INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

DEPARTMENT OF ELECTRICAL ENGINEERING

INTERNSHIP REPORT

ANDROID APPLICATION FOR TELUGU HANDWRITTEN WORD RECOGNITION

Author

Mohammad Al Fahim K

Supervisor

Dr. Srinivas Padmanabhuni

15th May, 2020 – 15th July, 2020

Abstract

This is a report of my summer Internship at Tarah AI, Bangalore. I worked in the Android Application development program under the supervision of Dr Srinivas Padmanabhuni. I was given the task of developing an Android application for a research project, “Telugu Handwritten word Recognition”, I was working on during my fifth semester.

The app is designed to take a Telugu handwritten text document as an input image, irrespective of the image taken using a camera or imported from the mobile’s gallery, to recognise the words in the document and finally print the recognised text in a word file in the same text format as in the input image. The app was designed using the Flutter development kit software and Dart programming language.

Acknowledgement

I would like to convey my deepest thanks to Tarah AI for providing me with this marvellous opportunity of interning in their company.

I would like to express my deepest gratitude to Dr Srinivas Padmanabhuni for sparing his time and energy for helping and clarifying my doubts. He provided me pointers and new ideas to try out to help improve my project. He taught me the industrial demands of today’s world and showed me how much Artificial Intelligence technology has progressed. I cannot thank him enough and will always be grateful for his contribution.

I consider this opportunity to be a major milestone in my career development. The skills and knowledge I have learned during this period will be very productive and I will always keep striving to enrich and enhance my knowledge and expertise in this domain.

Yours Sincerely,

Mohammad Al Fahim K

About Industry

Tarah AI is a boutique technology enabled business consulting and competency development company. It helps medium, small scaled enterprises and start-ups navigate the digital landscape via services in Big Data analytics, Machine Learning, IOT analytics and Digital Marketing, alongside mobile solution development.

TARAH Technologies

INTERNSHIP CERTIFICATE

This letter is to confirm the completion of online internship at Tarah technologies by Mohammed Al Fahim K, (EE17B021) a third year EE student of IIT Tirupati. His online summer internship started on 15th May 2020 and ended on 15th July 2020.

During the internship Mohammed worked on the problem of recognizing telugu characters with a RCNN architecture followed by development of a mobile app to recognize character image given as input to them.

He displayed a strong skill for learnability and openness to pick up complex new subjects. His drive to get things done even in tasks like app development the task is commendable. His work is worth publishing in a research forum.

We wish him all the luck in his future projects.

Dr Srinivas
Padmanabhuni,
CTO/Chief Mentor
Tarah Technologies
(Tarah.AI)
http://www.tarahtech.com
srinivas@tarahtech.com
+91 9845116391

16th July 2020
BANGALORE.

Contents

List of Figures

List of Tables

List of Abbreviations

1. Artificial Intelligence……AI
2. Optical Character Recognition……OCR
3. Convolutional Recurrent Neural Networks……CRNN
4. Maximally Stable Extremal Regions……MSER
5. Bidirectional Long Short Term Memory……Bi-LSTM
6. Connectionist Temporal Classification……CTC
7. Hyper Text Transfer Protocol……HTTP
8. Spatial Transformer Network……STN

1. Introduction

In the wake of the digital era and Artificial Intelligence revolution, the need to adapt and enforce AI into all domains and sectors of industry and research has become crucial. Digitalizing sounds, images and all sorts of signals and information has become the trend of today's world and with the rise of digital libraries, it has become imperative to preserve manuscripts, documents and all other kinds of hand written and typed documents for the future generation as they are on the brink of extinction because of the degrading quality of paper over years of usage and rough handling. Character recognition also makes it viable to translate text from one language to another and to also convert them into speech, making it easy for bind people to read by listening.

Convolutional neural networks provide a way to recognise characters in documents and thus arose Optical Character Recognition, which has proved to be very successful in digitalizing texts of English and other European languages. OCR excels in recognising printed characters and applying the same on handwritten characters has not shown similar results. This is due to the fact that handwritten words consist of overlapping and intertwined characters which are very difficult to segment. They also have different fonts and sizes which vary with every person's handwriting.

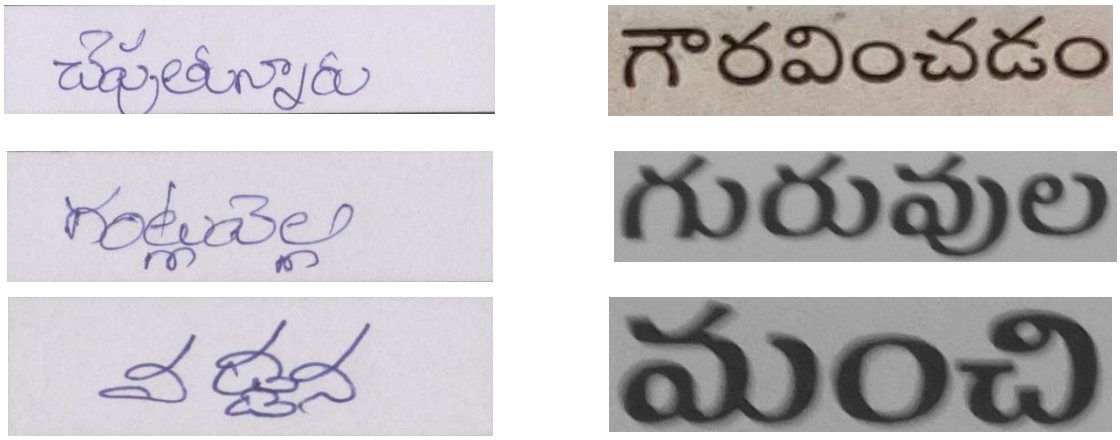

Fig.1 Examples of word images, handwritten words to the left and printed words to the right

To tackle this, a combination of convolution and recurrent neural networks are used, wherein the input images are whole words rather than individual characters. When trained on a large dataset of words written by various writers with contrasting fonts, they deliver astonishing results. Keep in mind that the level of results observed in European languages cannot be expected in Indic and other Asian languages due to the nature of joining characters in them and their various curves and loops.

The present work focuses on recognising handwritten Telugu documents by first segmenting the words in the image, pre-processing them to the required size, recognising the words and finally printing them out in the same format as they are in the document image. All of this was encapsulated in a mobile app for easy consumer usability.

2. Word Segmentation

First and foremost in word recognition is to process each word one by one as all words in the image cannot be recognised all at once. To do so, the words have to be segmented, subsequently pre-processed and sent to the CRNN model for recognition. The segmentation algorithm used in this work is borrowed from IIT Hyderabad's paper on optical character recognition [1]. They have used an altered MSER algorithm and processed its results to segment the words. Image skew correction is done prior to segmenting them using the de-skew python library.

MSER algorithm is used a method of blob detection in images. It extracts a comprehensive number of image elements which has led to better object detection algorithms.

The following figure may be sufficient enough to explain how the word segmentation algorithm works.

```
Regions = MSER(Image);
FilteredRegions = [ ] ;
for R ∀ Regions do
    e = height(R)/width(R);
    aC = area(R)/boundingBoxArea(R);
    if (e>0.1) and (e<10) and (aC>0.2) then
        FilteredRegions.append(R);
    end
end
Hist = hist(boundingBoxArea(FilteredRegions));
avgArea = averageArea(modeBin(Hist));
sideLength = sqrt(avgArea);
finImx = dilatex(RegImage,0.7*sideLength);
finalImage = dilatey(finImx,0.2*sideLength);
boxes = boundingBox(contours(finalImage));
```

Fig 2. Word segmentation algorithm, courtesy [1]

The MSER results are dilated so as to retain the dheergams and vattus of a word from being segmented separately. The variables for the extent of dilation are calculated approximately from the distribution of character sizes in the image. The amounts of dilation are different in the x and y directions so that the dheergams and vattus do not merge with neighbouring words and the above and below lines.

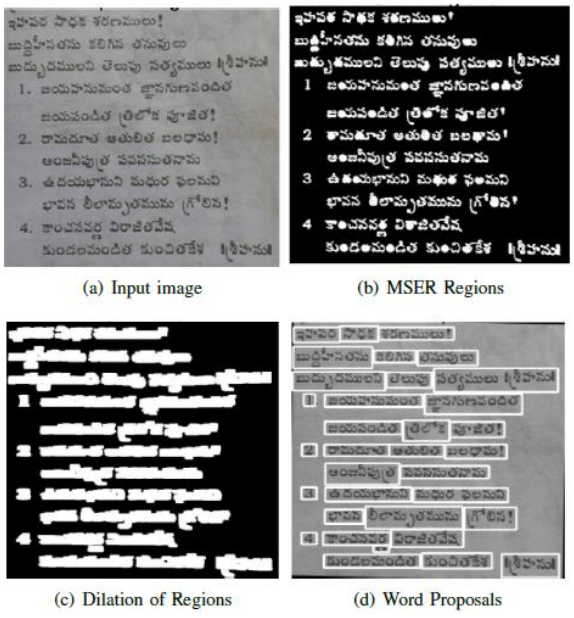

Fig 3. Results of each step of word segmentation, courtesy [1]

3. Word Image Processing

Since the CRNN model in this work requires input images to be of size 32 x 128, they have to be resized, passed through Otsu-thresholding and normalised. By simply resizing them, the characters seem to break and a lot of pixel information is lost. To solve this issue, this work utilizes an altered method of Otsu-threshloding.

Fig 4. Images with broken characters.

The word images are first resized to 32 x 128 size and subsequently they are thresholded using the following threshold algorithm.

$$Z = T + x * f \qquad \text{where } 0 \le x < 1$$

$$x = \frac{\left[\int_T^{255} G(\mathrm{i})\,\mathrm{di} - \int_0^T G(\mathrm{i})\,\mathrm{di}\right]}{\int_O^{255} G(\mathrm{i})\,\mathrm{di}}$$

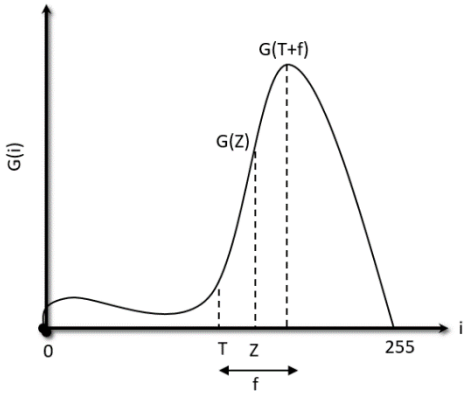

Fig 5. Histogram of image intensities

In the above formulae, T represents Otsu's threshold intensity, f is the distance from intensity with maximum number of pixels to T and x is a fraction. Z is the optimum threshold intensity by which the characters are neither too thick nor too thin that they seem to be broken. The images are thresholded with Z as threshold intensity. Note that for representational purposes, the histogram is shown to be a continuous curve but in reality, it is a discrete curve.

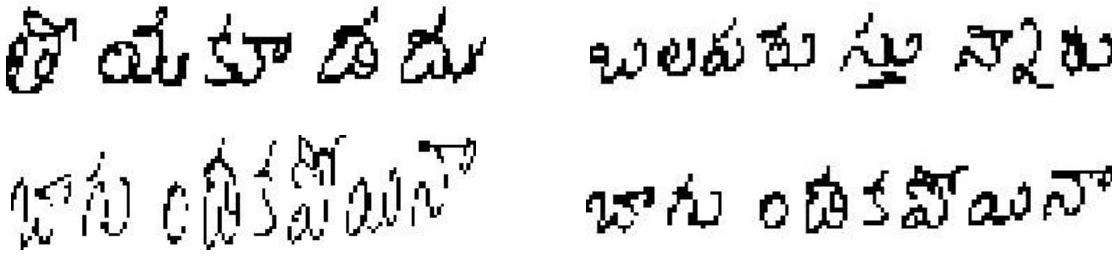

Fig 6. First and second images from fig.4 obtained using proposed thresholding algorithm, third and fourth is a comparison of results

As you can see in Fig.6, the characters are not too thick to be covering up the holes and loops in the letters.

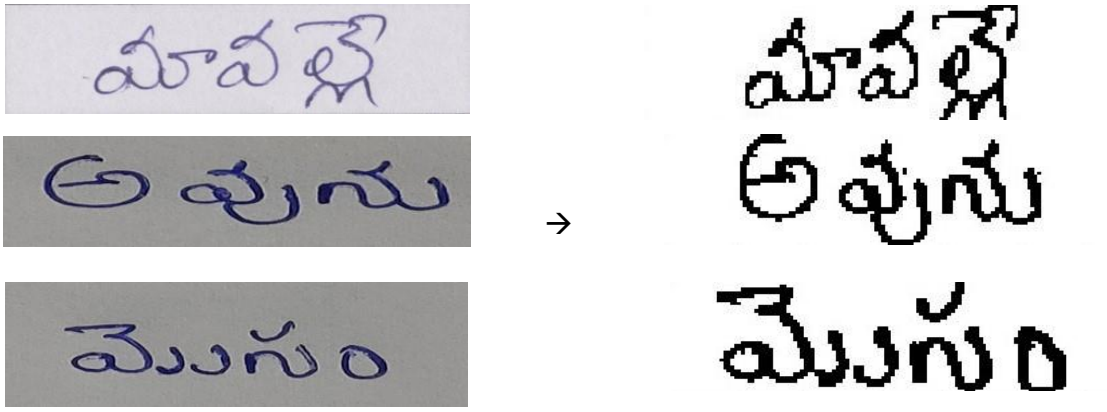

Fig 7. First column contains original images, second column contains results of proposed thresholding algorithm.

4. CRNN Model

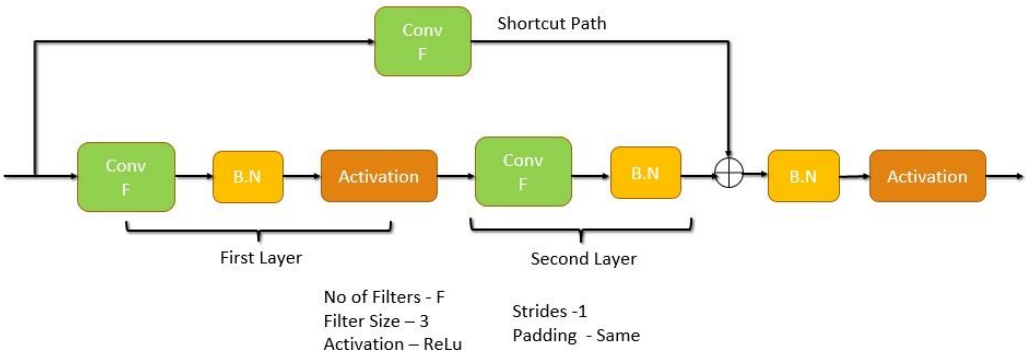

Fig 8. Residual block

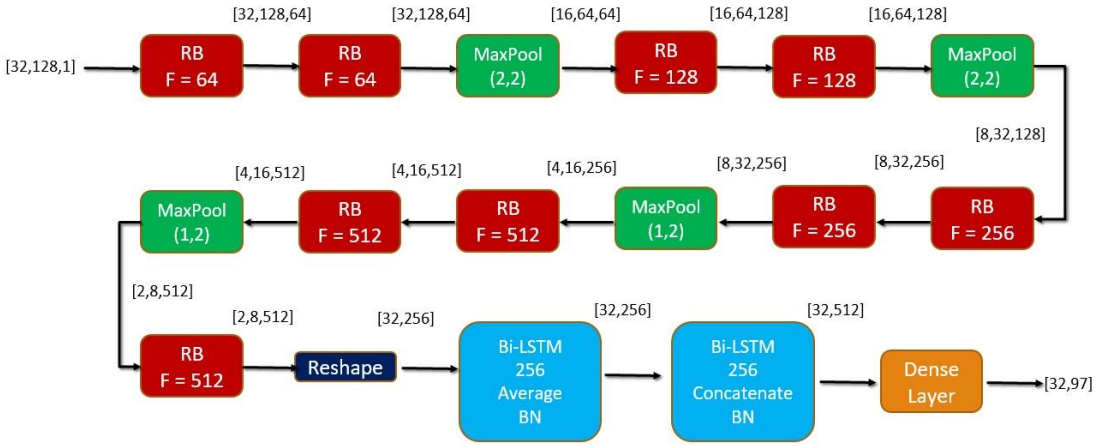

Fig 9. CRNN model architecture

The model architecture is adapted from B.Shi's paper on scene text recognition [2]. Every convolutional layer from their architecture is replaced here with a residual block, shown in fig.8. Max-Pooling layers of window size 1 x 2 are used to acquire more information along the width. This is because information about characters in contact with neighbouring characters can be obtained more across the width rather than the

height. Residual blocks helps in converging while training the model and Bi-LSTMs, each having 256 units, allow to learn the sequence of letters from the feature maps generated by the previous residual blocks.

The training dataset used here is from IIIT Hyderabad's paper on Indic handwritten word recognition [3]. It consists of 118,515 Telugu word images and their respective labels encoded in UTF-8. They are split into training, validation and test datasets roughly in 70:15:15 ratio. The dataset also contains a lexicon file containing approximately 13,000 words for lexicon based decoding.

The model was trained with CTC [4] loss function and ADADELTA [5] optimiser. CTC converts the predictions generated by the recurrent layers as a maximum probable sequence for the input. ADADELTA does not need any manual setting of parameters and helps in faster convergence. Beam search decoding [6] was used with a beam width of 3 and finally the result with maximum probability was saved.

5. Mobile Application

The application was designed in Flutter and Dart programming language and tested in Android Studio.

a) Homepage

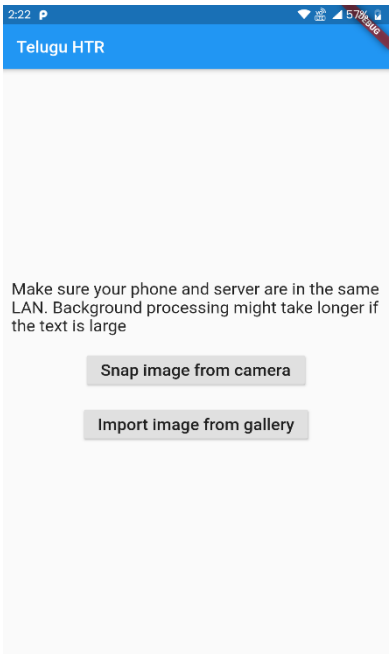

Fig 10. Homepage

The user will be able to either snap an image using the camera or use an image from their gallery.

b) Crop Image

The user can crop and resize the image to their liking using the four corner dots. On pressing the crop button, the image will be cropped and displayed in the next page.

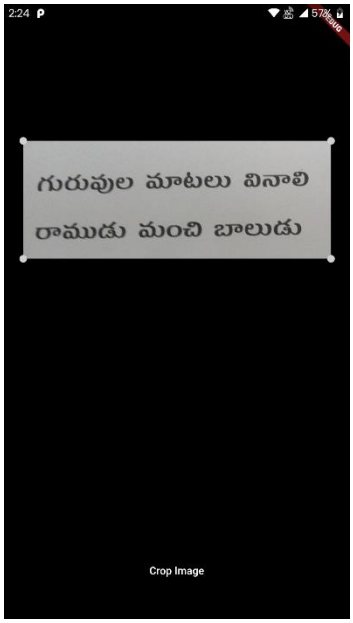

Fig 11. Crop image page

c) Upload Image

The model was uploaded and run in a cloud server and the image was uploaded to it by the HTTP POST method. In this page, the user can view the cropped image and on pressing the up arrow button, the image will be sent to the model for recognition.

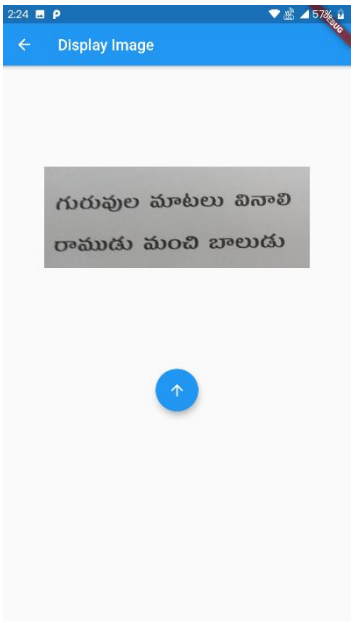

Fig 12. Upload page

d) Result page

The results will be presented to the user and they have the option to save it in a text file by pressing the save button.

Fig 13. Result page

6. Results

The model was trained for 100 epochs and maximum accuracy was found in the 83rd epoch. The 83rd epoch model was then trained on the augmented dataset for 50 epochs and it achieved the highest accuracy in the 33rd epoch.

Accuracies	This work's model	IIIT Hyderabad's results [3]
Character level accuracy	95.45%	95.42%
Word level accuracy	72.61%	76.01%
Lexicon based word accuracy	96.52%	98.93%

Table 1. Comparison of model accuracies

Even though, this work's model was not able to achieve better results than IIIT Hyderabad's work, there is roughly only 3% difference in both word level and lexicon based word accuracies. Their model uses a STN layer before the convolutional layers which detects the region of interest and spatially transforms that for spatial invariance, whereas using the same in this model, seemed to reduce the accuracy and performance. This can be attributed to cropping the dataset images so as to remove the white spaces around the word for retaining the word information better.

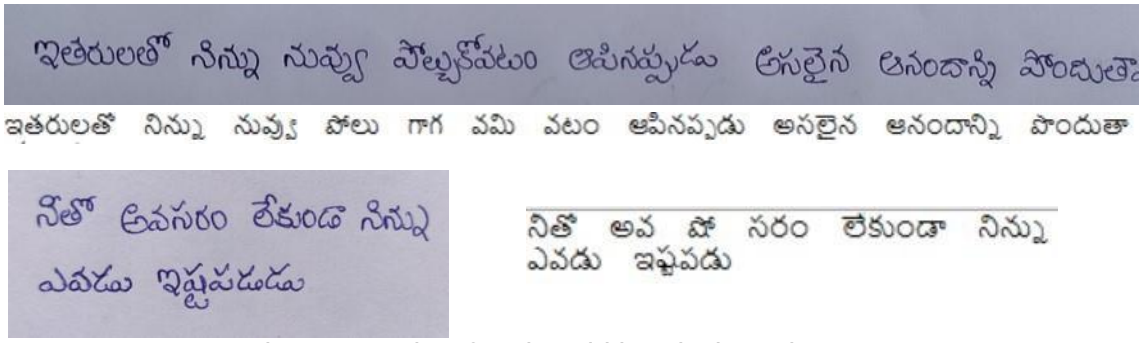

Fig 14. Results of real world handwritten documents

Even though, the model was trained on handwritten word images, it can be extended to recognise printed documents

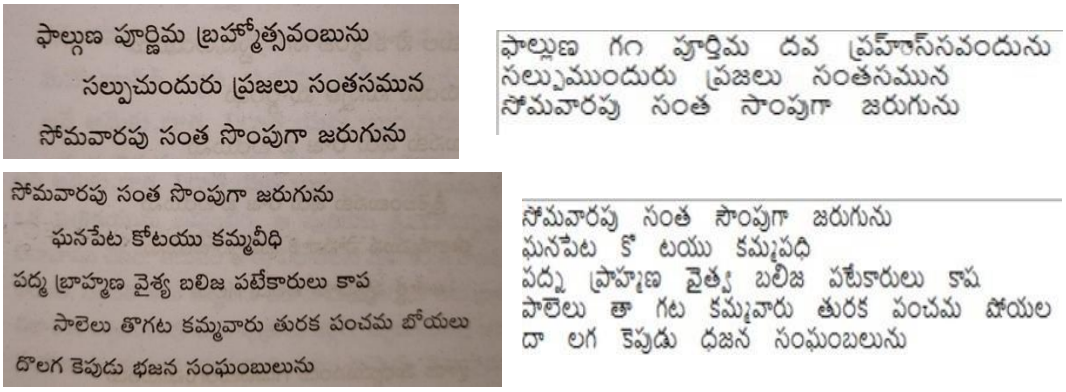

Fig 15. Results of printed documents

The word segmentation sometimes, seemed to fail and it segmented words into individual characters.

7. Conclusion

Since the model was not trained on roman numerals, numbers and punctuation marks, they will be recognised as some arbitrary letter. The word segmentation fails in some cases and the model's accuracies can be improved for reliable usage.

Regardless of its inadequacies, the whole pipeline of segmentation and recognition has performed notably well in most cases.

8. References

[1] Chandra Prakash, Konkimalla & Srikar, Y. & Trishal, Gayam & Mandal, Souraj & Channappayya, Sumohana. (2017). Optical Character Recognition (OCR) for Telugu: Database, Algorithm and Application.

[2] Shi, Baoguang & Bai, Xiang & Yao, Cong. (2015). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 10.1109/TPAMI.2016.2646371.

[3] Dutta, Kartik & Krishnan, Praveen & Mathew, Minesh & Jawahar, C.V.. (2018). Towards Spotting and Recognition of Handwritten Words in Indic Scripts. 32-37. 10.1109/ICFHR-2018.2018.00015.

[4] A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhu-´ ber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In ICML, 2006.

[5] M. D. Zeiler. ADADELTA: an adaptive learning rate method. CoRR, abs/1212.5701, 2012.

[6] Scheidl, Harald & Fiel, Stefan & Sablatnig, Robert. (2018). Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm. 253-258. 10.1109/ICFHR-2018.2018.00052.

9. Learning and Outcomes

Internship in Tarah AI was a learning yet, fun experience. Being a start-up, the level of experience of its founders, expertise and professionalism is extreme and one can learn lots from them. One will be allowed to work at their own pace with no sort of pressure and this makes it stress-free. I am thankful for the experience and professional ethics I learned during this period.

10. Summary

In the internship, since I had to work with Flutter and Dart programming language, it was very similar to C language, and I was able to grasp its concepts quickly and easily due to the computer science course in my first year. Computer Vision and Machine Learning for Image Processing courses were very helpful in the application of machine learning concepts and image processing.