Handwritten Text Recognition

A DISSERTATION

Submitted in partial fulfilment of

the requirements for the award of the degree

of

Master of Technology

in

Computer Science and Engineering

By

SHANU SHARMA

(Enrolment No. 17535025)

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE - 247667 (INDIA)

MAY 2019

CANDIDATE'S DECLARATION

I declare that the work presented in this dissertation with title "**Handwritten Text Recognition**" towards the fulfillment of the requirement for the award of the degree of Master of Technology submitted in the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, India. It is an authentic record of my own work carried out under the supervision of **Balasubramanian Raman**, Professor, Department of Computer Science and Engineering, IIT Roorkee.
The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

DATE : ..... SIGNATURE:

PLACE: Roorkee (SHANU SHARMA)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

DATE : ..... SIGNATURE:

(Balasubramanian Raman)

Professor

Dept. of CSE

IIT Roorkee

ACKNOWLEDGEMENT

On completion of my dissertation, I would like to express my deepest gratitude to my supervisor, **Balasubramanian Raman** (Professor, Department of Computer Science and Engineering) for his constant guidance, motivation and support. Throughout my thesis working period, he provided encouragement, sound advice, good teaching and lots of good ideas. He always managed to spare time for his student's research queries despite his extremely busy schedule.

I would like to take this opportunity to express my profound gratitude to my guide not only for his academic guidance but also for his interest in my project. Finally, I am very grateful to my Institution and colleagues whose constant encouragement served to renew our spirit. I wish to avail myself of this opportunity to express a sense of gratitude and love to my friends and my beloved parents for their support and strength.

Date : .....

Place : Roorkee (SHANU SHARMA)

Abstract

Offline handwritten text recognition from images is a significant is- sue for the organisations endeavoring to digitize huge volumes of handmarked scanned. Handwriting recognition is the capability of the computers to get and translate comprehensible handwritten in- put from sources for example paper reports, photos, contact screens and different gadgets into a digital format so that it can be used by computers for various purposes. In past there are various other tech- niques are used such as manual feature extraction, Hidden markov model etc. But these such techniques either requires substantially more development time or are not as much accurate. In this thesis, we created a neural network that is trained on word-pictures which is taken from the IAM dataset to translate word images into digital format.

Contents

List of Figures

Abbreviations

CNN	Convolution Neural Network
RNN	Recurrent Neural Network
HTR	Handwriting Text Recognition
OCR	Optical Character Reader
TF	Tensor Flow
LSTM	Long Short Memory
CTC	Connectionist Temporal Classification

Chapter 1

Introduction

We often come across few neglected but most wondered questions about :

- How can we recognize the given text in your samples or the dataset?
- How do we recognize the text in given lines or sentences?
- How to read the historical documents in the form of digital texts?

Another major issues which exist are those handwritten characters that are written by various author,writers and historians all across the globe that are not only nonidentical but also in varies in dif- ferent aspects such modified in varying sizes and shapes. Various varieties recorded as a hard copy styles of individual character make the recognition task troublesome; the likenesses of various character in shapes, the covers, and the interconnections of the neighboring characters further entangle the character recognition issue. The solution to all these problems are now framed under one roof named as, Handwritten text Recognition (HTR).This has been a noteworthy research issue for quite a few years and has increased late force because of the potential esteem that can be opened from removing the information put away in manually written reports and misusing it through current computer based intelligence frameworks. HTR is the Method of transcribing the handwritten text into the digital text. Transcribed content is a general term, and we needed to limit the extent of the venture by indicating the significance of manually written content for our motivations. In this task, we assumed the test of arranging the picture of any handwritten word, which can be of either recursive composition or square composition type. This task can be joined with calculations that fragment the word pictures in a given line picture, which can in turn be consolidated with algorithms that

fragment the line pictures in a given picture of an entire manually written page. HTR is partitioned into two classes:

- Offline recognition.
- Online recognition.

Handwriting recognition essentially involves optical character recog- nition. However, an entire HTR system likewise controls formatting, performs correct partition among the characters and locate the most conceivable words. In this work, we consider the offline recognition issue which is impressively all the more testing and considerably challenging as, not at all like the online mode which exploits properties like stroke data and direction in expansion to the content picture, offline mode has just the picture accessible for highlight extraction.We approach this issue with complete word pictures since CNNs will in general work better on crude info pixels rather than highlights or parts of a pic- ture [4]. Especially, we make the accompanying key commitment that:

- We present an end-to-end neural-network architecture which is composed of convolutional and recurrent networks to carry out efficient offline HTR on images of text lines.
- This project can be used with algorithms which segments the word images given in a line image, which in thus can be combined with algorithms segments the image of a line in a given image of a whole handwritten page.

Chapter 2

Review of Literature

2.1 Early Scanner

The very first driving force behind the need of Handwritten text classi

er was for the digits classification for the postal mails. Jacob Rabinows early postal readers embodied scanning equipment and hardwired logic systems to identify mano-spaced fonts[3]. Allum et. al further improved this by making a sophisticated scanner that al- lowed for far more variations in how the text was written as well as also the encoding the information data onto the barcode that was directly printed onto the letters.

2.2 The digital age scanners

The very first promising piece of OCR software was designed by Ray Kurzweil in 1974 as his software allowed for recognition for any font[5]. This software used a more advanced use of pattern matching (matrix method). Essentially, this software compares the bitmaps of the template character with the bitmaps of the readed character and would than compare them to identify which character it is very closely matched with. The main disadvantage of this software was its sensitivity to the variations in the sizing and the difference between the way of writing of each individuals. For further improvement on the templating, OCR software started using feature extraction inspite of the templating. For each and every character, software would look for the features like projection histrograms, geometric moments and zoning[6].

2.3 Machine Learning

Lecun et. al focused on using the gradient-based machine learning techniques using multi-module machine learning models, a forbearer to some of the initial end to end modern deep learning models[12]. Then there comes the major upgrade in producing high OCR accuracy was the use of Hidden Markov Model for the OCR. This approach uses letter as different states,which at that point takes into account the setting of the character to be represented while deciding the following hidden variable. [8] The most considered principle dis- advantage of this was still the manual extraction highlights, which requires earlier information of the language and was not especially vigorous to the assorted variety and intricacy of handwriting.

The approach of end-to-end Handwritten Paragraph Recognition with MDLSTM Attention [16] was to take a LSTM layer for each checking heading and encode the crude picture information to a feature map. The model would then utilize thoughtfulness regarding accentuate certain feature maps over others. After the feature map was developed, it would be nourished into the decoder which would foresee the character given the current picture synopsis and state.The key feature highlights of this methodology was very novel since it did not decouple the division and characterization forms as it did both inside a similar model [16]. But the drawback of this model is that it doesn't fuse a language model to produce the arrangement of characters and words. It is totally reliant on the visual order of each character without considering the setting of the developed word. Earlier Researchers found CS 231N task to be useful in managing with the task or errand .They utilized the Quicker RCNN model [10] to recognize the given singular characters inside a word and for order i.e whether an object exists within those limiting boundaries or not. This RCNN model uses a sliding window over the picture to initially decide if an item exists inside the limits. That limited picture is then ordered to its relating character. They likewise executes alter separate which takes into consideration making adjustments to the characterized word to decide whether another arranged word is more likely to be correct.

Chapter 3

Proposed Model

In our proposed approach we uses Neural Network (NN), consisting of Convolutional NN (CNN) layers, Recurrent NN (RNN) layers and a final Connectionist Temporal Classification (CTC) layer for the objective.

3.1 Convolution Neural Network

CNN or ConvNet abbreviated for The Convolutional Neural Network is a Deep Learning algorithm that takes in an input image, projects various importance aspects or the objects in that image that would differentiate one image from the another one. Various important aspects includes learnable weights and biases. The pre-processing required in Convolutional Neural Network is quite less compared to those in other classification algorithms. While in crude strategies filters are hand-designed, with enough preparing, ConvNets can get familiar with these filters/attributes. A ConvNet can successfully get the Spatial and eeting conditions in a picture using relevant channels. The engineering plays out a better fitting than the picture dataset due to the decline in the amount of parameters included and reusability of loads. By the day's end, the framework can be set up to appreciate the progression of the picture much better. Convnets primarily consists of three different types of layers:

- Convolution Layers
- Pooling layers
- Fully connected layer.

3.1.1 Convolution Layer

In Convolutional Layers, a matrix which is known as Kernel matrix is passed over the input matrix to create a feature map for the following layer. The components of the kernel can likewise be changed in accordance with produce an alternate feature map, or to extend the information along one dimension while diminishing its size along different axes. Once in a while, values on the feature map are computed by taking the sum of the result of an element-wise multiplication of the kernel and an appropriately sized section of the input matrix. Often, a dot product is used in place of the element-wise multiplication, but this modification can be done for better (or worse) results. We can make use of many kernels in a given convolutional layer and concatenate the given results for creating the feature map. Since, one kernel can be used for the entire image, hence it makes the convolutional neural networks location-invariant and preventing the same from overfitting.

3.1.2 Pooling Layer

Next, a pooling layer is applied on to the feature map which is produced by the convolution layer. Its function is to simultaneously reduce the spatial size of the matrix representation to decrease the number of parameters and the amount of computation applied on the network. Pooling layer independently operates on each feature map. Max pooling, is the most frequent type of pooling used in the Convolutional Neural Network (CNN), which simply means picking up the maximum value from a given list of numbers. In this case, We splits up the feature map into a couple of boxes and choose only the maximum value from each and every box. Here is what it actually looks like:

3.1.3 Fully Connected Layer

The final layer of a convolutional neural network (CNN) is called the fully connected layer. This layer is a standard neural network layer in which some non linearity function such as ReLu, tanh, sig- moid, etc. is applied to the dot product of an input and a matrix of weights. Then a softmax function can convert the output into an array of probabilities for the purpose of classification. Convolutional neural networks (CNN's) usually have far more than just three layers. Convolutions and max-pooling layers can be made to be stacked on top of each other indefinitely for better and better results depending on certain thresholds. Here is a picture of a very deep convolutional neural network consisting of

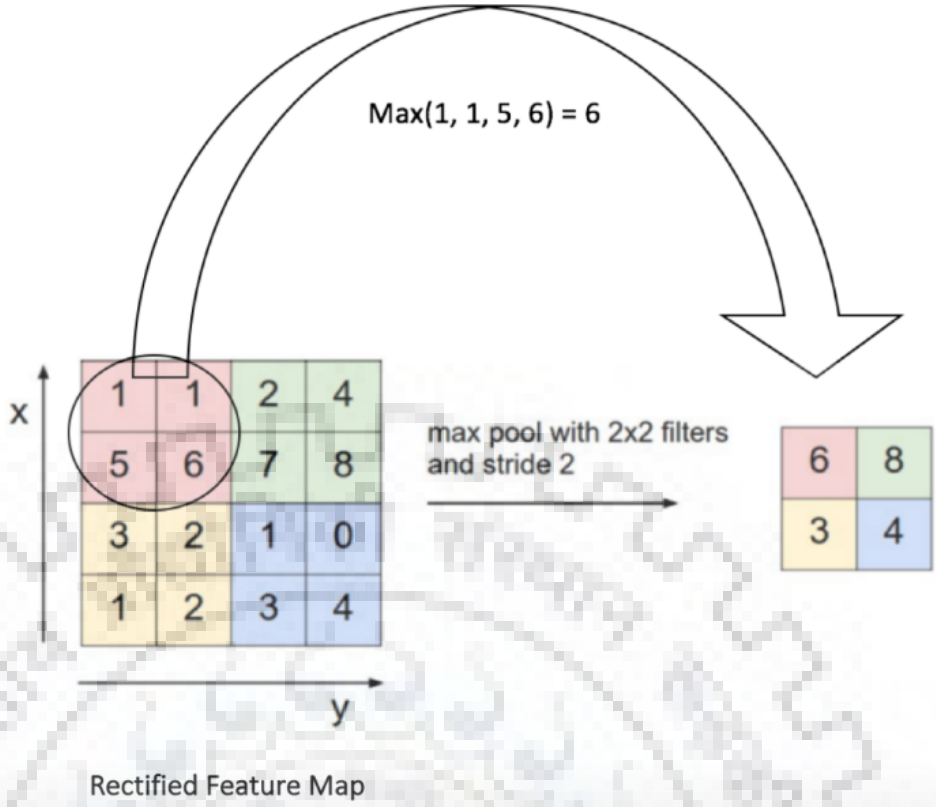
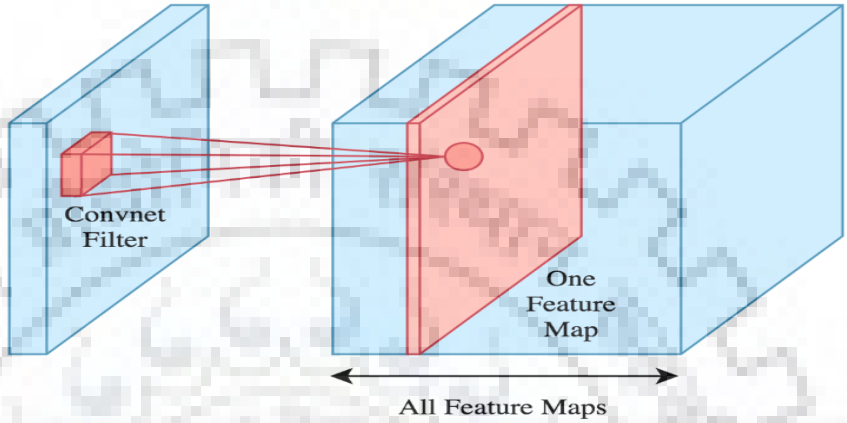


Fig. 3.1. Convolutional Layer [5]

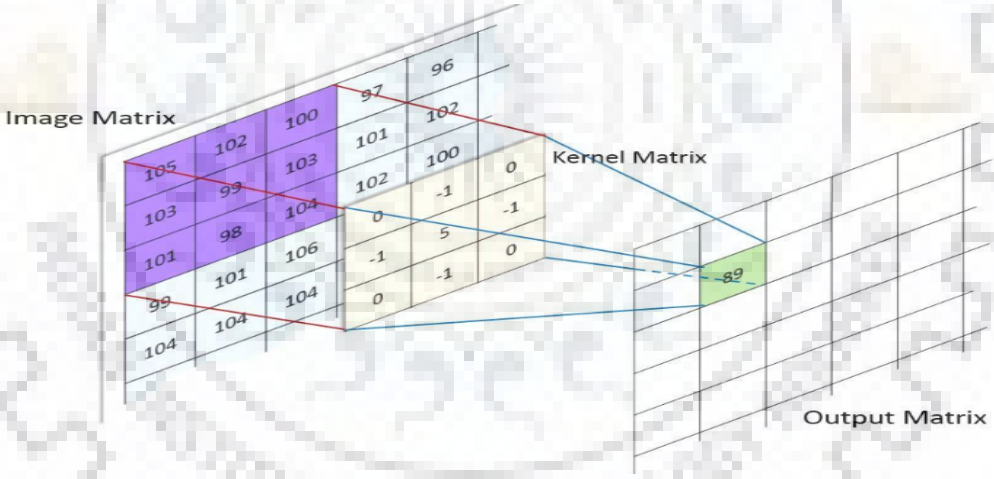
many layers: Our Convolutional Neural Network (CNN) uses total 10 layers .The input image is fed into the CNN layers.These layers are then fur- ther trained to extract relevant features from the image. First, the convolution operation, which applies a filter kernel of size 6*6 in the first four layers and 4*4 in the last six layers to the input. Then, the non-linear RELU function is applied. Finally, a pooling layer sums up the image regions and outputs a reduced size version of the input.

3.2 Recurrent Neural Network

Recurrent Neural Networks (RNN) are very powerful and robust type of neural networks and belongs to one of the most promising algorithms out there at the moment due to fact that they are the only ones having an internal memory. Because of its internal memory, Recurrent Neural Networks are able to keep track of important things pertaining to the input that they have received, which makes them to be very precise and accurate in predicting what's about to come next. This is the main reason why they are the preferred al- gos for sequential data such as time series data, speech data, text data, financial data, audio data, video data, weather data and much more because of the reason that they can



(a) Convolutional Layer



(b) Low Level Diagram of A List Of Points That Filter Maps From An Input Matrix To A Single Node In The Following Layer

Fig. 3.2. Overview Of Convolutional Layer [7]

construct a much deeper knowledge of a sequence and its context, as in comparision to other related algorithms. In a Recurrent Neural Networks, the data and the information re- garding the cycles passing through the loop. During the processing of the results, it takes into account, the recent input and also what it has learned from the previously received inputs. Recurrent Neural Network has a short duration memory. Combined with a Long Short Term Memory model they have a long duration memory. This is the main reason we use Long Short Term Memory (LSTM) model in our proposed solution.

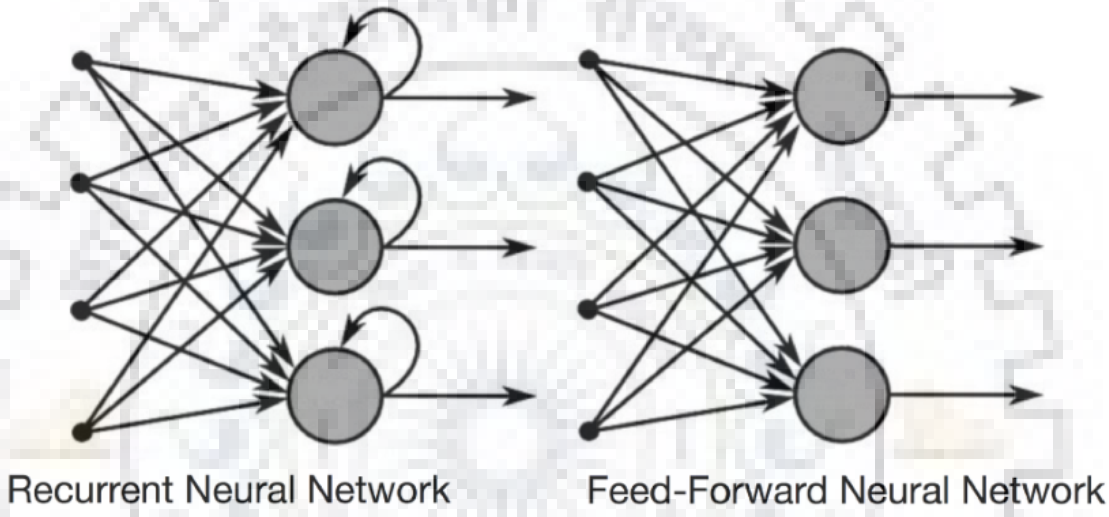


Fig. 3.3. Recurrent Neural Network and Feed Forward Network [8]

3.3 Long-Short Term Memory

Long Short-Term Memory (LSTM) networks systems are an expan- sion for recurrent neural networks (RNN's), which basically extends their memory. which basically extends their memory. Therefore it is well suited to learn from important experiences that have very long time lags in between. LSTM's enable RNN's to remember their inputs over a long period of time. This is because LSTM's contain their information in a memory, that is much like the memory of a computer because the LSTM can read, write and delete information from its memory. This memory can be viewed as a gated cell, where gated implies that the cell chooses whether or not to store or erase data (e.g on the off chance that it opens the doors or not), founded on the sig- nificance it doles out to the data. The doling out of significance occurs through loads, which are additionally learned by the

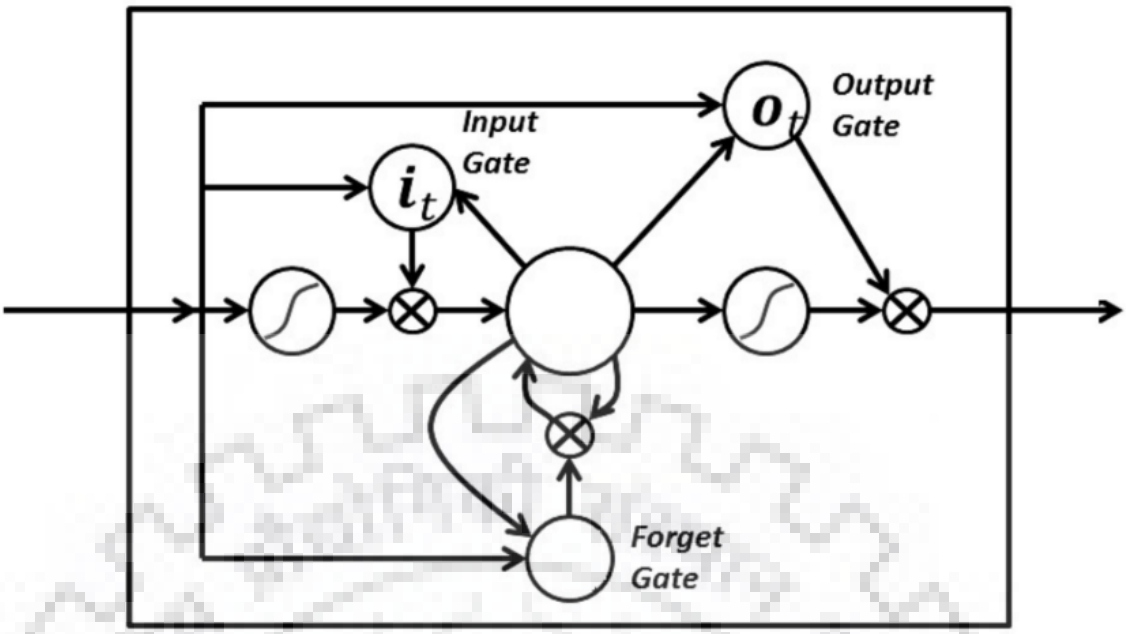


Fig. 3.4. Long Short Term Memory [10]

calcula- tion. This essentially implies it learns after some time which data is significant and which not. In a LSTM you have three gates:

- Input gate,
- Forget gate
- Output gate.

These gates decide if to give new info access (input gate), erase the data since it isn't significant (forget gate) or to give it a chance to affect the yield at the present time step (output gate). You can see a representation of a RNN with its three gates beneath: The gates in a LSTM are simple, as sigmoids, implying that they extend from 0 to 1. The way that they are analog, empowers them to do backpropagation with it. The tricky issues of vanishing gradient is understood through LSTM on the grounds that it keeps the gradients soak enough and thus the preparation moderately short and the precision high. The feature sequence contains 256 features for each time-step, the RNN engenders pertinent data through this grouping. The well known Long Short-Term Memory (LSTM) execution of RNNs is utilized, as it can spread data through longer separations and gives more strong preparing attributes than vanilla RNN. The RNN yield succession is mapped to a lattice of size 3280. The IAM dataset com- prises of 79 unique characters, further

one extra character is required for the CTC activity (CTC clear mark), along these lines there are 80 passages for every one of the 32 time-steps.

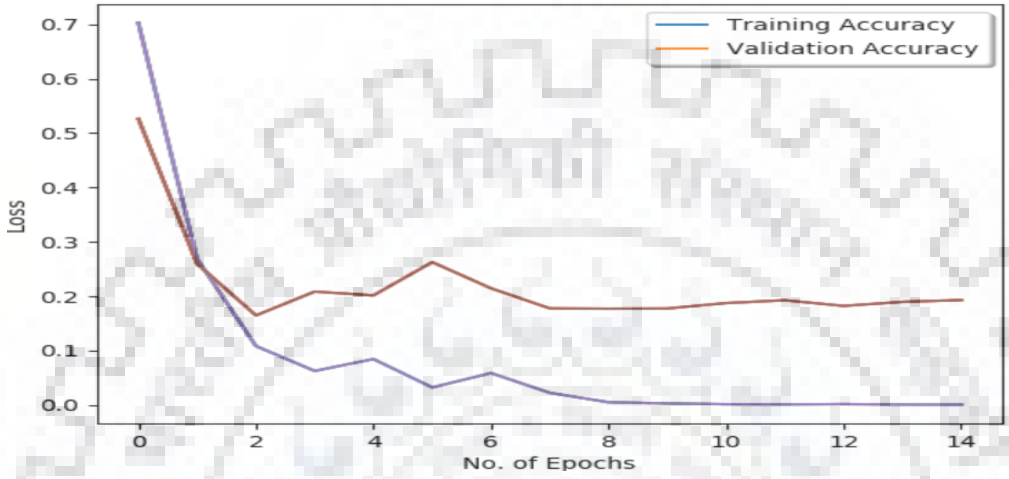
3.4 CTC

CTC scores would then be able to be utilized with the backengendering calculation to refresh the neural system loads. while setting up the NN, the CTC is given the RNN matrix and the ground truth text data and it ascertains the loss value. While translating, the CTC is simply given the matrix and it deciphers it into the final content. Both the ground truth information and the perceived content information can be of most extreme 32 characters long. For misfortune computation, we feed both the ground truth con- tent and the network to the activity. The ground truth content is encoded as a scanty tensor. The length of the information arrange- ments must be passed to both CTC activities. The mean of the misfortune estimations of the group components is utilized to prepare the NN: it is bolstered into an analyzer, for example, RMSProp.

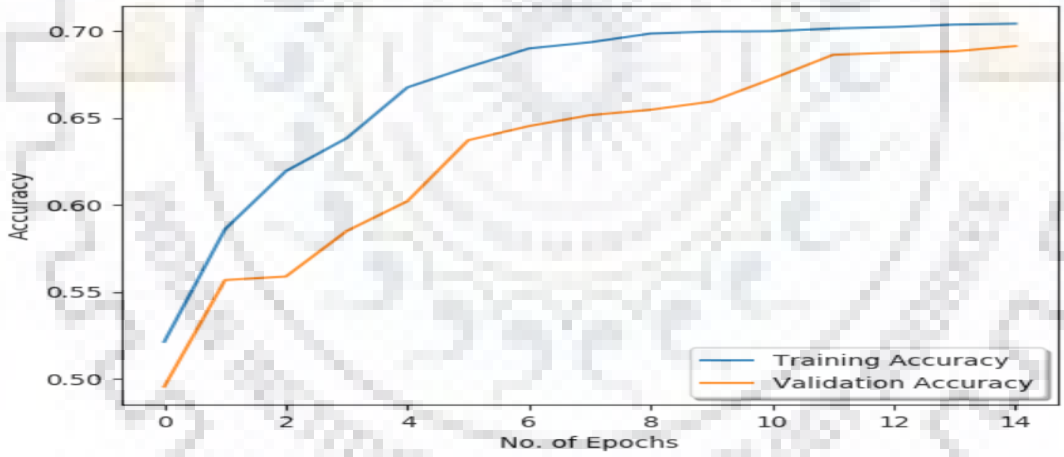
3.5 Result

Initially, when we start training our model i.e, from the very scratch our model has high losses as shown in the simulation results. The loss is defined as the square of the difference between the actual value and the predicted value. This is because our Neural Network does not know anything before. This value is further used in back propagation process to adjust the weights and biases. So, as we gradually train our model, the NN model will adjust its weights and biases according to the pattern it adapted from previous training data. The predicted value now gets closer and closer to the actual value, thus decreasing the loss value. The simulated result shown in fig. 3.5(a) shows the trend of Loss values with the number of epochs as the number of epochs increases.

Since over the course of training Loss value decreases which means prdicted value gets closer to the actual value. As a result, Accuracy gets better and better since accuracy shows how close our predicted value is as compared to the actual value. Simulated result shown in fig. 3.5(b) shows how the Accuracy is improving as we train our model over the course of time.



(a) Handwritten Text Recognition Loss



(b) Handwritten Text Recognition Accuracy

Fig. 3.5. Simulated Results

Chapter 4

Conclusion

Our used model which consists of CNN along with LSTM seems to be a promising approach. Since, using CNN the User don't have to go through a rigorous method of sophisticated feature extraction and classification of appropriate symbol used in the given text. Our neural network model will automatically train on the given the training set and will predict the output appropriately.

With the improved number of CNN layers and our LSTM model We certainly improved the accuracy of predicting the word to 70.341 percent, which is quite decent as compared to previous implementation. Also, since our dataset contains cursive words, Our model learned the cursive writing pattern. So, given any word which is clearly written our model can predict it quite accurately whether it is a cursive or simple handwritting.

Hence, in our proposed solution, implementation using TF (Tensor- flow) is provided. Finally, a way to improve the accuracy of Hand- written Text Recognition is given.

Chapter 5

Future Scope

- Increase the dataset-size by applying further random transfor- mations to the input images dataset.
- Increase input size.
- Use token-passing to restrict the output to dictionary words,in the event that the perceived word isn't contained in a lexicon, scan for the most comparative one.
- Use of either 2D LSTM or GRU in place of LSTM and check the accuracy.