INTERNSHIP: PROJECT REPORT

Name Of The Student	Jaswanth Gollamudi
Project Title	Automate identification and recognition of hand written text from image.
Name of the Company	TCS iON
Name of the Industry Mentor	Anamika chatterjee
Name of the Institute	Amrita School of Engineering

Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools used
04 june 2020	23 july 2020	210	Google colab.	IAM dataset, Tensorflow2.0, keras, Google colab Gpu.

Project Synopsis:

- Recognize and identify handwritten text from an image is the motto of the project. The primary reason to choose this project is one can able to work on images with CNN's, After the enhancement in CNN's people try to solve more complex problems. There are many research done on handwritten text recognition by many companies and very good techniques are available like OCR,EAST many other API's as well. And there are many ways text recognition is used and implemented in numerous projects, A few of them are number plate detection, handwriting cheat detectors and many more two come. Researches are going on more text recognition techniques are going to be widely used wherever is necessary which helps in many ways.
- **Handwritten Text Recognition** (**HTR**) is the ability of a computer to receive and interpret intelligible handwritten input from sources and understand different styles of handwriting and learn features and find relations and map them to the appropriate character, Such techniques are widely know as OCR, Intelligent word detection.
- The abstract of the project is to create a model which help us in Automating identification and recognition of handwritten words from an image. We are going to use combination of CNN and RNN to solve this problem

Solution Approach:

There are several approaches to tackle the problem. I tried a few model but the most prioritized models are two,

The 1st approach, i tried two implement a model[inspired] pipeline which takes forms as a IAM data set and does 1]paragraph segementation 2] line segmentation and then3]handwritten recognition , for every step we use CNN's and lstms's , mxnet and many other , This complex approach is crashing on colab when I am trying to implement. I am not going into details for this approach, I am going to link the model below.

The 2nd model which I used to finish this project is a CRNN+ CTC loss function which involves taking data set, Pre-processing images, encoding labels, Mapping labels with images, Training and testing with neural net architecture, Predicting hand written recognition from images.

- I started the model with using kaggle API for collecting data instead of uploading in drive and using it. After getting the data set which contains images of handwritten words and we need (ascii) control file for the word images in order to get the labels[text for handwritten words], Then we need to do pre-processing images such that convert image to grayscale and make size of each images as (128,32) using padding and dimensions as (128,32,1) to input them in architecture and then normalize each images. For labels encode each character of a word into some numerical value by creating a function , Pad them according to RNN architecture.

- The model architecture with loss function as follows, Input the images of height 32 and width 128. we used seven convolution layers of which 6 are having kernel size (3, 3) and the last one is of size (2.2). And the number of filters is increased from 64 to 512 layer by layer. Two max-pooling layers are of size (2, 2) and two max-pooling layers of size (2, 1) are added. Then we used a lambda function to squeeze the output from the last layer and input it with LSTM layer. Then used two Bidirectional LSTM layers , This RNN layer gives the output.

- After that we need to create a custom loss function and then pass it to the model with the required parameters for CTC function.

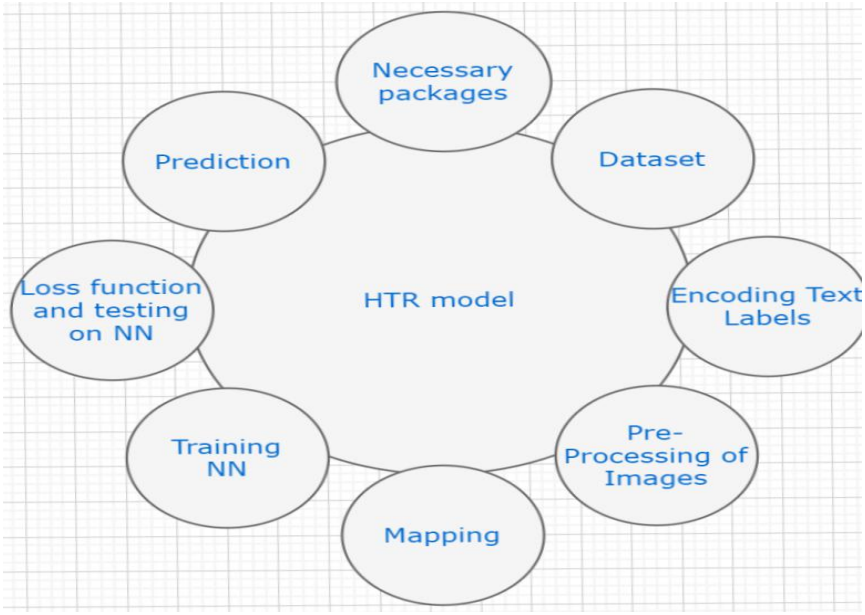
- To train the model I used Adam optimizer and save the best weights on the basis of validation loss. Now train our model on 3914 training images and 437 validation images. We can alter the training samples by changing record_count in the model.

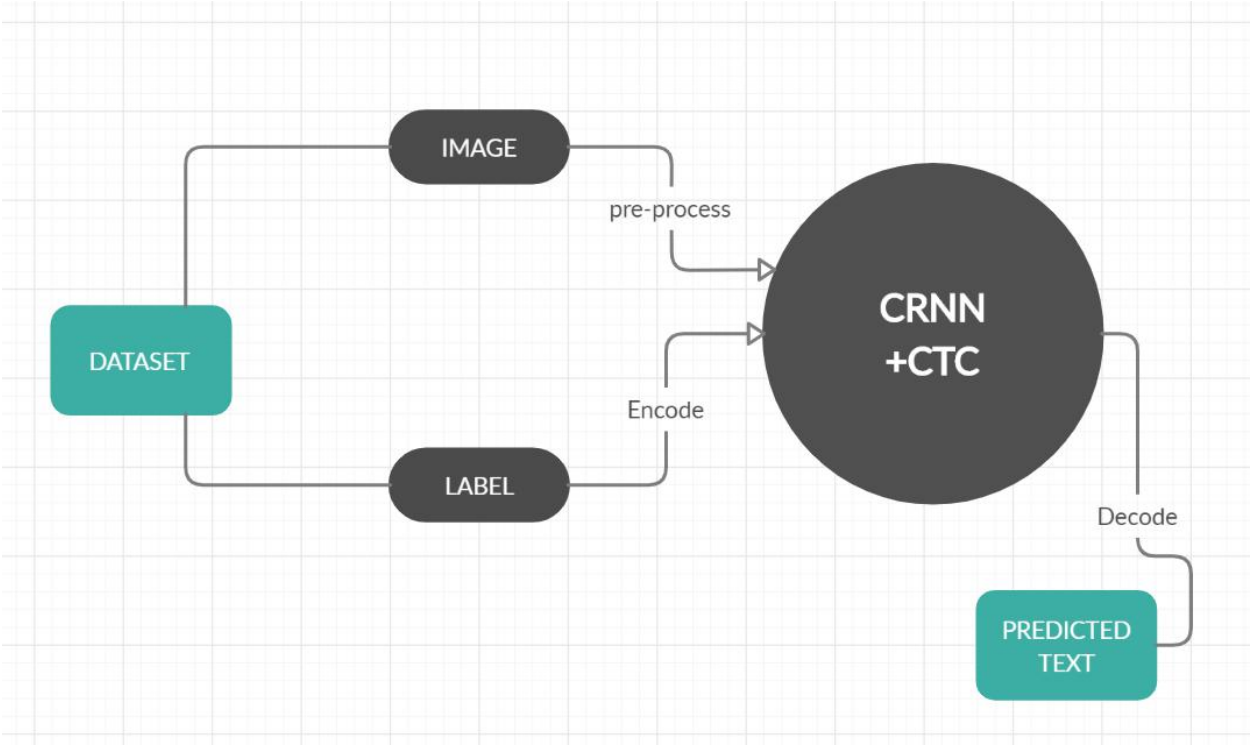
- After that testing and predicting our model I used ” act_model ” which input as test images. Here I used the CTC decoder to get the output text which follows by levenshtein method to test accuracy.

Assumptions:

1] The images are handwritten with no augmentation .
2] The word images are already bounding boxed in the data set.
3] All necessary packages with suitable versions.

Project Diagrams:





Algorithms:

The model we used is **CRNN + CTC loss function**.
To implement this model we need IAM dataset, Google Colab , Tensorflow 2.0, Keras 2.3.0, OpenCV.

The architecture of the model as follows:

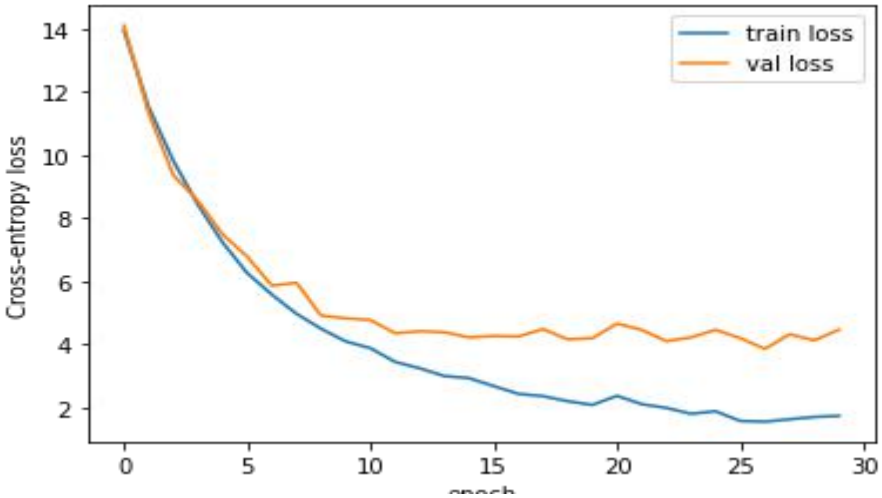
+ Code + Text

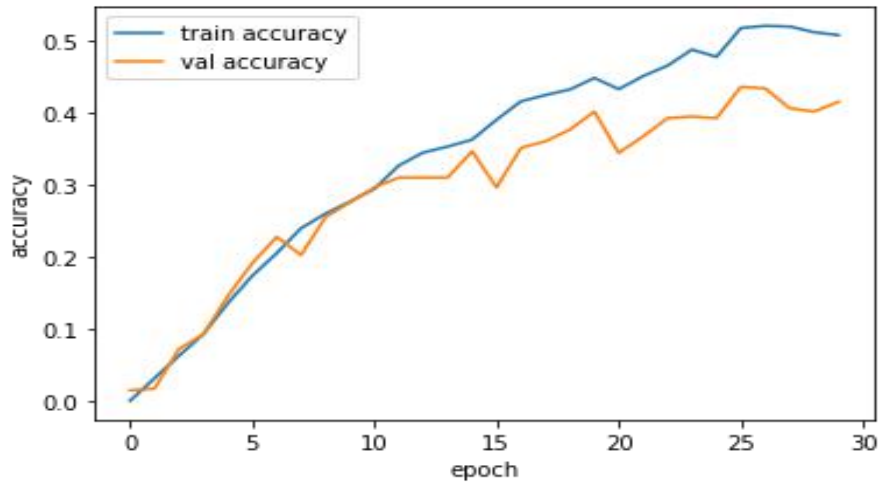
input_1 (InputLayer)	(None, 32, 128, 1)	0
conv2d_1 (Conv2D)	(None, 32, 128, 64)	640
max_pooling2d_1 (MaxPooling2	(None, 16, 64, 64)	0
conv2d_2 (Conv2D)	(None, 16, 64, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 8, 32, 128)	0
conv2d_3 (Conv2D)	(None, 8, 32, 256)	295168
conv2d_4 (Conv2D)	(None, 8, 32, 256)	590080
max_pooling2d_3 (MaxPooling2	(None, 4, 32, 256)	0
conv2d_5 (Conv2D)	(None, 4, 32, 512)	1180160
batch_normalization_1 (Batch	(None, 4, 32, 512)	2048
conv2d_6 (Conv2D)	(None, 4, 32, 512)	2359808
batch_normalization_2 (Batch	(None, 4, 32, 512)	2048
max_pooling2d_4 (MaxPooling2	(None, 2, 32, 512)	0
conv2d_7 (Conv2D)	(None, 1, 31, 512)	1049088
lambda_1 (Lambda)	(None, 31, 512)	0
bidirectional_1 (Bidirection	(None, 31, 512)	1574912
bidirectional_2 (Bidirection	(None, 31, 512)	1574912
dense_1 (Dense)	(None, 31, 79)	40527

- The model architecture with loss function as follows, Input the images of height 32 and width 128. we used seven convolution layers of which 6 are having kernel size (3, 3) and the last one is of size (2.2). And the number of filters is increased from 64 to 512 layer by layer. Two max-pooling layers are of size (2, 2) and two max-pooling layers of size (2, 1) are added. Then we used a lambda function to squeeze the output from the last layer and input it with LSTM layer. Then used two Bidirectional LSTM layers , This RNN layer gives the output. After that we need to create a custom loss function and then pass it to the model with the required parameters for CTC function.

Outcome:

The model able to predict **"**handwritten word from an image**"** with good accuracy according to number of training samples. I had taken around 3.7k images for training got an accuracy of **"**88.7**"**. Plots for loss and accuracy as follows:





Exceptions considered:

1] The image should contain a bounding box across the word.
2] There shouldn't be to noise around the handwritten text.
3] The colour backgrounds are not be multicoloured.

Enhancement Scope:

1] This model can be enhanced into predicting lines, sentences and even paragraphs.
2] Creating good Data sets which help us to predict handwritten pages or paragraphs.
3] Using great encoding and decoding techniques and good pre-processing techniques to solve complex handwritten image problem.

References:

1] https://github.com/TejasReddy9/handwriting_cnn/blob/master/neural_net_model.ipynb
2] https://github.com/awslabs/handwritten-text-recognition-for-apache-mxnet/blob/master/3_handwriting_recognition.ipynb
3] https://github.com/shaheerakr/cheating-detector
4] https://learn-neural-networks.com/handwriting-recognition-by-using-convolutional-neural-network/
5] https://www.appliedaicourse.com/course/11/Applied-Machine-learning-course
6] https://keras.io/examples/vision/captcha_ocr/

7]Blogs and self_learning links which are provided throughout the internship.

Link to Code and executable file:

1] final model:-
https://colab.research.google.com/drive/1b4zT57vhu_fmPpEo-_izTd74PRa2hlD1?usp=sharing

2] First approach:-
https://colab.research.google.com/drive/1qEKq-lCncSfHRpyp8JMCpOTOekoEpvBK?usp=sharing