Hybrid networks for handwritten word recognition in JPEG compressed domain

Instructor: Dr. Mohammed Javed

Members:

1. Moksh Grover(IIT2018186)
2. Abhishek kumar Gupta (IIT2018187)
3. Ayush Raj (IIT2018188)
4. Utkarsh Priyam (IIT2018197)

Introduction

Handwritten Text Recognition (HTR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices.

Scope

It has a wide variety of applications which include:

- Reading postal cheque addresses
- Reading cheque amounts
- Data extraction from forms
- Handwritten biometric recognition

Problem Statement And Objectives

In the domain of Handwritten Word recognition, recently it was shown that a hybrid scheme of using convolutional recurrent architecture where the convolutional layers are meant for feature extraction, which are subsequently given to a RNN network along with CTC loss, work better than other schemes,but main pitfall is the high computational cost of neural networks, and there is great potential for reducing overall computational cost, since there is less data in the compressed domain than in the original uncompressed domain. So, our main objective is train the CNN-RNN hybrid network on compressed images.

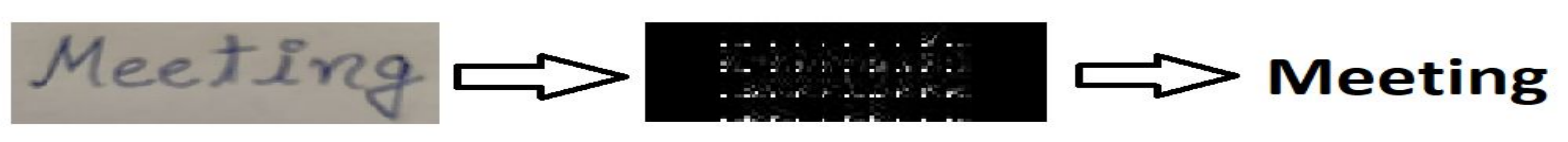


Fig. 1

Literature Review

A lot of papers were included in the research comprising various methodologies mixed with various advantages and disadvantages :

Paper [1] used an ANN trained with Resilient Back Propagation and scaled conjugate gradient to get a final accuracy score of 95% on manually created dataset. Various steps in preprocessing included the tilting of images followed by segmentation and feature extraction. As the method suggested fewer features it took slightly less time for computation.

In [2] an approach was made using a CNN- RNN along with a CTC layer along with a spell checker to suggest possible options thus achieving an accuracy of 90.3% on the IAM dataset.

Literature Review

In [3] a modified CNN-RNN hybrid architecture with a major focus on effective training using: efficient initialization of the network using synthetic data for pre-training,image normalization for slant correction and domain specific data transformation and distortion for learning important invariances.

In [4] they used a 1D LSTM followed by CTC training for the final layer where the input to the 1D LSTM consisted of geometrically normalized text lines. Hence, it was concluded that a combination of convolutional layers, max pooling, and 1D LSTM can result in substantially lower error rates than the simple 1D LSTM networks.

Literature Review

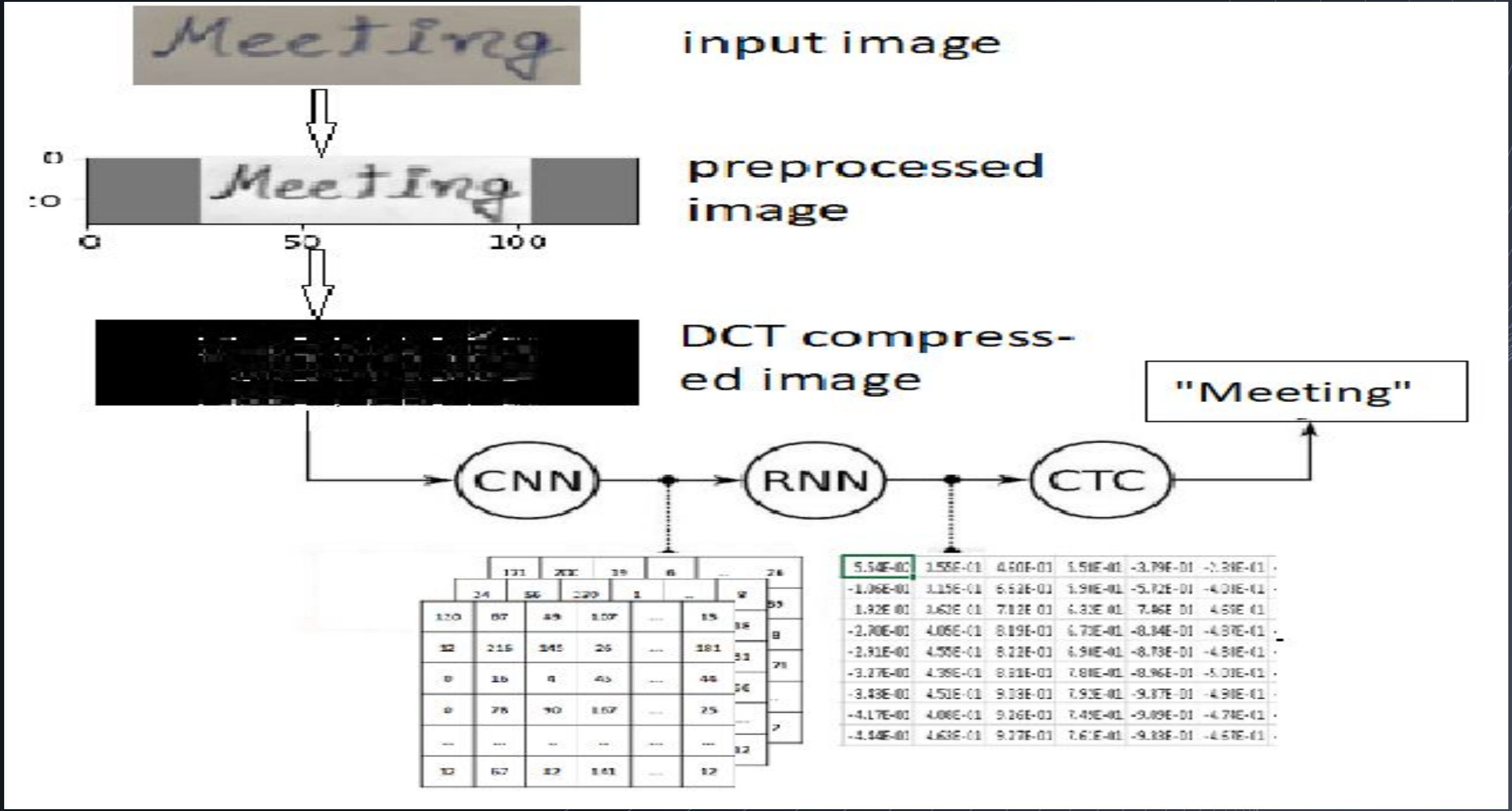
In [5] a new approach was discussed which incorporated word beam search decoding, which constrains words to those contained in a dictionary, allows arbitrary non-word character strings between words and optionally integrates a word-level language model and has a better running time than token passing. For the proposed WBS a prefix tree was used to query the characters.

In [6] a brief overview of various methodologies was done for the purpose of content based image retrieval uncompressed domain, all on the MPEG dataset and hence concluded that CBIR possesses approximate 15% less computational complexity in compressed image domain as compared to normal images.

Proposed Methodology

A. Model Overview:

Fig. 2

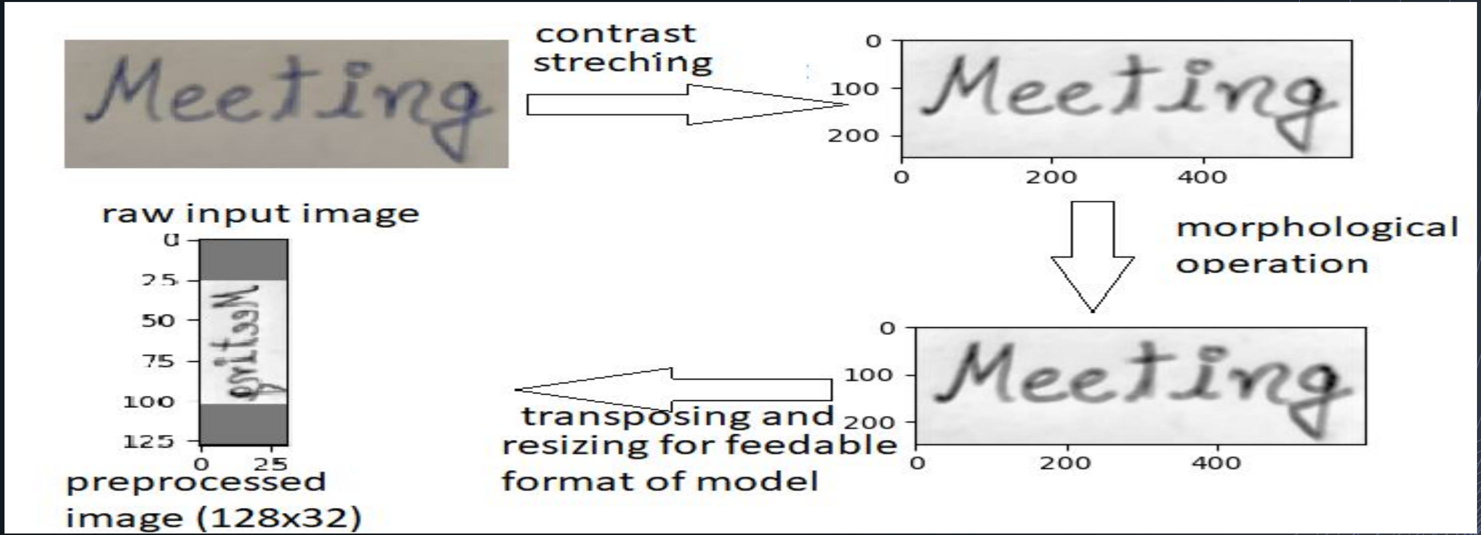


120	87	49	107	...	15
12	215	145	26	...	181
0	16	4	45	...	44
0	78	90	167	...	25
...	...	...	...	...	...
12	67	12	141	...	12

5.54E-01	1.55E-01	4.60E-01	5.51E-01	-3.79E-01	-2.38E-01	·
-1.06E-01	3.15E-01	6.62E-01	5.90E-01	-5.72E-01	-4.28E-01	·
1.92E 01	3.62E 01	7.12E 01	6.33E 01	7.46E 01	4.59E 01	
-2.70E-01	4.05E-01	8.19E-01	6.73E-01	-8.34E-01	-4.37E-01	·
-2.91E-01	4.55E-01	8.22E-01	6.90E-01	-8.73E-01	-4.38E-01	·
-3.27E-01	4.39E-01	8.31E-01	7.80E-01	-8.96E-01	-5.00E-01	·
-3.43E-01	4.51E-01	9.13E-01	7.93E-01	-9.87E-01	-4.90E-01	·
-4.17E-01	4.08E-01	9.26E-01	7.45E-01	-9.09E-01	-4.74E-01	·
-4.44E-01	4.63E-01	9.27E-01	7.61E-01	-9.33E-01	-4.67E-01	·

B. pre-processing [7]

Fig. 3



C. Image compression

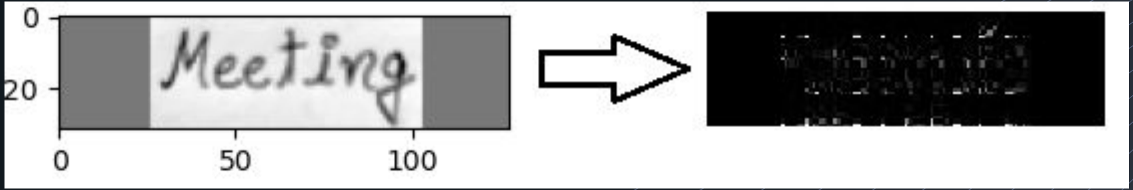


Fig. 4

D. CNN (multiscale feature extraction)

For each CNN layer, create a kernel of size k×k to be utilized in the convolution operation. Then, RELU operation again to the pooling layer with size px×py and step-size sx×sy with results of the convolution. These steps are repeated for all layers during a for-loop.
The convolution operation, 5×5 filter is used in the first two layers and 3×3 filter used in the last three layers to the input.

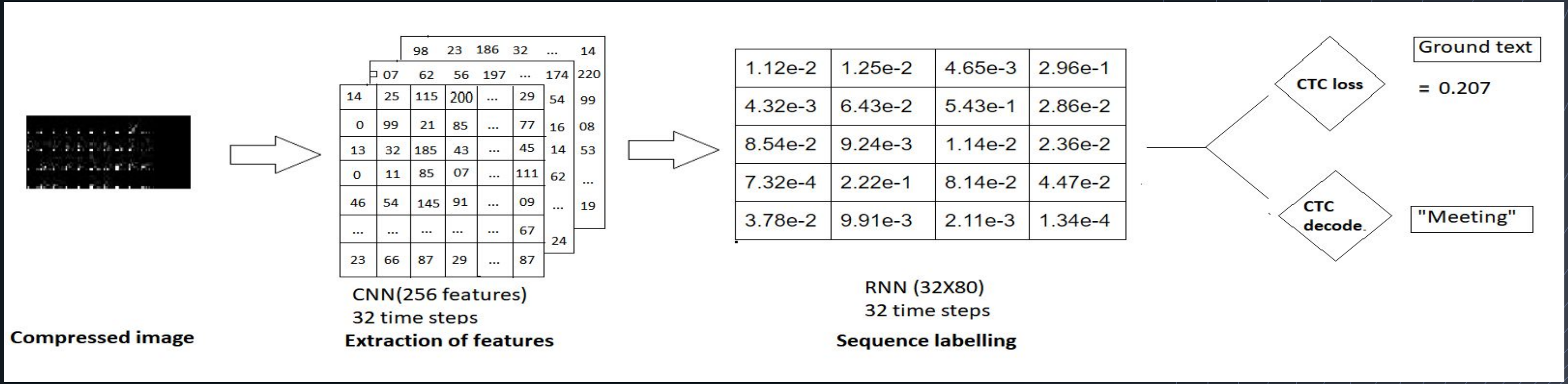
F. RNN (Sequence Labeling) (BLSTM-CTC)) [4]

The feature sequence consists of 256 features per time-step, the RNN propagates forward and backward layer relevant information through this sequence.

The RNN output sequence is mapped to a matrix of 32×80. The IAM dataset contains 79 different characters, further one additional character is required for the CTC operation (CTC blank label), so that there are 80 entries for every of the 32 time-steps.

G. CTC decoding and word beam search (Transcription) [5]

How CTC decoding work?

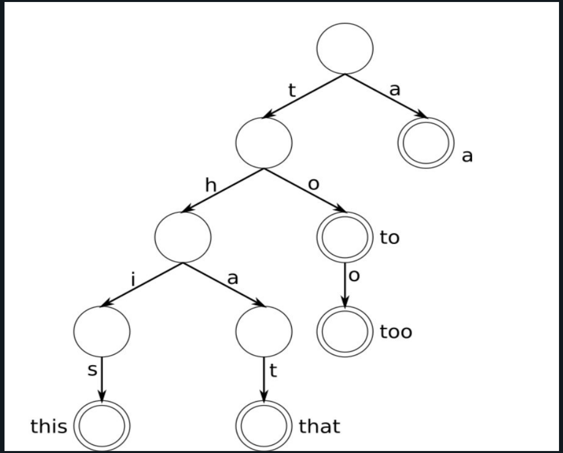


14	25	115	200	...	29
0	99	21	85	...	77
13	32	185	43	...	45
0	11	85	07	...	111
46	54	145	91	...	09
...	...	...	...	...	67
23	66	87	29	...	87

1.12e-2	1.25e-2	4.65e-3	2.96e-1
4.32e-3	6.43e-2	5.43e-1	2.86e-2
8.54e-2	9.24e-3	1.14e-2	2.36e-2
7.32e-4	2.22e-1	8.14e-2	4.47e-2
3.78e-2	9.91e-3	2.11e-3	1.34e-4

Fig. 5

CNN RNN output flow



How our CTC decoder works?
When adding a word like “too”, we start at the root node, add (if it does not yet exist) an edge labeled with the first character “t” and a node, then add an edge “o” to this new node, and again add an “o”. The final node gets marked to indicate the end of a word (double circles). If we repeat this for the words “a”, “to”, “too”, “this” and “that”, we get the tree shown in Fig.

Fig. 6

Model Architecture

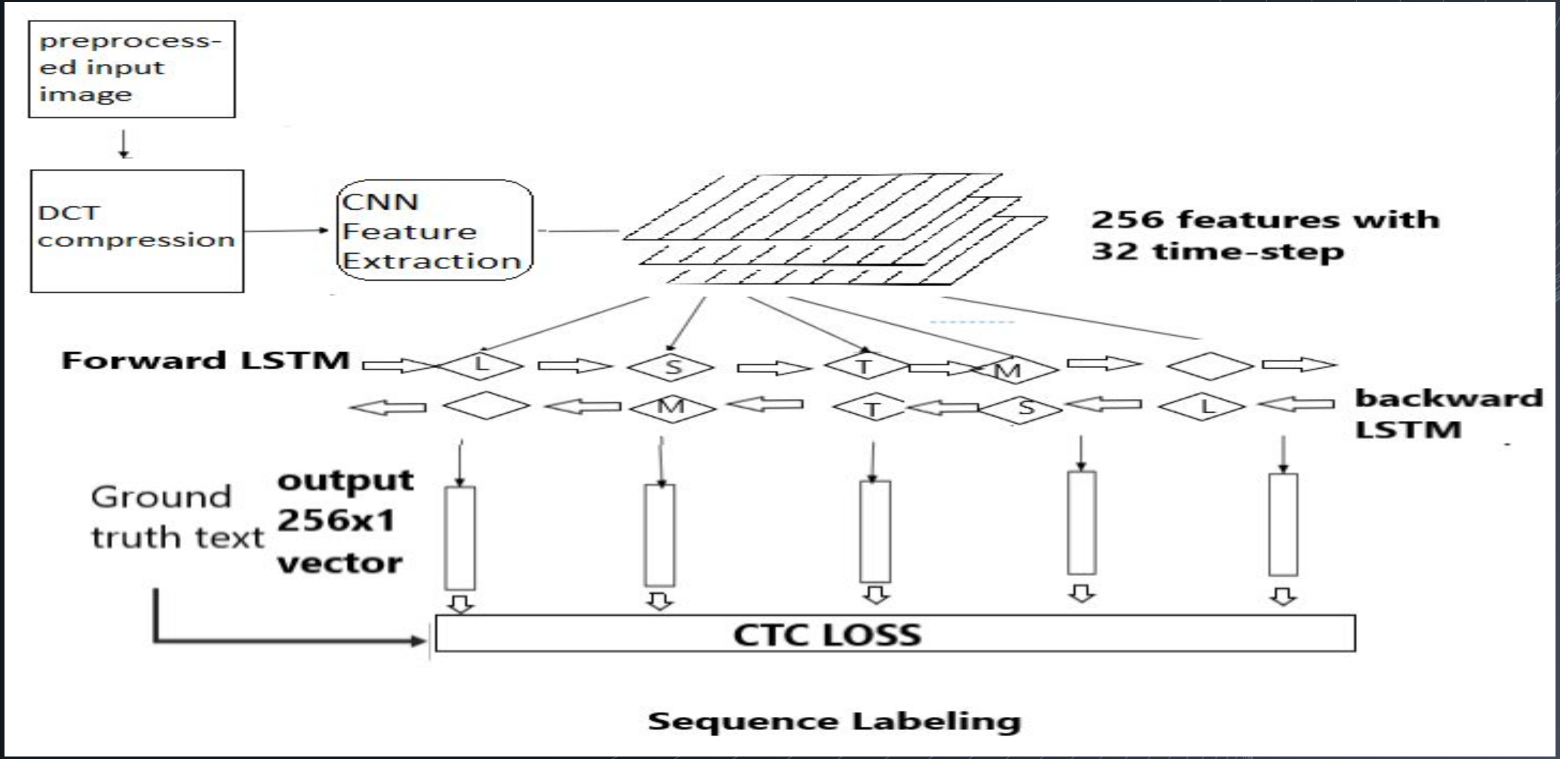


Fig. 7

Libraries used

- Tensorflow
- lmdb
- Numpy
- Pillow
- openCV
- Random
- Pickle
- EditDistance

Dataset Description

The dataset used is the IAM Handwriting Dataset[6] , which has images having text written in English language.

The characteristics of the IAM Handwriting Database are as follows: 657 writers contributed samples of their handwriting , 1539 pages of scanned text , 5685 isolated and labeled sentences ,13353 isolated and labeled text lines, 115,320 isolated and labeled words.The words have been extracted from pages of scanned text using an automatic segmentation scheme and were verified manually. The segmentation scheme has been developed by us.

The dataset was first published in 1999 by the International Conference on Document Analysis and Recognition(ICDAR).

Comparative Evaluation

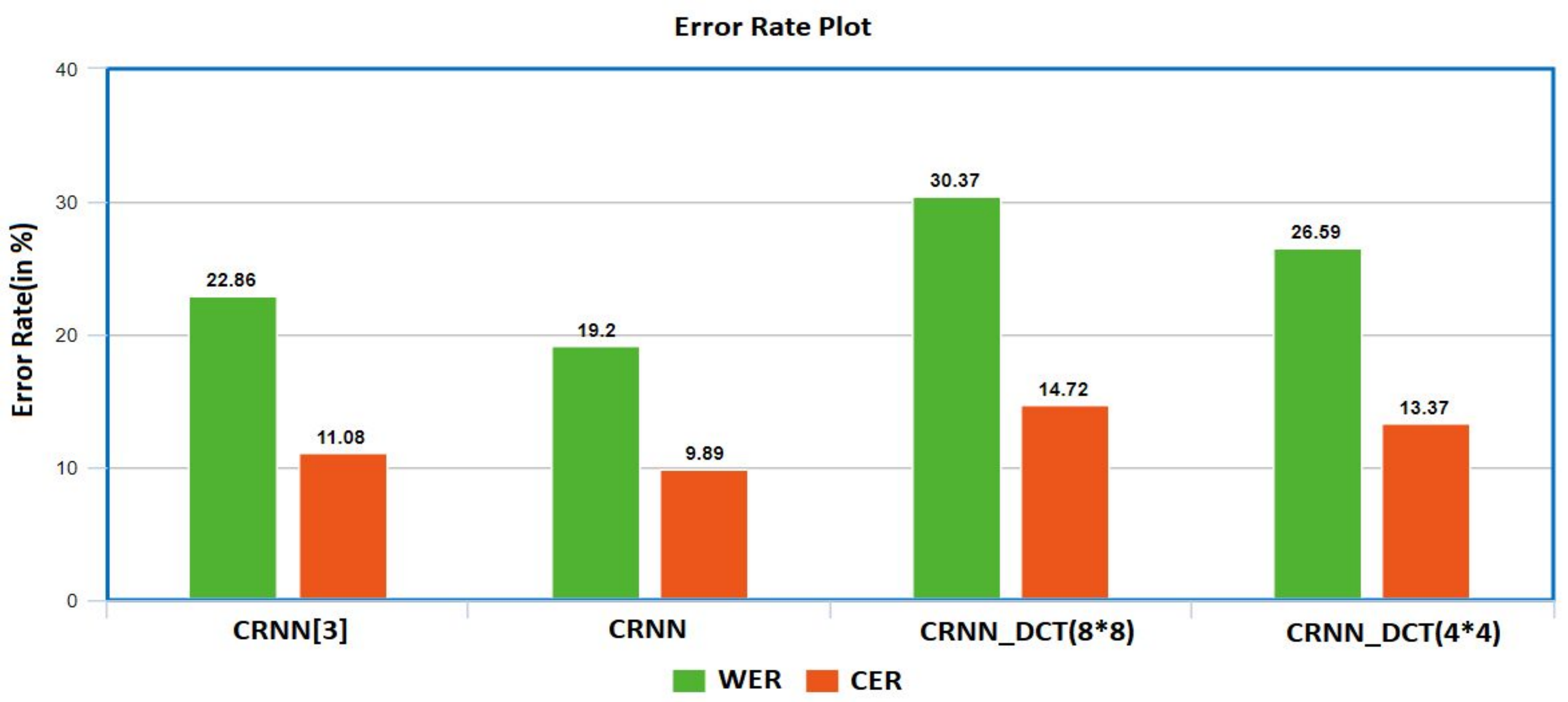
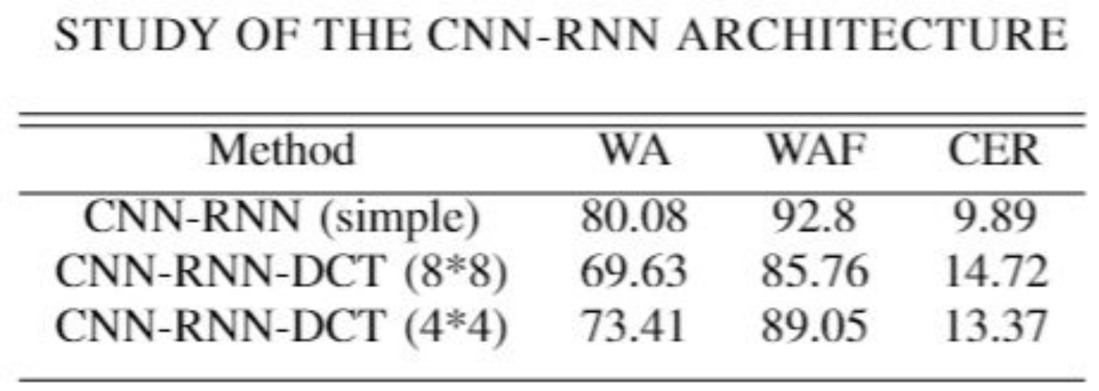


Fig. 8

Result

A detailed study of papers was done to understand the concepts of image compression and a brief study of papers also helped in selecting the hybrid network (CNN-RNN) to achieve our goal. We have done our experiment without using compressed domain images and then extended the work to compressed domain with block 8x8 and 4x4 kernel as shown in Table 1.



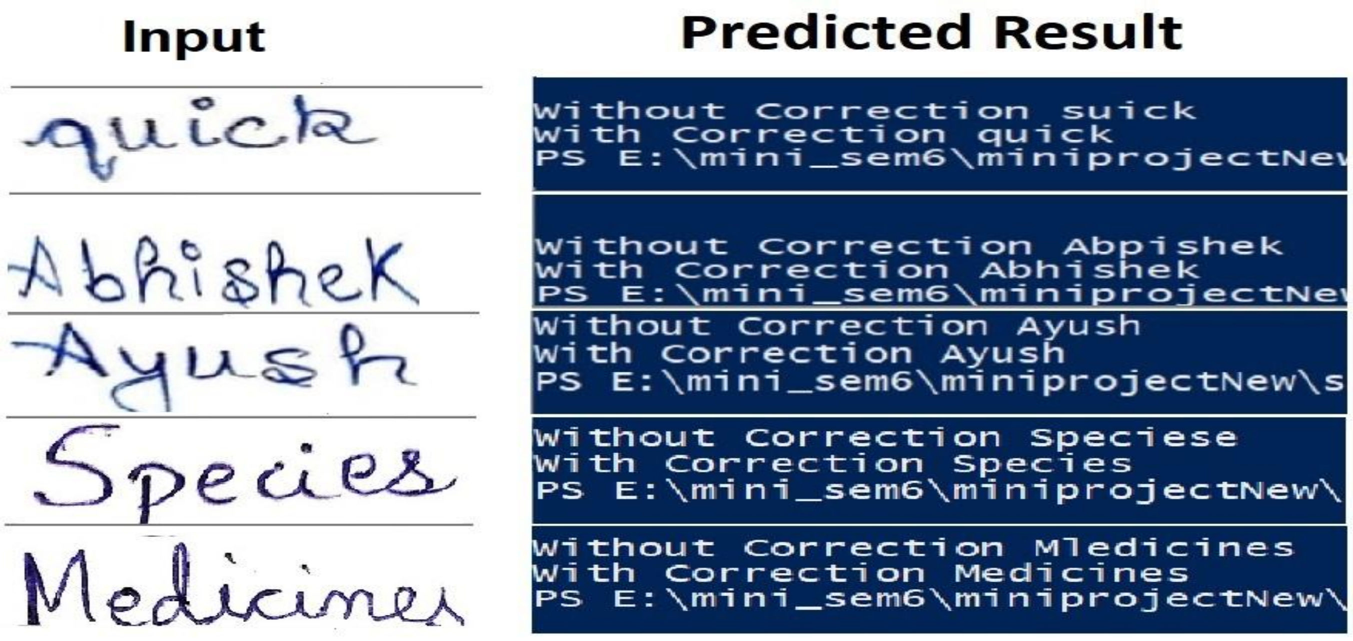
STUDY OF THE CNN-RNN ARCHITECTURE

Method	WA	WAF	CER
CNN-RNN (simple)	80.08	92.8	9.89
CNN-RNN-DCT (8*8)	69.63	85.76	14.72
CNN-RNN-DCT (4*4)	73.41	89.05	13.37

Fig. 9

Result

Predicted result for some self handwritten images.



```
Without Correction suick
With Correction quick
PS E:\mini_sem6\miniprojectNe
```

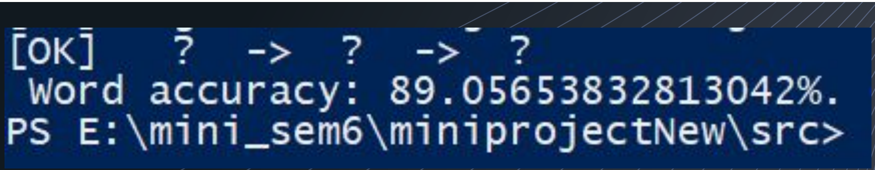
```
Without Correction Abpishek
With Correction Abhishek
PS E:\mini_sem6\miniprojectNe
```

```
Without Correction Ayush
With Correction Ayush
PS E:\mini_sem6\miniprojectNew\s
```

```
Without Correction Speciese
With Correction Species
PS E:\mini_sem6\miniprojectNew\
```

```
Without Correction Mledicines
With Correction Medicines
PS E:\mini_sem6\miniprojectNew\
```

Fig. 10



```
[OK]   ?  ->  ?  ->  ?
 Word accuracy: 89.05653832813042%.
PS E:\mini_sem6\miniprojectNew\src>
```

Fig. 11

Conclusion

In this work, we presented a CNN-RNN hybrid architecture trained on images that are in DCT compressed domain.The accuracy was achieved as expected and the computational time required decreased when compared to HWR without DCT compression thus saving a whole lot of valuable time.When DCT compression was applied by using (4*4) lesser collision of blocks containing letters occur as compared to the DCT compression using (8*8) kernel, hence an increase in accuracy was seen when a kernel size of (4*4) was considered.

References

[1] Obaid, A. M., et al. "Handwritten text recognition system based on neural network." Int. J. Adv. Res. Comput. Sci. Technol.(IJARCST) 4.1 (2016): 72-77.

[2] Manchala, Sri Yugandhar, et al. "Handwritten text recognition using deep learning with Tensorflow." International Journal of Engineering and Technical Research 9.5 (2020).

[3] Dutta, Kartik, et al. "Improving cnn-rnn hybrid networks for handwriting recognition." 2018 16th international conference on frontiers in handwriting recognition (ICFHR). IEEE, 2018.

[4] Breuel, Thomas M. "High performance text recognition using a hybrid convolutional-lstm implementation." 2017 14th IAPR international conference on document analysis and recognition (ICDAR). Vol. 1. IEEE, 2017.

[5] D. Edmundson and G. Schaefer, "An overview and evaluation of JPEG compressed domain retrieval techniques," Proceedings ELMAR-2012, Zadar, Croatia, 2012, pp. 75-78.

[6] U. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In Proc. of the 5th Int. Conf. on Document Analysis and Recognition, pages 705 - 708, 1999.

Thank You

End of slides