Handwritten Text Recognition using Deep Learning for Automated Paper Checking

Prof. Jayakumar K.[1]
Associate Professor Grade, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology Vellore, India

Prakhar Dungarwal[2]
Student, *School of Computer Science and Engineering (SCOPE),* Vellore Institute of Technology *Vellore, India*

---***---

***Abstract*—** Amid these deceptive occasions of the COVID-19 pandemic, everybody needs to build up a fully-automated online assessment framework. Because of the expanding number of courses and showing up understudies numerous long periods of inspector and a ton of endeavors are needed for the viable assessment. Technologies also, advances can be utilized to take care of such a complex issue.

The objective is to assess and appoint scores to engaging answer which are tantamount to those of human allowed score by coupling deep learning advances and Image Processing Techniques. Our proposed framework incorporates a calculation that is equipped for assessing an answer content dependent on your penmanship and contrasting it and the at first entered answer watchwords both catchphrase savvy and looking at every one of these watchwords.

The goal of our task is to decrease the ideal opportunity for manual paper checking by our proposed HTR Model. We in this paper have executed Convolutional Recurrent Neural Networks (CRNN) which are superior to convention Hidden Markov Model (HMM) both for execution effectiveness and better text recognition.

So the info offered to the framework is catchphrases of the response, handling in the framework is through Handwritten Text Recognition (HTR) models and picture preparing methods yield is the level of imprints out of absolute imprints to be granted to the framework. So it benefits both the educator and understudy, instructors spare their paper checking time as there is no compelling reason to check the entire answer they can grant marks dependent on rate assessed by the framework and understudy benefits as regardless of whether his answers don't contain the specific watchwords the Deep Learning and Image Processing Systems are sufficiently wise to give marks dependent on coordinating catchphrases and gets a decent lot of imprints.

Keywords*— *Handwritten Text Recognition (HTR), fully- automated, Convolutional Recurrent Neural Networks (CRNN), Hidden Markov Model (HMM), Deep Learning.

I. INTRODUCTION

Handwriting recognition has been one of the most interesting and testing research territories in the field of image processing and example recognition in the ongoing years. It contributes hugely to the headway of a mechanization cycle and can improve the interface between man and machine in various applications. A few examination works have been zeroing in on new procedures and techniques that would diminish the processing time while giving higher recognition exactness.

The main significant advance in any handwritten recognition framework is pre-processing followed by division and highlight extraction. Pre-processing incorporates the means that are needed to shape the information image into a structure appropriate for the division. In the division, the info image is portioned into singular characters, and afterward, each character is resized into m x n pixels towards the preparation organization.

The manual framework for the assessment of Subjective Answers for specialized subjects includes a great deal of time and exertion of the evaluator. Emotional answers have different boundaries whereupon they can be assessed, for example, the inquiry explicit substance and composing style. Assessing abstract answers is a basic undertaking to Perform. At the point when an individual assesses anything, the nature of assessment may change alongside the feelings of the individual. Assessing PCs utilizing clever procedures guarantees consistency in stamping as a similar derivation component is utilized for all the understudies. In Machine Learning, all outcome is just founded on the info information gave by the client.

- Our Proposed System utilizes Deep Learning and Image Processing to tackle this issue.
- Our Algorithm plays out an errand like Tokenizing words and sentences, Part of Speech labeling, Chunking, chinking, Lemmatizing words, and Wordnetting to assess the emotional answer.

- Along with it, our proposed calculation gives the semantic importance of the context.

Our System is partitioned into two modules:

Extracting the information from the examined images and coordinating it in the best possible way and the product will take a checked duplicate of the appropriate response as info and afterward after the preprocessing step, it will separate the trial of the appropriate response.

1)This text will again experience processing to manufacture a model of catchphrases and capabilities. Model answer sets and watchwords ordered as a reference will be the contribution also. The classifier will at that point, in light of the preparation will offer imprints to the responses. Imprints to the appropriate response will be the last yield.

The requirement for online assessment excited for the most part to beat the downsides of the current framework just as the worldwide COVID-19 emergency. The primary point of the venture is to guarantee easy to use and more intuitive programming to the client. The online assessment is a lot quicker and clear technique to characterize all the applicable stamping plans. It carries a lot of straightforwardness to the current technique for answer checking The responses to all the inquiries after the extraction would be put away in an information base. The information base is planned as with the end goal that it is effectively open. Robotizing dreary errands has been the principal point of mechanical and innovative upheaval.

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 the 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 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 writer presents the distinctive pre- processing approaches coping with significant sorts of inconstancies in handwriting, to be specific: size, slant, inclination, guidelines, and clamor. A few methodologies depended on carefully assembled heuristics while others depended on scholarly techniques utilizing neural organizations.

The last ones could accomplish better execution, however, they require manual labeling of the training information. They have exhibited that guidelines have an enormous impact on the presentation and proposed ways to deal with recognize and eliminate them. This has prompted a nearby execution for pictures with and without guidelines, which implies that the impact of the guidelines was effectively reduced. In the end, we demonstrated the improvement brought by all the preprocessing procedures embraced on the presentation of our baseline frameworks, HMM, and BLSTM.

V. Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C. (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 creator introduced powerful approaches to train a CNNRNN half breed design using synthetic information and domain explicit picture standardization and increase. He has likewise demonstrated the individual commitments of every one of these modules for improving the acknowledgment rates at both line and word levels. In the future, he might likewise want to integrate our line-level acknowledgment model with language model based decoding to further upgrade our acknowledgment execution.

[2] 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 utilized a profound organization model to perceive manually written writings. Preprocessing of the dataset is likewise considered as the main consideration behind the precision of the models.

Training of an organization requires strong equipment to actualize enormous datasets productively. It is important to train the organization with an enormous measure of the dataset so the organization can undoubtedly perceive the character. Consequently, there is a necessity for high memory and high processing rate to accomplish a proficient network. The model comprises of 5 layers of Convolutional Neural Network (CNN), 2 layers of Recurrent Neural Network (RNN) and yields a character- likelihood framework. This framework is either utilized for CTC misfortune estimation or CTC decoding.

[3] **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 manages the improvement of framework- based technique which is a combination of picture centroid zone a lot centroid zone of the individual character of the mathematical picture. In component extraction using network or zone-based methodology individual character or mathematical picture is isolated into n equivalent estimated lattices or zones then normal separation of all pixels as for picture centroid or matrix centroid is figured. In the combination of picture centroid and zone centroid approach, it figures normal separation of all pixels present in every lattice concerning picture centroid just as zone centroid which gives include a vector of size 2xn highlights. This component vector is introduced to take care of forwarding neural organization for acknowledgment. A complete cycle of Devanagari character acknowledgment works in stages as record preprocessing, division, including extraction using framework based methodology followed by acknowledgment using feed-forward neural organization.

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

In this paper,. essential assignment of this expert's thesis is to make a theoretical and commonsense premise of pre-processing of printed text for optical character acknowledgment using forward-feed neural networks. Exhibition application was made and its boundaries were set according to aftereffects of acknowledged examinations.

[4] Nisha Sharma et al, "Recognition for handwritten English letters: A Review"International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, January 2013.

This paper gives a diagram of exploration work completed for acknowledgment of transcribed English letters. In Hand composed content there is no constraint on the writing style. Manually written letters are hard to perceive because of assorted human handwriting style, variety in point, size and state of letters. Different methodologies of transcribed character acknowledgment are examined here alongside their presentation.

[5] 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 organization is depicted in the paper. Another technique, called, corner to corner based element extraction is introduced for extracting the highlights of the transcribed letter sets. The proposed acknowledgment framework performs very well yielding more significant levels of acknowledgment 6 precision contrasted with the frameworks employing the regular even and vertical techniques for highlight extraction.

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

Subsequent to going through the paper, it was reasoned that include extraction strategy like inclining and course procedures are path better in generating high precision results contrasted with a large number of the conventional vertical and flat techniques. Additionally using a Neural organization with best attempted layers gives the in addition to highlight of having a higher resilience to commotion in this manner giving precise outcomes.

[7] 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).

In this paper, a diagram of different content acknowledgment procedures, techniques and acknowledgment calculations has been introduced. In light of the writing audit different content acknowledgment calculations precision are talked about. The itemized steps and stream of the content acknowledgment strategies by surveying that picture securing, preprocessing, include extraction, grouping, and post-processing from many exploration articles. Benefits and negative marks of text acknowledgment calculations are examined.

[8] 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.

The goal of the paper is to build up a web application which mimics an optical character acknowledgment framework by providing the usefulness of text acknowledgment is fulfilled. This usefulness is effectively actualized using profound learning by applying a CRNN design which is a combination of CNN and RNN. The improvement of this undertaking using this design permitted the investigation of various neural organization models leading to a half and half structure of using CNNs and RNNs alongside CTC misfortune.

III. METHODOLOGY

Proposed system of.different types of.neural organizations to foresee groupings of characters or words. For many years, transcribed content acknowledgment systems have utilized the Hidden Markov Models (HMM for the record task, however, recently, through Deep Learning, the Convolutional Recurrent Neural Networks (CRNN) approach has been utilized to conquer a few restrictions.HMM. To exemplify a CRNN.model.

The work process can be separated into 3 Modules :

Module 1: the information picture is taken care of into the CNN layers to remove highlights. The yield is an element map.

Module 2: through the usage of. Long Short-Term Memory (LSTM), the RNN can engender data over longer separations and give more hearty highlights to preparing.

Module 3: with the RNN yield framework, the Connectionist Temporal Classification (CTC) computes misfortune esteem and deciphers it into the last content.

Clarifying module 3 (CTC) is the equivalent for all models introduced, at that point the Vanilla Beam Search technique is utilized, since it doesn't need a dictionary for its application, dissimilar to other known strategies, for example, Token Passing and Word Beam Search. Along these lines, the designs presented in the accompanying segments only act in Module 1 and 2.

Moreover, the charset for encoding text is additionally the equivalent for all datasets. Thus, the rundown utilized comprises of 95 printable characters from the ASCII table by default and doesn't contain emphasized letters.

CNN: the info picture is taken care of into the CNN layers. These layers are prepared to separate important highlights from the picture. Each layer comprises of three activity. To start with, the convolution activity, which applies a channel portion of size 5×5 in the initial two layers and 3×3 in the last three layers to the information. At that point, the non-direct RELU work is applied. Finally, a pooling layersummarizes picture locales and yields a scaled back variant of the input. While the picture stature is cut back by 2 in each layer, highlight maps (channels) are added, with the goal that the yield includes a guide (or succession) has a size of.32×256.

RNN: the element succession contains 256 highlights for each time-step, the RNN proliferates significant data through this grouping. The well known Long Short-Term Memory (LSTM) usage of.RNNs is utilized, as it can proliferate data through longer separations and gives more powerful preparing qualities than vanilla RNN. The RNN yield succession is planned to a lattice of size 32×80. The IAM dataset comprises of 79 various characters, further one extra character is required for the CTC activity (CTC clear name), along these lines, there are 80 passages for each of the 32 time-steps.

CTC (Connectionist Temporal Classification): while preparing the NN, the CTC is given the RNN yield grid and the ground truth text and it registers the misfortune esteem. While construing, the CTC is only given the network and it deciphers it into the last content. Both the ground truth text and the perceived content can be all things considered 32 characters in length.

CNN yield: The yield of the CNN layers which is an arrangement of. length 32. Each entry contains 256 highlights. Of. course, these features are additionally handled by the RNN layers, nonetheless, a few highlights already show a high relationship with certain significant level properties. The input picture: there are highlights which have a high connection with characters (for example "e"), or with copy characters (for example "tt"), or with character-properties, for example, circles (as contained in transcribed "l"s or "e"s).

RNN yield: A representation of.the RNN yield network for a picture containing the content "pretty much nothing". The network appeared in the top-most chart contains the scores for the characters including the CTC clear name as its last (80th) entry. The other framework passages, through and through, relate to the accompanying characters:

!#&'()*+,./0123456789:;?ABCDEFGHIJKLMNOPQRS

TUVWXYZabcdefghijklmnopqrstuvwxyz.

It very well may be seen that most of. the time, the characters are anticipated exactly at the position they show up in the picture (for example think about the position of. the "I" in the picture and the diagram). Only the last character "e" isn't adjusted. Be that as it may, this is OK, as the CTC activity is without division and couldn't care less about total positions. From the base most chart demonstrating the scores for the characters "l", "i", "t", "e" and the CTC clear name, the content can easily be decoded: we simply take the most likely character from each time-step, this structures the supposed best way, at that point we toss away rehashed characters and finally all spaces: "l - ii- - t-t- - l-… - e" → "l - I- - t-t- - l-… - e" → "little".hlight all of the contents and import your prepared text file.

IV. ARCHITECTURAL FRAMEWORK

CRNN includes a convolutional highlight extractor that encodes visual subtleties into dormant vectors, trailed by a repetitive succession decoder that transforms the idle vectors into human-justifiable characters. The entire design is prepared start to finish by means of Connectionist Temporal Classification (CTC) misfortune capacity or consideration instrument.

Inside CRNN, the part of the grouping decoder for example LSTM has been accounted for to fill in as a language model. It is seen that OCR model achieves higher exactness for important content line than for arbitrary content line.

CRNN right off the bat consolidated CNN and RNN to remove consecutive visual highlights of a given book picture, and afterward straightforwardly took care of them into a CTC decoder to foresee the best character classification of each time step, where CTC just boosted the likelihood of the relative multitude of ways that can arrive at the ground truth as per the visual grouping of each position.

Improvement over RNN:

LSTM (Long Short-Term Memory) Networks :

To add another data, RNN changes the current data totally by applying a capacity. To beat this restriction, we use LSTM calculation. LSTMs make little adjustments to the data by increases and augmentations. With LSTMs, the data moves through a component known as cell states. Thusly, LSTMs can specifically recall or fail to remember things. The data at a specific cell state has three unique conditions.

These conditions can be summed up as:

1) The past cell state for example the data that was available in the memory after the past time step.

2) The past shrouded state i.e., this is equivalent to the yield of the past cell.

3) The contribution at the current time step for example the new data that is being taken care of in at that point.

Working of LSTM is isolated into 3 stages:

1) The initial phase in LSTM is to choose what data we will discard from the cell state. The choice is made by the entryways.

2) The subsequent stage is to choose what new data we will store in the cell state. This has two sections. Initial, a sigmoid layer called the or the information entryway layer chooses which esteems is to be refreshed. Next, the tanh layer makes a vector of new up-and-comer esteems, that could be added to the state. After this we update the old cell state with the upgraded one.

3) The last advance is to choose the yield. The yield will be founded on our cell state. To begin with, we run a sigmoid layer which chooses what parts of the cell state we will yield. At that point, we put the cell state through tanh and increase it by the yield of the sigmoid door, so we yield the parts that we have chosen to.

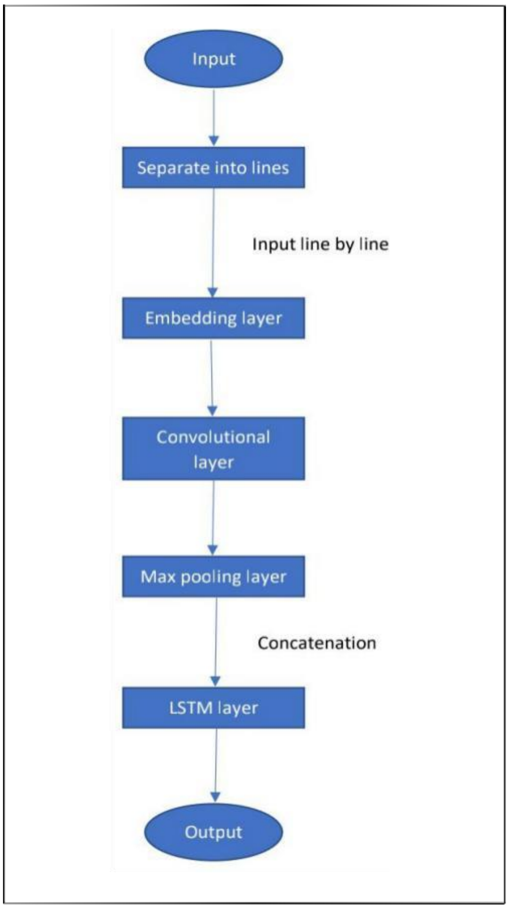


Fig. Flor Architecture

Along these lines, the proposed Gated-CNN-BLSTM engineering jam the low number of.parameters (around.820 thousand) and high acknowledgment rate. It shows in detail the appropriation of.parameters and hyperparameter through 11 convolutional layers (5 gated included) and 2 BLSTM. (Bidirectional Long Short Term Memory) .Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

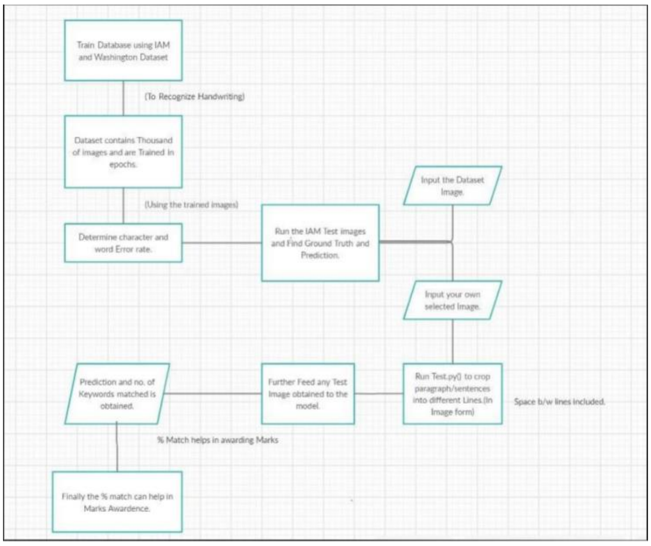

Train Database using IAM and Washington Dataset
(To Recognize Handwriting)
Dataset contains Thousand of images and are Trained in epochs.
Input the Dataset Image.
(Using the trained images)
Determine character and word Error rate.
Run the IAM Test images and Find Ground Truth and Prediction.
Input your own selected Image.
Prediction and no. of Keywords matched is obtained.
Further Feed any Test Image obtained to the model.
Run Test.py() to crop paragraph/sentences into different Lines.(In Image form)
Space b/w lines included.
% Match helps in awarding Marks
Finally the % match can help in Marks Awardence.

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 FOR
END FOR
END IF
Save the cropped image
```

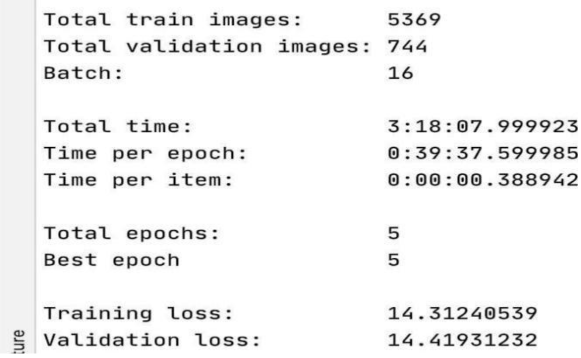
V. EXPERIMENTAL RESULTS

Command to train the dataset : prakhardungarwal@Prakhars-MacBook-Air

source_main % python3 main.py --source iam --arch flor -

-train --epochs 5 Output :

Algorithm



```
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
```

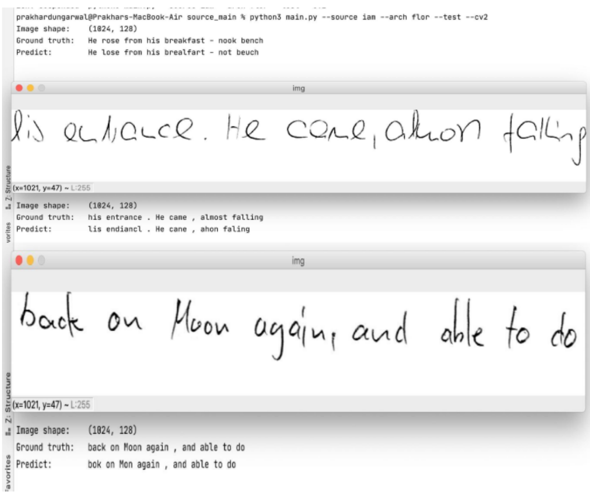
Fig. Training Result

Command to Show trained Dataset Output :

prakhardungarwal@Prakhars-MacBook-Air

source_main % python3 main.py --source iam --arch flor -

-test --cv2 Output :



```
prakhardungarwal@Prakhars-MacBook-Air source_main % python3 main.py --source iam --arch flor --test --cv2
Image shape:    (1024, 128)
Ground truth:   He rose from his breakfast - nook bench
Predict:        He lose from his brealfart - not beuch
```

img
lis enhance. He cone, alhon falling
(x=1021, y=47) ~ L:255

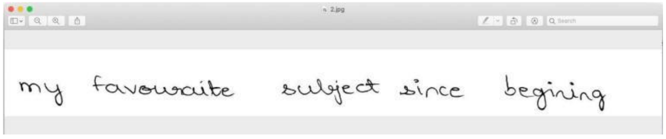
```
Image shape:    (1024, 128)
Ground truth:   his entrance . He came , almost falling
Predict:        lis endiancl . He cane , ahon faling
```

img
back on Moon again, and able to do
(x=1021, y=47) ~ L:255

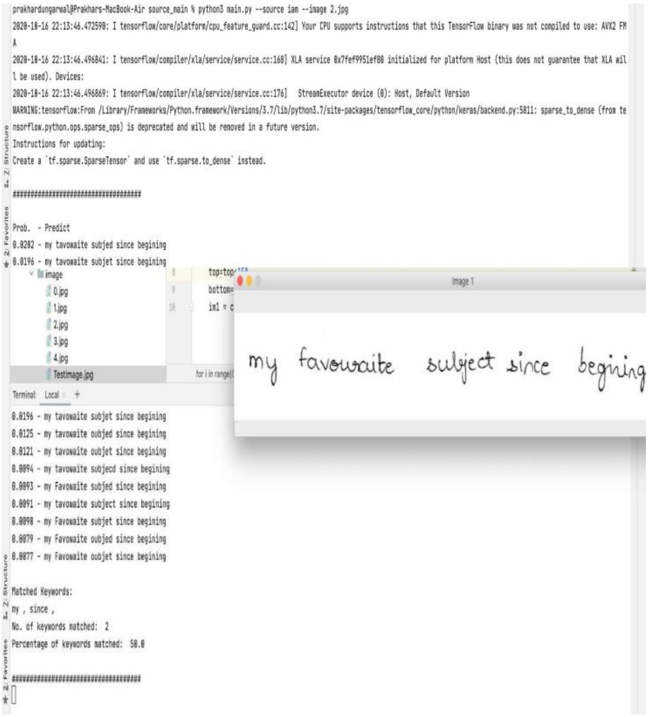
```
Image shape:    (1024, 128)
Ground truth:   back on Moon again , and able to do
Predict:        bok on Mon again , and able to do
```

Fig. Output of IAM Dataset Trained Images

Command to run model for self image : Input Image :



Output :



```
prakhardungarwal@Prakhars-MacBook-Air source_main % python3 main.py --source iam --image 2.jpg
2020-10-16 22:13:46.472590: 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
2020-10-16 22:13:46.496841: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fef9951ef80 initialized for platform Host (this does not guarantee that XLA wil
l be used). Devices:
2020-10-16 22:13:46.496869: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
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.

####################################

Prob.  - Predict
0.0202 - my tavowaite subjed since begining
0.0196 - my tavowaite subjet since begining
```

```
0.0196 - my tavowaite subjet since begining
0.0125 - my tavowaite oubjed since begining
0.0121 - my tavowaite oubjet since begining
0.0094 - my tavowaite subjecd since begining
0.0093 - my Favowaite subjed since begining
0.0091 - my tavowaite subject since begining
0.0090 - my Favowaite subjet since begining
0.0079 - my Favowaite oubjed since begining
0.0077 - my Favowaite oubjet since begining

Matched Keywords:
my , since ,
No. of keywords matched:  2
Percentage of keywords matched:  50.0

####################################
```

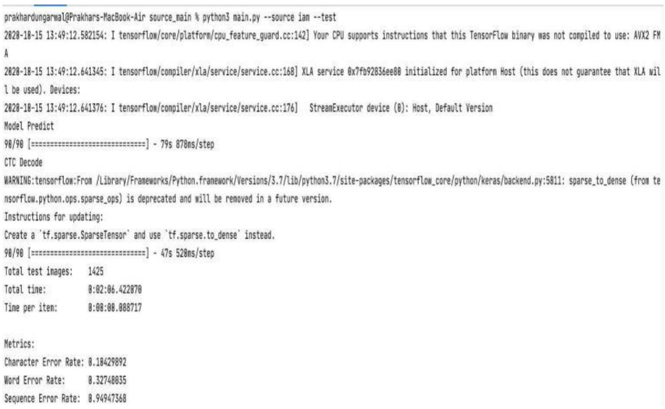
Fig. Output for the Handwritten Self Image

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

No. of Trained Epochs	Character Error Rate	ord 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

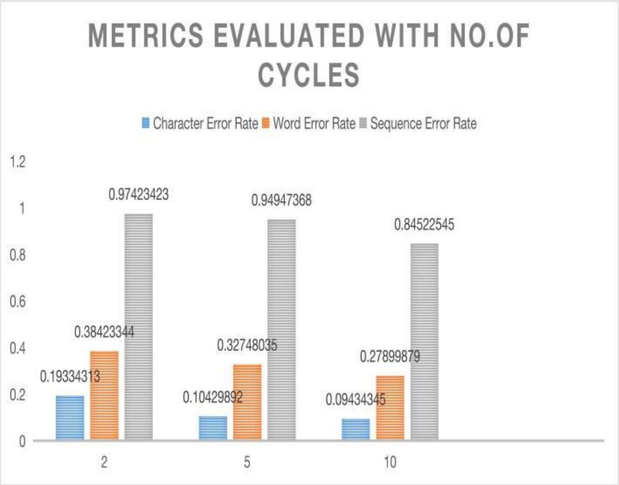


Fig. Comparing Metrics for text recognition rate vs No. Of training cycles or epochs

Table showing Comparision of Various Algorithms

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

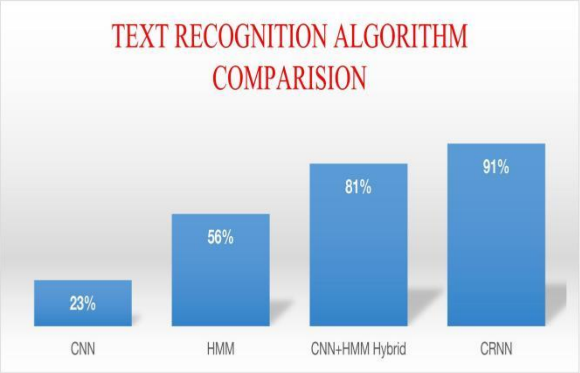


Fig. Comparing Different Algorithms for text Recognition

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. CONCLUSIONS

The proposed evaluation model recommended here can do extraordinary assistance for our schooling framework particularly in future occasions as our schooling frameworks would change these frameworks will help diminish instructor's time in checking enlightening answers and offer them a chance to chip away at more profitable things.

The framework here has ended up being fit for coordinating the catchphrases of the modular answer and granting marks dependent on the level of coordinating of these watchwords. Consequently, the said framework could be of extraordinary utility to the teachers at whatever point they have to step through a fast exam for amendment purpose, as it spares them the difficulty of assessing the heap of papers.

Paper checking techniques will assist with assessing the understudy's answer. Our proposed technique assesses it all the more effectively and precisely.

The proposed model incorporates extraordinary picture preparing and profound learning procedures to help produce the best route for paper checking and text acknowledgment yet what we can enhance this is that the proposed framework isn't equipped for examining an entire archive and afterward perceiving text it just sweeps a page and afterward crops it into text lines to perceive if this can additionally be streamlined to make all in all report scanner and auto-evaluator it could end up being an incredible transformation in the paper checking technique and can spare a huge number of long periods of instructors.

VIII. REFERENCES

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

[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.

[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.

[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.

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

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

[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.

[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).

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

[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).

[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.

[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.

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

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

[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.