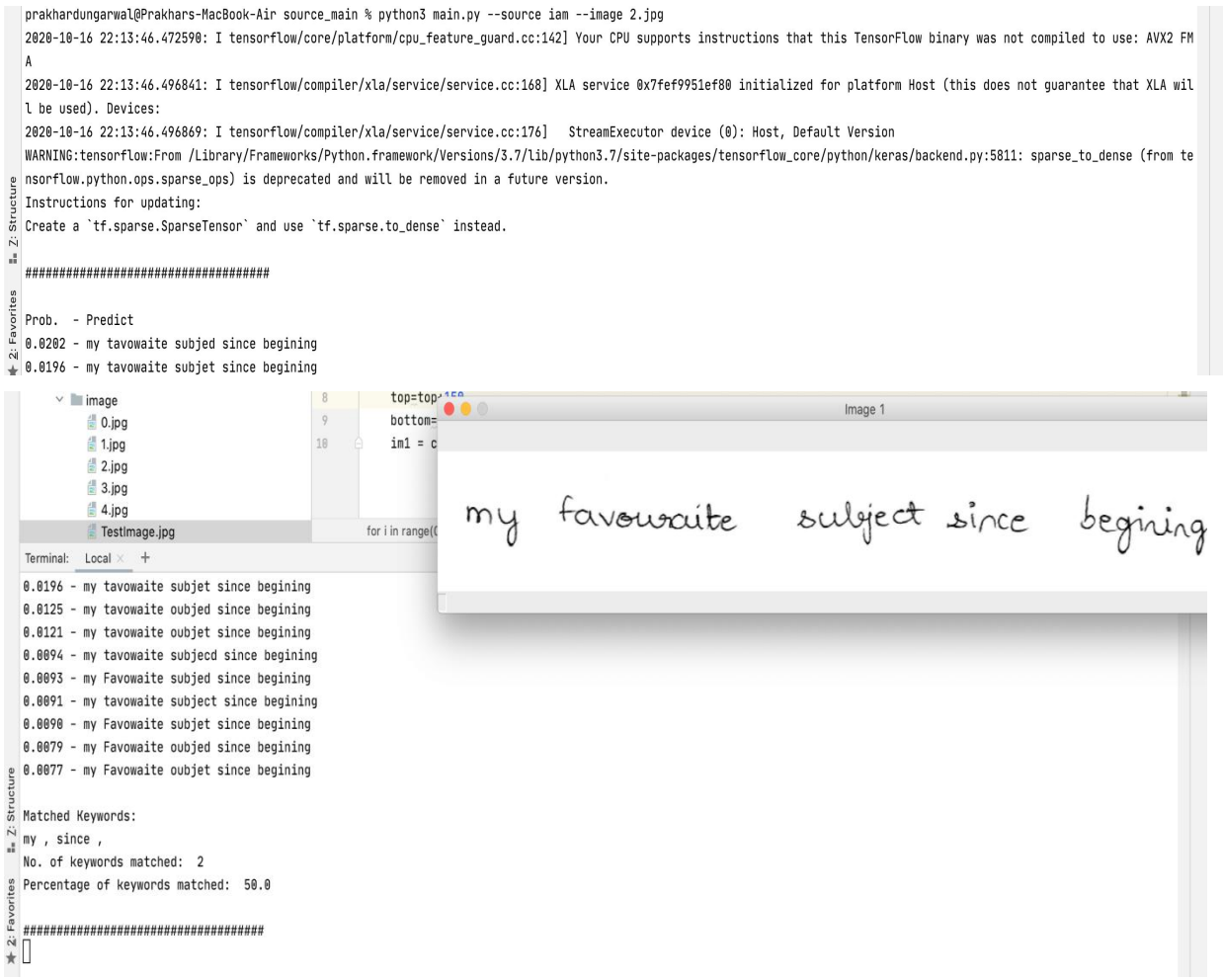


Fig. Output for self handwritten Image

VI. **PERFORMANCE BASED EVALUATION :**

Command to find the character error rate, word error rate etc. :

prakhardungarwal@Prakhars-MacBook-Air
source_main % python3 main.py --source iam --test

Output :

```
prakhardungarwal@Prakhars-MacBook-Air source_main % python3 main.py --source iam --test
2020-10-15 13:49:12.582154: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FM
A
2020-10-15 13:49:12.641345: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fb92836ee80 initialized for platform Host (this does not guarantee that XLA wil
l be used). Devices:
2020-10-15 13:49:12.641376: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
Model Predict
90/90 [==============================] - 79s 878ms/step
CTC Decode
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tensorflow_core/python/keras/backend.py:5811: sparse_to_dense (from te
nsorflow.python.ops.sparse_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Create a `tf.sparse.SparseTensor` and use `tf.sparse.to_dense` instead.
90/90 [==============================] - 47s 520ms/step
Total test images:    1425
Total time:           0:02:06.422070
Time per item:        0:00:00.088717

Metrics:
Character Error Rate: 0.10429892
Word Error Rate:      0.32748035
Sequence Error Rate:  0.94947368
```

Metrics Evaluated :

Character Error Rate: 0.10429892
Word Error Rate: 0.32748035
Sequence Error Rate: 0.94947368

No. of Trained Epochs	Character Error Rate	Word Error Rate	Sequence Error Rate
2	0.19334313	0.38423344	0.97423423
5	0.10429892	0.32748035	0.94947368
10	0.09434345	0.27899879	0.84522545

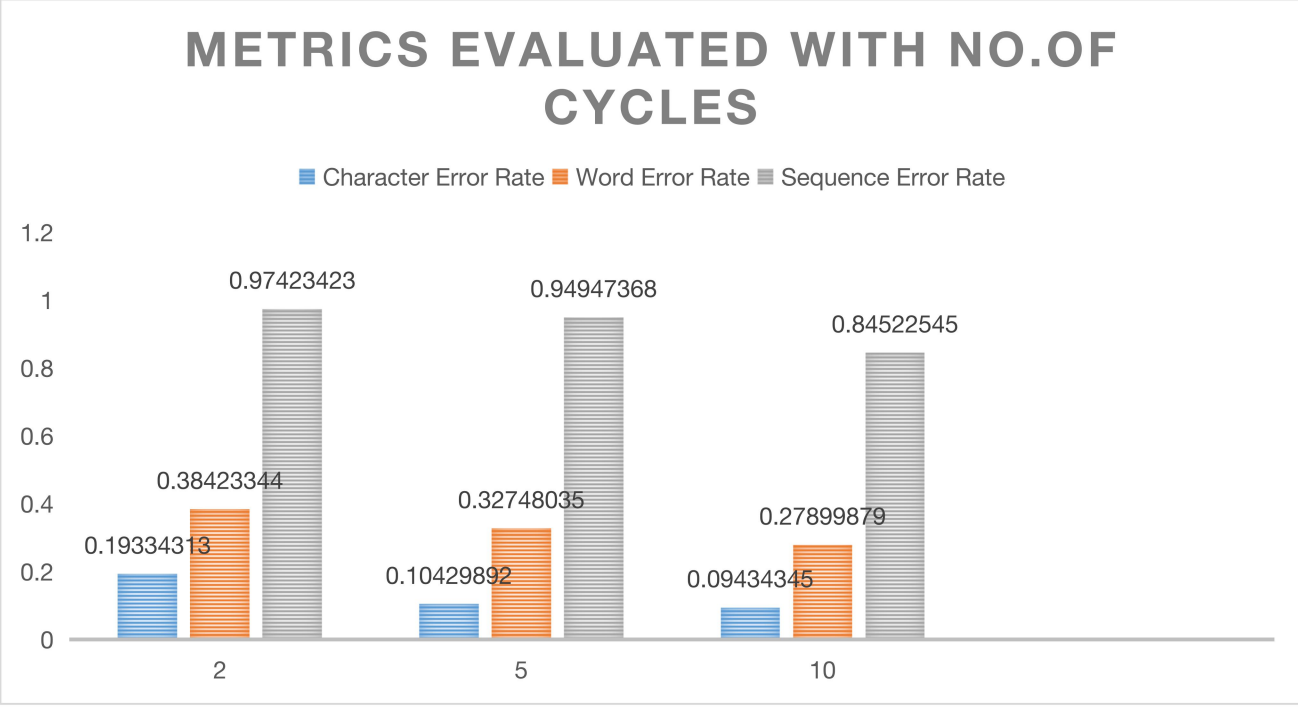


Fig. Comparing Metrics for text recognition rate vs No. Of training cycles or epochs

TABLE SHOWING COMPARISION OF VARIOUS ALGORITHMS :

Algorithm Used	Accuracy Percentages
CNN	23%
HMM	56%
CNN+HMM Hybrid	81%
CRNN	91%

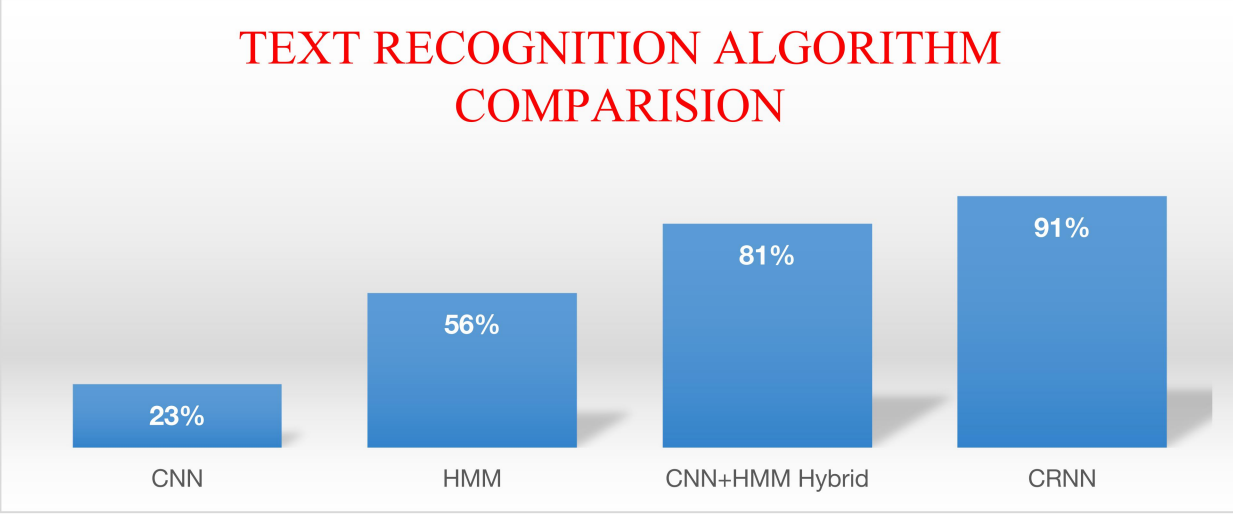


Fig. Comparing Different Algorithms for text Recognition

From the above observations it is clear that CRNN is the best algorithm for text recognition as it has the highest accuracy, as we train our dataset constantly for more cycles (more epochs) the error rates will come down making the recognition and detections even more accurate.

VII. **CONCLUSION :**

The proposed evaluation model suggested here can do great help for our education system especially in future times as our education systems would change these systems will help reduce teachers time in checking descriptive answers and give them a opportunity to work on more productive things. The system here has proved to be capable of matching the keywords of the modal answer and awarding marks based on the percentage of matching of these keywords. Hence the said system could be of great utility to the educators whenever they need to take a quick test for revision purpose,as it saves them the trouble of evaluating the bundle of papers.

In future online teaching learning methods will be widely used in many institutions.
Descriptive checking methods will help to evaluate students answer. Our proposed method evaluate it more efficiently and accurately. In our system, computers have been programmed to scan the papers, recognize the possible right responses and compile the marks.

The propsoed model incoperates great image processing and deep learning techniques to help generate the most effective way for paper checking and text recognition but what we can improve on this is that the proposed system is not capable of scanning a whole document and then recognizing text it just scans a page and then crops it into text lines to recognize, if this can further be optimized to make as a whole document scanner and auto-evaluator it could turn out to be a great revolution in the paper checking methodology and can save millions of hours of teachers.

VIII. **REFERENCES :**

[1] PAQUET, M. T. (2017). Structuring of Hidden Information in Markov Modeling with Application to Handwriting Recognition (Doctoral dissertation, telecom-paristech).
https://www.researchgate.net/publication/335170252_Structuring_of_Hidden_Information_in_Markov_Modeling_with_Application_to_Handwriting_Recognition

[2] Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C. V. (2018, August). Improving cnn-rnn hybrid networks for handwriting recognition. In 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 80-85). IEEE.
http://cdn.iiit.ac.in/cdn/cvit.iiit.ac.in/images/ConferencePapers/2018/improving-cnn-rnn.pdf

[3] Balci, B., Saadati, D., & Shiferaw, D. (2017). Handwritten text recognition using deep learning. CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring, 752-759.
http://cs231n.stanford.edu/reports/2017/pdfs/810.pdf

[4]Dongare, S. A., Kshirsagar, D. B., & Waghchaure, M. S. V. (2014). Handwritten devanagari character recognition using neural network. IOSR J Comput Eng (IOSR-JCE) Ume, 16(2), 74-79.
https://link.springer.com/chapter/10.1007/978-3-030-43192-1_52

[5]]Miroslav NOHAJ, Rudolf JAKA, "Image preprocessing for optical character recognition using neural networks"Journal of Patter Recognition Research, 2011.
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.226&rep=rep1&type=pdf

[6]]Nisha Sharma et al, "Recognition for handwritten English letters: A Review"International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, January 2013
https://www.researchgate.net/publication/258725510_Recognition_for_Handwritten_English_Letters_A_Review

[7] J.Pradeep et al,, "Diagonal based feature extraction for handwritten alphabets recognition System using neural network"International Journal of Computer Science and Information Technology (IJCSIT), Vol 3, No 1, Feb 2011
https://ieeexplore.ieee.org/document/5941921

[8] Agarwal, M., Shalika, V. T., & Gupta, P. (2019). Handwritten Character Recognition using Neural Network and Tensor Flow. International Journal of Innovative Technology and Exploring Engineering (IJITEE).
https://www.ijitee.org/wp-content/uploads/papers/v8i6s4/F12940486S419.pdf

[9] Karthikeyan, U., & Vanitha, M. (2019). A Study on Text Recognition using Image Processing with Datamining Techniques.
https://www.researchgate.net/publication/331590226_A_Study_on_Text_Recognition_using_Image_Processing_with_Datamining_Techniques/link/5c82494b458515831f8fc6a0/download

[10]Chen, L., & Li, S. (2018, November). Improvement research and application of text recognition algorithm based on CRNN. In Proceedings of the 2018 International Conference on Signal Processing and Machine Learning (pp. 166-170).
https://dl.acm.org/doi/abs/10.1145/3297067.3297073

[11] Stricker, D. (2019, March). Multi-font Printed Amharic Character Image Recognition: Deep Learning Techniques. In Advances of Science and Technology: 6th EAI International Conference, ICAST 2018, Bahir Dar, Ethiopia, October 5-7, 2018
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.3198&rep=rep1&type=pdf

[12]Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR). IEEE Access, 8, 142642-142668.
https://arxiv.org/pdf/2001.00139.pdf

[13]Panwar, N. N. S. (2012). Handwritten text recognition system based on neural network. Journal of Computer and Information Technology, 2(2), 95-103.
https://www.semanticscholar.org/paper/RECOGNITION-SYSTEM-BASED-ON-NEURAL-NETWORK-Neeta-Panwar/81e64b3694bcd3f18b24aafbb09690e78f67ce84?p2df

[14]Singh, S. (2013). Optical character recognition techniques: a survey. Journal of emerging Trends in Computing and information Sciences, 4(6), 545-550.
http://www.ijesrt.com/issues%20pdf%20file/Archive-2018/April-2018/34.pdf

[15] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012, November). End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st international conference on pattern recognition (ICPR2012) (pp. 3304-3308). IEEE.
https://ieeexplore.ieee.org/abstract/document/6460871