VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

Internship Report on

"AUTOMATE IDENTIFICATION AND RECOGNITION OF HANDWRITTEN TEXT FROM AN IMAGE"

Submitted in partial fulfillment for the award of Degree of,

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

By

B A SOHANKUMAR 4AL18CS013

Under the Supervision of

MRS.REENA LOBO

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ALVA'S INSTITUTE OF ENGINEERING AND TECHNOLOGY

MOODBIDRI-574225, KARNATAKA

2021 – 2022

ALVA'S INSTITUTE OF ENGINEERING AND TECHNOLOGY

MIJAR, MOODBIDRI D.K. -574225

KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Internship report on "Automate identification and recognition of handwritten text from an image" submitted by B A SOHANKUMAR 4AL18CS013 is work done by him/her and is submitted during the academic year 2021 – 2022, in partial fulfilment of the requirements for the award of the degree of BACHELOR OF ENGINEERING in DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING of VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Internship report has been approved as it satisfies the academic requirements in respect of Internship work prescribed for the Bachelor of Engineering Degree.

Internship Guide
Department of CS&E

Internship Coordinator
Department of CS&E

Head of the Department
Department of CS&E

Examiners

Name of the Examiner **Signature with Date**

1.

2.

ACKNOWLEDGEMENT

First I would like to thank **Mr. Debashis Roy*, TCS iON*,** for giving me the opportunity to do an internship within the organization.

I would also like to thank all the people who worked along with me in **TCS Ion** with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Managing Trustee **Mr. Vivek Alva** and Principal **Dr.Peter Fernandes, Alva's *Institute of Engineering and Technology, Mijar*** for the facilities provided to accomplish this internship.

I thank my Head of the Department **Dr. Manjunath Kotari, *Professor, Department of Computer Science and Engineering*** for his constructive criticism throughout my internship.

I am thankful to Internship coordinator, **Mrs. Merlyn Melita Mathias**, ***Assistant Professor, Department of Computer Science and Engineering*** for her valuable guidance, support and advices to get and complete internship in above said organization.

I would like to thank mu internship guide **Mrs. Reena Lobo, Assistant Professor, *Department of Computer Science and Engineering*** for **his** constant help and support throughout the internship.

I am extremely grateful to my department staff members and friends who helped me in successful completion of this internship.

B A SOHANKUMAR

4AL18CS013

TCS iON

TATA

CERTIFICATE of INTERNSHIP

This is to certify that

B A SOHANKUMAR

has successfully completed Remote Internship

for ***210 hours*** *in project titled*

Automate Identification and Recognition of Handwritten Text from an Image

by TCS iON from 17 Nov 2020 *to* 12 Feb 2021 .

TCS iON REMOTE INTERNSHIPS
Academic Credits with Industry Mentors
Cert. ID.: 358-11573908-1016
Dated: 03 Jan 2021

CERTIFIED

Mehul Mehta

Mehul Mehta
Global Delivery Head - TCS iON,
Tata Consultancy Services

COMPANY PROFILE

TCS iON is a strategic unit of Tata Consultancy Services focused on Manufacturing Industries (SMB), Educational Institutions and Examination Boards. TCS iON provides technology by means of a unique IT-as-a-Service model, offering end-to-end business solutions. It caters to the needs of multiple industry segments, through innovative, easy-to-use, secured, integrated, hosted solutions in a build-as-you-grow, pay-as-you-use business model. TCS iON serves its clients with the help of best practices gained through TCS' global experience, domestic market reach, skills, and delivery capabilities. TCS iON's Cloud Based Solution is highly modular, scalable and configurable giving businesses and educational institutions the benefits of increased efficiencies, faster go to market, predictability of technology as well as spend and better business results.

ABSTRACT

Handwritten text acknowledgment is yet an open examination issue in the area of Optical Character Recognition (OCR). This paper proposes a productive methodology towards the advancement of handwritten text acknowledgment frameworks. The primary goal of this task is to create AI calculation to empower element and information extraction from records with manually written explanations, with an, expect to distinguish transcribed words on a picture.

The main aim of this project is to extract text, this text can be handwritten text or it can machine printed text and convert it into computer understandable or we can say computer editable format. To implement this project we have used PyTesseract which is an open-source OCR engine used to recognize handwritten text and OpenCV a library in python used to solve computer vision problems. So the input image is executed in various steps, first there is pre-processing of an image then there is text localization after that there is character segmentation and character recognition and finally we have post-processing of image. Further image processing algorithms can also be used to deal with the multiple characters input in a single image, tilt image, or rotated image. The prepared framework gives a normal precision of more than 95% with the concealed test picture.

TABLE OF CONTENTS

LIST OF FIGURES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

1st WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Tuesday	17/11/2020	Cleared the RIO-pre assessment test	Yes
Thursday	19/11/2020	Basics of Artificial intelligence and objects of AI	Yes
Monday	21/11/2020	History of AI And Modern of AI	Yes
Tuesday	24/11/2020	Learned about Health care application and genomic of AI	Yes

2nd WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	27/11/2020	Basics of Applications of AI	Yes
Tuesday	28/11/2020	How to work with different AI Application and modules.	Yes
Wednesday	30/11/2020	Learned about staffing structure.	Yes
Thursday	01/12/2020	Concepts of Data science organizational structure and centralized and distributed teams	Yes
Friday	03/12/2020	How to work with different AI to optimize operations.	Yes
Saturday	04/12/2020	Concepts of how the AI works in Enterprise.	Yes

3rd WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	09/12/2020	Decision Making a Deployment of problem statement.	Yes
Tuesday	10/12/2020	Doing Research work on the problems statement.	Yes
Wednesday	11/12/2020	Working with different algorithm for solution.	Yes
Thursday	12/12/2020	Training Data using dataset to train model.	Yes
Friday	13/12/2020	Working on over fitting, training, and testing data modules.	Yes
Saturday	14/12/2020	Basics of Python.	Yes

4th WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	16/12/2020	Concept of K Nearest Neighbors (KNN) Algorithm.	Yes
Tuesday	17/12/2020	Concept of Exabyte's Per Day.	Yes
Wednesday	18/12/2020	How to Access the Data Web scraping.	Yes
Thursday	19/12/2020	Concepts of Plot to analyze the complexity of model.	Yes
Friday	20/12/2020	About Datacenters and Gateways.	Yes
Saturday	21/12/2020	Concept of End-to-End computing for AI.	Yes

5th WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	23/12/2020	Working on existing code optimization.	Yes
Tuesday	24/12/2020	Concepts of NumPy.	Yes
Wednesday	25/12/2020	Working on NumPy libraries.	Yes
Thursday	26/12/2020	Concept of NumPy Arrays.	Yes
Friday	27/12/2020	Concept of binary Formats in python.	Yes
Saturday	28/12/2020	SciPy builds on NumPy.	Yes

6th WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	30/12/2020	Basics About of SciPy.	Yes
Tuesday	01/01/2021	Concept of Type handling and other useful functions.	Yes
Wednesday	02/01/2021	Installation of ipython its interactive shell.	Yes
Thursday	03/01/2021	Learning and predicting of model persistence.	Yes
Friday	04/01/2021	Working with different exercise.	Yes
Saturday	05/01/2021	Concepts of pandas cookbook.	Yes

7th WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	07/01/2021	Importing libraries on pandas cookbook.	Yes
Tuesday	08/01/2021	Concepts related to Keras API.	Yes
Wednesday	09/01/2021	Basics of Keras functions.	Yes
Thursday	10/01/2021	Import the Fashion MNIST dataset.	Yes
Friday	11/01/2021	Building the neural network using MNIST dataset.	Yes
Saturday	12/01/2021	Downloading CIFAR 10 dataset.	Yes

8th WEEK

DAY	DATE	NAME OF THE TOPIC/MODULE	COMPLETED
Monday	15/01/2021	Developing different layers in the model.	Yes
Tuesday	19/01/2021	Working on the Text classification with an RNN.	Yes
Wednesday	21/01/2021	Created the text encoder in the model.	Yes
Thursday	24/01/2021	Work on the image classification using keras to solve the problem	Yes
Friday	25/01/2021	About Image classification Configure the dataset for performance	Yes
Saturday	12/02/2021	Preparation of Viva report is done with academic mentor.	Yes

CHAPTER 1

INTRODUCTION

TCS iON is a strategic unit of Tata Consultancy Services focused on Manufacturing Industries (SMB), Educational Institutions and Examination Boards. TCS iON provides technology by means of a unique IT-as-a-Service model, offering end-to- end business solutions. It caters to the needs of multiple industry segments, through innovative, easy-to-use, secured, integrated, hosted solutions in a build-as-you-grow, pay- as-you-use business model. TCS iON serves its clients with the help of best practices gained through TCS' global experience, domestic market reach, skills, and delivery capabilities. TCS iON's Cloud Based Solution is highly modular, scalable and configurable giving businesses and educational institutions the benefits of increased efficiencies, faster go to market, predictability of technology as well as spend and better business results.

As we know now a days there are different websites that are developed by the companies for various purposes, which require testing before deploying them to the market. This project is basically focused on identification and recognition of handwritten text from an image. It is based on enhancement of optical character recognition system. Optical character recognition or optical character reader is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image. An optical character recognition problem is basically a type of image-based sequence recognition problem. And for sequence recognition problem, most suited neural networks are recurrent neural networks (RNN) while for an image-based problem most suited are convolution neural networks (CNN). To cop up with the OCR problems we need to combine both of these CNN and RNN. The purpose of OCR in text identification and recognition from an image is to extract handwritten data to digital data. So, one can easily handle this digital data by editing, adding new information in that text. OCR is developing field of Computer Vision, Pattern Recognition and Artificial Intelligence.

CHAPTER 2

SYSTEM ANALYSIS

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

2.1 Requirement Specification

To be used efficiently, all computer software needs certain hardware components orthe other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

2.1.1 Hardware Requirements

Hardware components deal with the basic hardware requirement to develop and to run any system. It includes processor, memory etc.

Hardware Constraints:

PROCESSOR: 1.4GHz Intel core i5

RAM : 4GB

2.1.2 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software Constraints:

OPERATING SYSTEM : Linux or Windows

LANGUAGE : python

2.2 Language Used for Implementation

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system.

2.3 Platform Used for Implementation

- Google colab.

CHAPTER 3

OBJECTIVE ACHIEVEMENT

To develop machine learning algorithms in order to enable entity and knowledge extraction from documents with handwritten annotations, with an aim to first identify handwritten words on an image and then recognize the characters to transcribe the text.

An optical character recognition problem is basically a type of image-based sequence recognition problem. And for sequence recognition problem, most suited neural networks are recurrent neural networks (RNN) while for an image based problem most suited are convolution neural networks (CNN). To cop up with the OCR problems we need to combine both of these CNN and RNN. So, I used Convolutional Recurrent Neural Network (CRNN) to tackle the both the problems.

3.1 Skills Learnt

- **Python**

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.

- **Open CV**

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

- **Tensorflow**

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019. TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as JavaScript, C++, and Java.

- **Google colab**

Google is quite aggressive in AI research. Over many years, Google developed AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply Colab. Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold per-use basis.

3.2 Step-by-Step guide to creating Google-Colab and Deploying TensorFlow

3.2.1 Creating a document in a google colab

- **Google colab Chrome Extension**

If you want to create a machine learning model but say you don't have a computer that can take the workload, Google Colab is the platform for you. Even if you have a GPU or a good computer creating a local environment with anaconda and installing packages and resolving installation issues are a hassle. Collaborator is a free colab notebook environment provided by Google where you can use free GPUs and TPUs which can solve all these issues.

Opening Google Colab Notebook

On opening the website you will see a pop-up containing following tabs

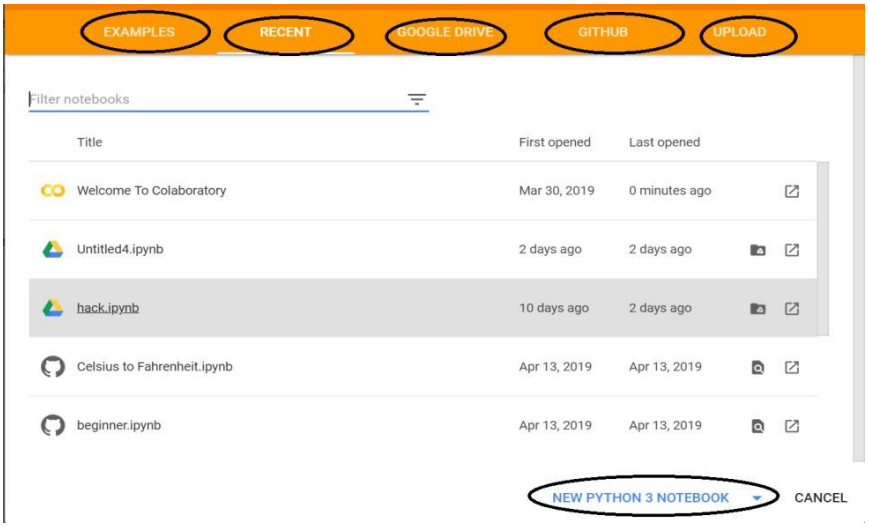

Fig 3.1: Google Colab Notebook

- **EXAMPLES:** Contain a number of Colab notebooks of various examples.
- **RECENT:** Colab notebook you have recently worked with.
- **GOOGLE DRIVE:** Colab notebook in your google drive.
- **GITHUB:** You can add Colab notebook from your GitHub but you first need to connect Colab with GitHub.
- **UPLOAD:** Upload from your local directory.

Else you can create a new Colab notebook by clicking New Python3 Notebook or New Python2 Notebook at the bottom right corner.

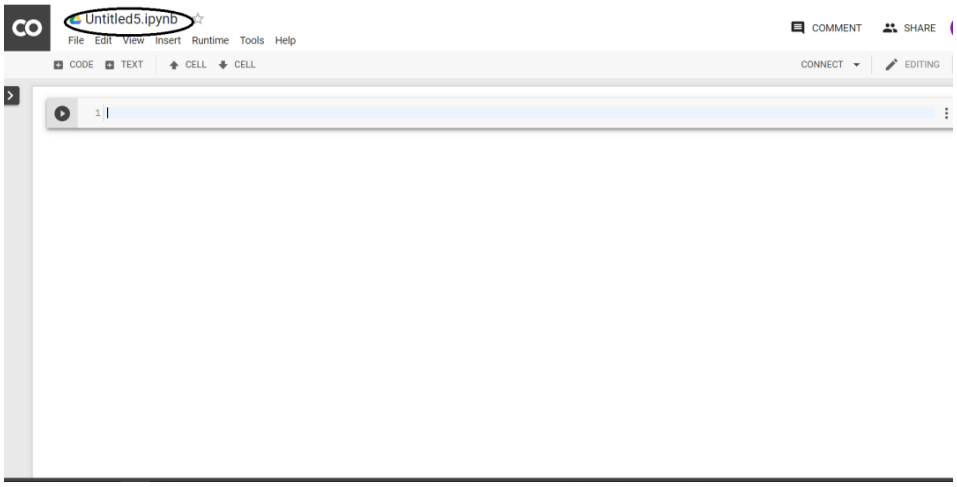

Fig 3.2 Colab Notebooks

On creating a new notebook, it will create a Colab notebook with Untitled0.ipynb and save it to your google drive in a folder named Colab Notebooks. Now as it is essentially a Colab notebook, all commands of Colab notebooks will work here. Though, you can refer the details in getting started with Colab Notebook.

CHAPTER 4

IMPLEMENTION

4.1 Tools and Technologies

- **TensorFlow**

TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019. TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as Javascript, C++, and Java.

- **Python**

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

- **Jupiter Notebook**

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from iPython in 2014 by Fernando Pérez and Brian Granger.

- **Google colab**

Google is quite aggressive in AI research. Over many years, Google developed AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply Colab.

- **IAM Handwriting Database**

The IAM Handwriting Database contains forms of handwritten English text which can be used to train and test handwritten text recognizers and to perform writer identification and verification experiments.

4.2 Instructions

1. Directory Structure :
   - Create directory called TCS Project in Google drive
   - Upload the IAM Dataset to the drive
2. Collecting the Dataset
   - We used IAM Handwritten dataset this is good dataset with several preprocessing already done.

```
[ ]  1 import numpy as np
     2 import cv2
     3 import os
     4 import pandas as pd
     5 import string
     6 import matplotlib.pyplot as plt
     7
     8 import os
     9 from google.colab import drive #To use googel drive to get files.
    10
    11 from keras.preprocessing.sequence import pad_sequences
    12
    13 from keras.layers import Dense, LSTM, Reshape, BatchNormalization, Input, Conv2D, MaxPool2D, Lambda, Bidirectional
    14 from keras.models import Model
    15 from keras.activations import relu, sigmoid, softmax
    16 import keras.backend as K
    17 from tensorflow.keras.utils import to_categorical
    18 from keras.callbacks import ModelCheckpoint
    19 from keras_tqdm import TQDMNotebookCallback
    20
    21 from sklearn.model_selection import train_test_split
    22 from sklearn.preprocessing import MinMaxScaler
```

Fig 4.2.1: Collecting the Dataset

3. Uploading the Dataset on GDrive and accessing it.
   - After uploading the dataset accessing the content stored in GDrive.

```
+ Code  + Text                                                                                    Connect  Editing

 1 from tensorflow.python.client import device_lib
 2
 3 # Check all available devices if GPU is available
 4 print(device_lib.list_local_devices())
 5 sess = tf.compat.v1.Session(config=tf.compat.v1.ConfigProto(log_device_placement=True))

[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 7568759996650258084
xla_global_id: -1
]
Device mapping: no known devices.

[ ]  1 tf.config.experimental.list_physical_devices('GPU')

[]

[ ]  1 tf.test.gpu_device_name()

''

[ ]  1 drive.mount("./gdrive", force_remount=True)
     2 with open('/content/gdrive/MyDrive/Colab Notebooks/TcsInternship/HTR_Using_CRNN/Data/words.txt') as f:
     3     contents = f.readlines()
     4
     5 lines = [line.strip() for line in contents]
     6 lines[0]

Mounted at ./gdrive
'a01-000u-00-00 ok 154 408 768 27 51 AT A'
```

Fig 4.2.2: Uploading the Dataset

4. Preprocessing the Data.
    - After fetching the dataset we will preprocess the data.
5. Dividing the dataset into train, test and validation sets.
    - After splitting we have 3240 train images, 835 test and validation images each.

Fig 4.2.3: Testing Dataset sets.

Fig 4.2.4: Validation of dataset.

6. Creating the defining the model/network architecture.
    - Adding the Pooling Layers, CNN Layers, Activation Functions, etc.

7. Training the model.
    - Using 4 epochs of 3000 train samples and 280 val samples.

Fig 4.2.5: Training the Model.

8. Saving the model.
    - Saving the model using HDF5 file extension.

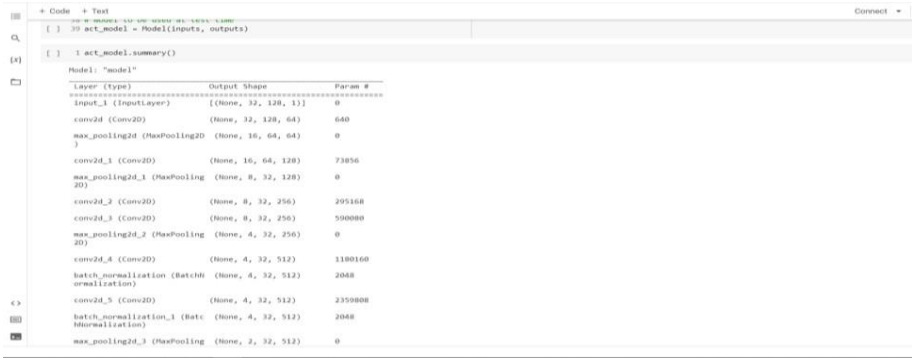

Fig 4.2.6: Saving model

9. Testing the model.

Fig 4.2.7: Testing model

10. Prediction.

    - Prediction being done on Test Files.

11. Plotting the loss and accuracy plots.

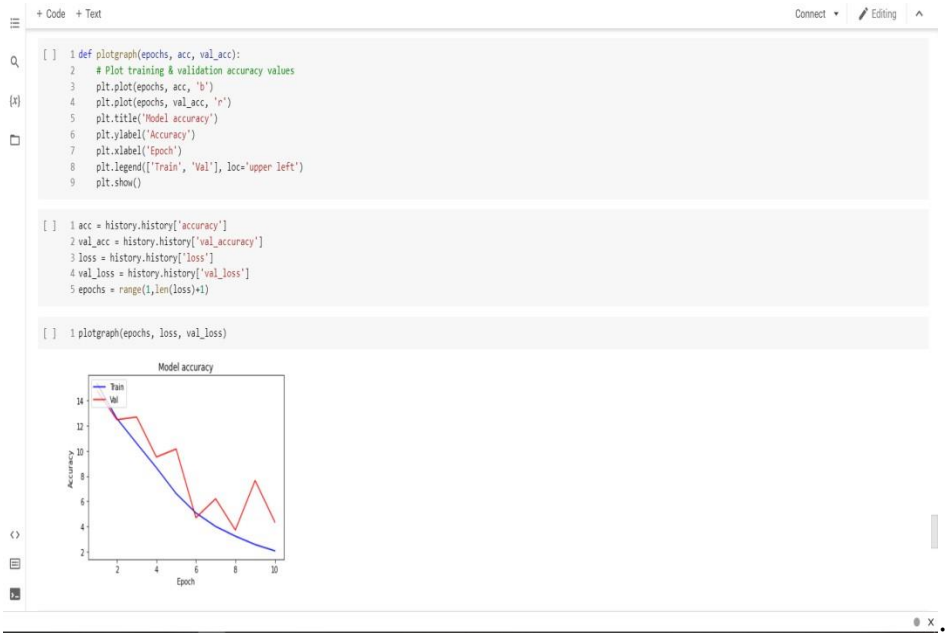

Fig 4.2.8: Plotting of loss plot.

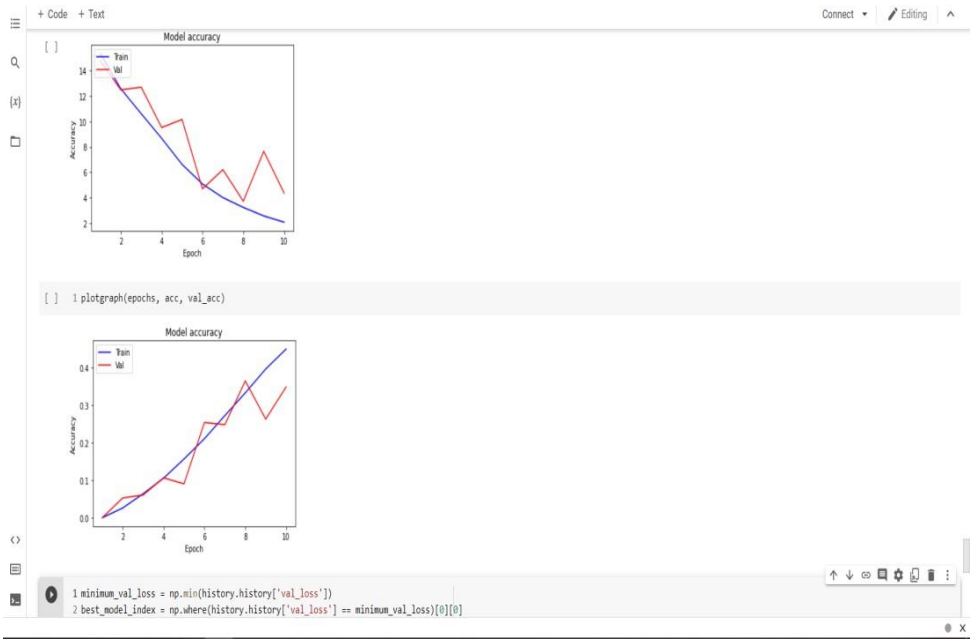

Fig 4.2.9: Plotting of accuracy plot

CHAPTER 5

CONCLUSION

In this project classification of characters takes place. The project is achieved through the conventional neural network. The accuracy we obtained in this is above 78.3%. This algorithm will provide both the efficiency and effective result for the recognition. The project gives best accuracy for the text which has less noise. The accuracy completely depending on the dataset if we increase the data, we can get more accuracy. If we try to avoid cursive writing then also it gives best results. Future Work: In future we are planning to extend this study to a larger extent where different embedding models can be considered on large variety of the datasets. The future is completely based on technology no one will use the paper and pen for writing. In that scenario they used write on touch pads so the inbuilt software which can automatically detects text which they writing and convert into digital text so that the searching and understanding very much simplified.

REFERENCES

1. https://github.com/praneetahandral/TCS_ION
2. https://www.leewayhertz.com/ethereum-smart-contract-tutorial/
3. https://www.intel.com/content/www/us/en/developer/learn/course-artificialintelligence.html
4. https://numpy.org/doc/
5. https://www.tensorflow.org/tutorials/images/classification