VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI - 590018

Report
on

CHARACTER RECOGNITION USING NEURAL NETWORKS

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

1BG18CS008 ANAGHA R
1BG18CS023 CHANDAN KUMAR G

Vidyayâmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU,
Approved by AICTE, Accredited as Grade A Institution by NAAC.
All eligible UG branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited
by NBA for academic years 2018-19 to 2021-22 & valid upto 30.06.2022
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru-560070, INDIA
Ph: 91-80-26711780/81/82 Email: principal@bnmit.in, www.bnmit.org

Department of Computer Science and Engineering

2021 - 2022

B.N.M. Institute of Technology

An Autonomous Institution under VTU,
Approved by AICTE, Accredited as Grade A Institution by NAAC.
All eligible UG branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited
by NBA for academic years 2018-19 to 2021-22 & valid upto 30.06.2022
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru-560070, INDIA
Ph: 91-80-26711780/81/82 Email: principal@bnmit.in, www.bnmit.org

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the project work entitled **Character Recognition using Neural Networks** carried out by Ms. **Anagha R** USN **1BG18CS008**, Mr**. Chandan Kumar G** USN **1BG18CS023**, are bonafide students of VII Semester, BNM Institute of Technology in partial fulfillment for the award of Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the year 2021-22. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Prof. Abhijit Das
Assistant Professor
Department of CSE
BNMIT, Bengaluru

Dr. Sahana D. Gowda
Professor and HOD
Department of CSE
BNMIT, Bengaluru

Dr.Krinshammurthy GN
Principal
BNMIT, Bengaluru

Examiner 1:

Examiner 2:

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

We would like to place on record our sincere thanks and gratitude to the concerned people, whose suggestions and words of encouragement has been valuable.

We would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMIT, Bengaluru for providing excellent academic environment in the college.

We would like to sincerely thank **Prof. T J Rama Murthy**, Director, BNMIT, Bengaluru for having extended his support and encouragement during the course of the work.

We would like to sincerely thank **Dr. S Y Kulkarni**, Assistant Director, BNMIT, Bengaluru for having extended his support and encouragement during the course of the work.

We would like to express my gratitude to **Prof. Eishwar N. Maanay**, Dean, BNMIT, Bengaluru for his relentless support, guidance, and assistance.

We would like to thank **Dr. Krishnamurthy G N**, Principal, BNMIT, Bengaluru for his constant encouragement.

We would like to thank **Dr. Sahana D. Gowda**, Professor and Head of the Department of Computer Science and Engineering who has shared her opinions and thoughts which helped me in giving my presentation successfully.

We would like to thank **Prof. Abhijit Das,** Associate Professor, Department of Computer Science and Engineering for their constant guidance and assistance in the project work.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, teaching & non-teaching staffs of the department and all our friends, for giving us valuable advice and support at all times in all possible ways.

Anagha R-1BG18CS008

Chandan Kumar G-1BG18CS023

ABSTRACT

Character recognition is a process by which computer recognizes letters, numbers or symbols and turn them into digital form that a computer can use. In today's environment character recognition has gained lot of concentration in the field of pattern recognition. Handwritten character recognition is useful in cheque processing in banks,form processing systems and many more. Character recognition is one of the well-liked and challenging area of research. In future character recognition create paperless environment. In this paper we describe the detail study on existing method for handwritten character recognition. We provide a literature review on various techniques used in offline english character recognition.

Due to its broad range of applications, handwritten character recognition is widespread. Processing application forms, digitizing ancient articles, processing postal addresses, processing bank checks, and many other handwritten character processing fields are increasing in popularity. Since the last three decades, handwritten characters have drawn the attention of researchers. For successful recognition, several methods have been suggested.

Chapter 1

INTRODUCTION

Handwritten Character Recognition (HCR) is a classic pattern recognition application, to begin with; Bezdek et al. described pattern recognition as recognizing structure in data by comparisons to known systems. The available system is created through a classification method. Handwritten character recognition, in general, is the method of classifying characters from handwritten input texts into predefined character groups. HCR has a wide range of applications, including character recognition, character recognition, character recognition, character recognition, character recognition, character recognition, character recognition, character recognition. Handwritten records are digitized. Reading the application form and making decisions based on the Unknown language's data is recognized and translated into a known language by a translation device. Blind people's reading aids, bank check handling Verification of signatures Number plates for automobiles To postal mail, for example, an automatic pin code reader is used.

Character awareness is something we do all the time in our everyday lives. Our brain constantly performs the HCR when reading notes, signs, or novels. We compare it to our previous experiences and memories and then respond, act, or infer new information. As a result, this is how we recognize characters naturally. During, who attempted to create an aid for the visually impaired, was the first to recognize characters. In the 1940s, the first character recognizer was built. Previously, almost all works were based on machine-printed text or a limited collection of handwritten symbols or texts.

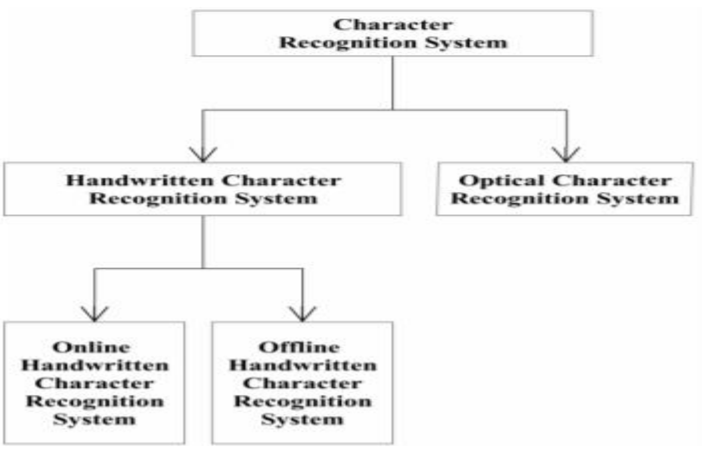

Figure 1.1: Character Recognition System Types

With online and offline methods, HCR quickly gained interest in research from 1980 to 1990. After 1990, image processing and pattern recognition combined with Artificial Intelligence, resulting in the development of highly efficient and powerful computers and gadgets such as scanners, cameras, and other specialized devices. Handwritten character recognition covers a significant portion of the application space. Despite all of this testing, there is still no device that genuinely achieves the goal of handwritten character recognition.

In general, image acquisition, preprocessing, segmentation, feature extraction, classification phases are included in all handwritten character recognition systems

Chapter 2

PROBLEM DEFENITION AND ALGORITHM

2.1 Task Definition

The main purpose of this scheme is to build a predictive model in recognizing the English Uppercase letters from external source as well as from the training and testing sets. It is necessary to recognize the characters in applications in electronic display devices for some readings.

The aim of the project is to build a predictive model to recognize the English uppercase letters.

2.2 Algorithm Definition

The choice of optimization algorithm for your deep learning model can mean the difference between good results in minutes, hours, and days.

1.Adam Optimizer:

The **Adam optimization algorithm** is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

When introducing the algorithm, the authors list the attractive benefits of using Adam on non-convex optimization problems, as follows:

- Straightforward to implement.
- Computationally efficient.
- Little memory requirements.
- Invariant to diagonal rescale of the gradients.
- Well suited for problems that are large in terms of data and/or parameters.
- Appropriate for non-stationary objectives.
- Appropriate for problems with very noisy/or sparse gradients.
- Hyper-parameters have intuitive interpretation and typically require little tuning.

Adam is different to classical stochastic gradient descent.Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training. A learning rate is maintained for each network weight (parameter) and separately adapted as learning unfolds. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients.

The authors describe Adam as combining the advantages of two other extensions of stochastic gradient descent. Specifically:

- **Adaptive Gradient Algorithm** (AdaGrad) that maintains a per-parameter learning rate that improves performance on problems with sparse gradients (e.g. natural language and computer vision problems).
- **Root Mean Square Propagation** (RMSProp) that also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight (e.g. how quickly it is changing). This means the algorithm does well on online and non-stationary problems (e.g. noisy).

Adam realizes the benefits of both AdaGrad and RMSProp.

Instead of adapting the parameter learning rates based on the average first moment (the mean) as in RMSProp, Adam also makes use of the average of the second moments of the gradients (the uncentered variance).

Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 control the decay rates of these moving averages.

The initial value of the moving averages and beta1 and beta2 values close to 1.0 (recommended) result in a bias of moment estimates towards zero. This bias is overcome by first calculating the biased estimates before then calculating bias-corrected estimates.

Chapter 3

EXPERIMENTAL EVAUATION

3.1 Methodology

Dataset:

The dataset contains 26 folders (A-Z) containing handwritten images in size 28*28* pixels, each alphabet in the image is centre fitted to *20*20 pixel box.Each image is stored as Gray-level Kernel CSV*To*Images contains script to convert .CSV file to actual images in .png format in structured folder. It might contain some noisy image as well

Data sets link: https://www.kaggle.com/sachinpatel21/az-handwritten-alphabets-in-csv-format

Implementation:

- First of all, we make the necessary import statements. To implement our model, we need the following frameworks:

    1. Numpy (version 1.16.5)
    2. cv2 (openCV) (version 3.4.2)
    3. Keras (version 2.3.1)
    4. Tensorflow (Keras uses TensorFlow in backend and for some image preprocessing) (version 2.0.0)
    5. Matplotlib (version 3.1.1)
    6. Pandas (version 0.25.1)

- **Read the data**: Now we are reading the dataset using the **pd.read_csv()** and printing the first 10 images using **data.head(10)**
- **Split data into images and their labels:**Splitting the data read into the images & their corresponding labels. The '0' contains the labels, & so we drop the '0' column from the data dataframe read & use it in the y to form the labels.
  In the above segment, we are splitting the data into training & testing dataset using train_test_split(). Also, we are reshaping the train & test image data so that they can be displayed as an image, as initially in the CSV file they were present as 784 columns of pixel data. So we convert it to 28×28 pixels. All the labels are present in the form of floating point values, that we convert to integer values, & so we create a dictionary word_dict to map the integer values with the characters.

- **Plotting the number of alphabets in the dataset:** Here we are only describing the distribution of the alphabets. Firstly we convert the labels into integer values and append into the count list according to the label. This count list has the number of images present in the dataset belonging to each alphabet. Now we create a list – alphabets containing all the characters using the values() function of the dictionary. Now using the count & alphabets lists we draw the horizontal bar plot.
- **Shuffling the data:** Now we shuffle some of the images of the train set. The shuffling is done using the shuffle() function so that we can display some random images. We then create 9 plots in 3×3 shape & display the thresholded images of 9 alphabets.
- **Data Reshaping**: Now we reshape the train & test image dataset so that they can be put in the model.
- **Compiling and Fitting Model**: Here we are compiling the model, where we define the optimizing function & the loss function to be used for fitting.The optimizing function used is Adam, that is a combination of RMSprop & Adagram optimizing algorithms.The dataset is very large so we are training for only a single epoch, however, as required we can even train it for multiple epochs (which is recommended for character recognition for better accuracy)
- **Getting the train validation accuracies and losses**.
- **Doing some predictions on test data**: Here we are creating 9 subplots of (3,3) shape & visualize some of the test dataset alphabets along with their predictions, that are made using the **model.predict()** function for text recognition

- **Doing Prediction on external image**: Here we have read an external image that is originally an image of alphabet 'B' and made a copy of it that is to go through some processing to be fed to the model for the prediction that we will see in a while. The img read is then converted from BGR representation (as OpenCV reads the image in BGR format) to RGB for displaying the image, & is resized to our required dimensions that we want to display the image in. We convert the image from BGR to grayscale and apply thresholding to it. We don't need to apply a threshold we could use the grayscale to predict, but we do it to keep the image smooth without any sort of hazy gray colors in the image that could lead to wrong predictions. The image is to be then resized using **cv2.resize()** function into the dimensions that the model takes as input, along with reshaping the image using **np.reshape()** so that it can be used as model input.

3.2 Results

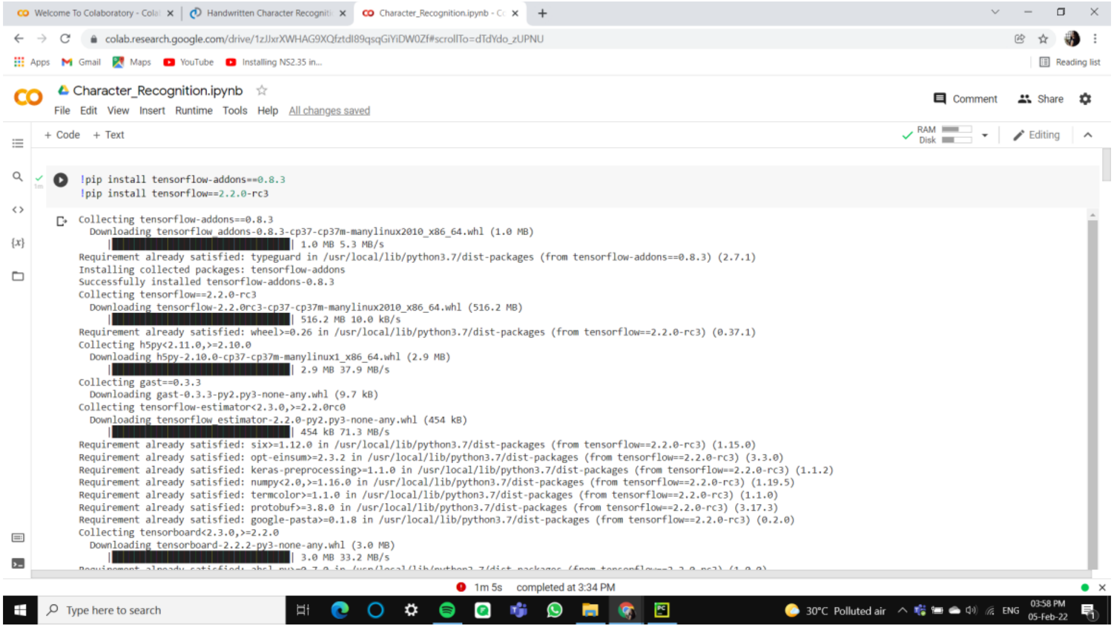

Fig 3.1: Output 1

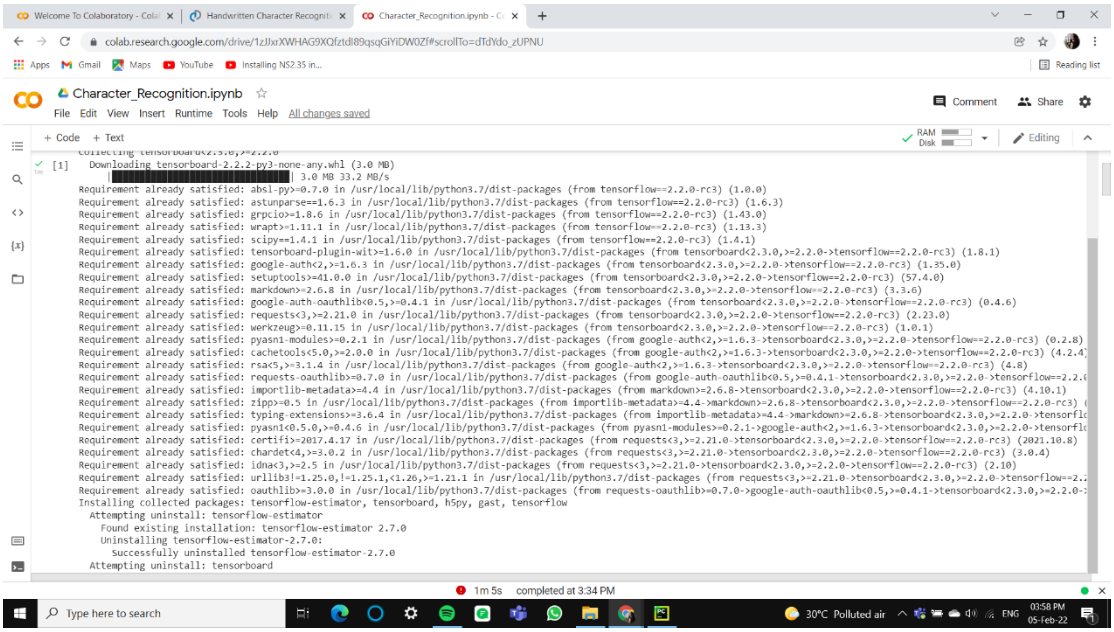

Fig 3.2: Output 2

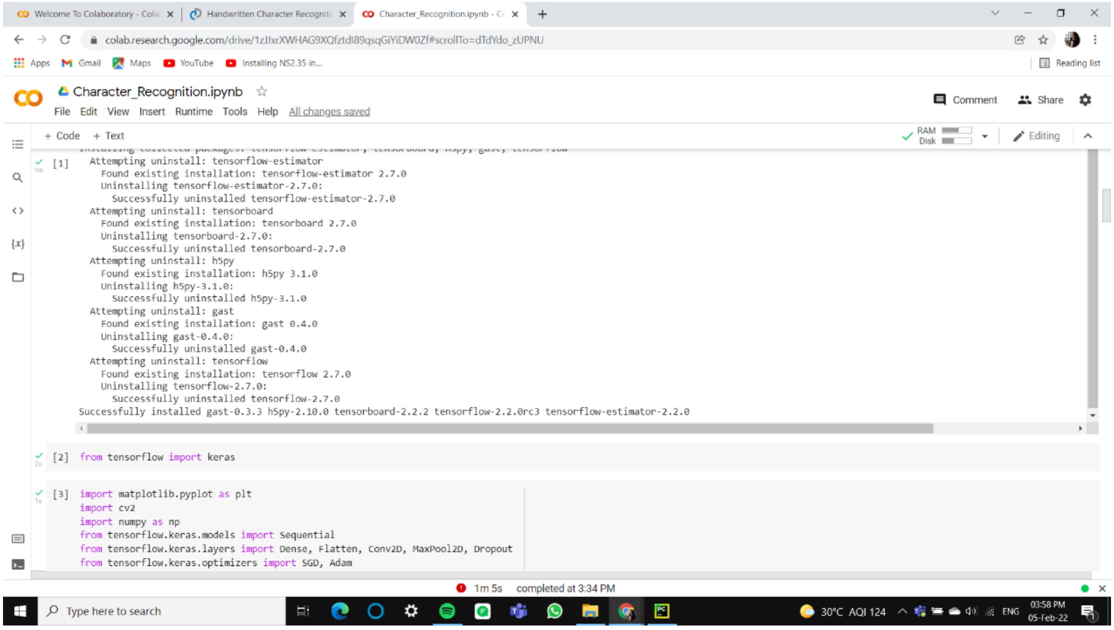

Fig 3.3: Output 3

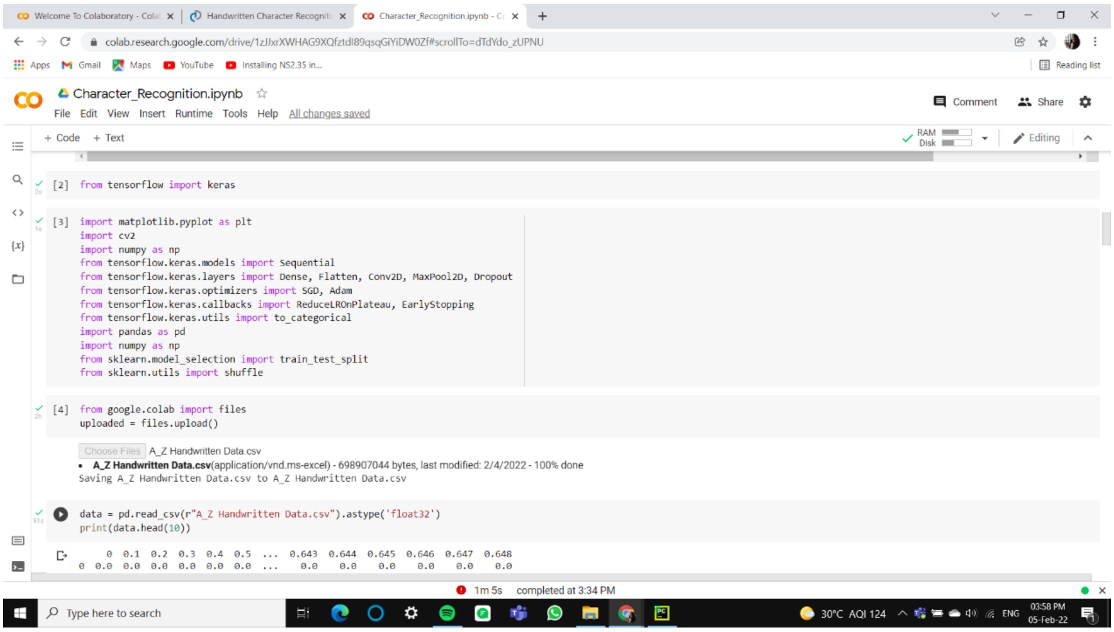

Fig 3.4: Output 4

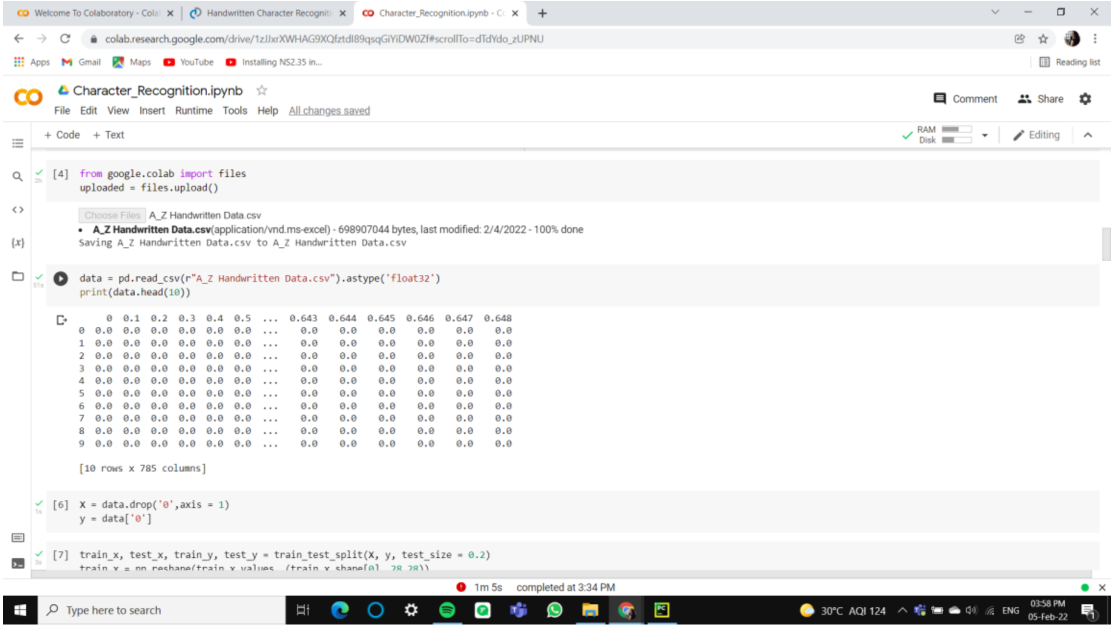

Fig 3.5: Output 5

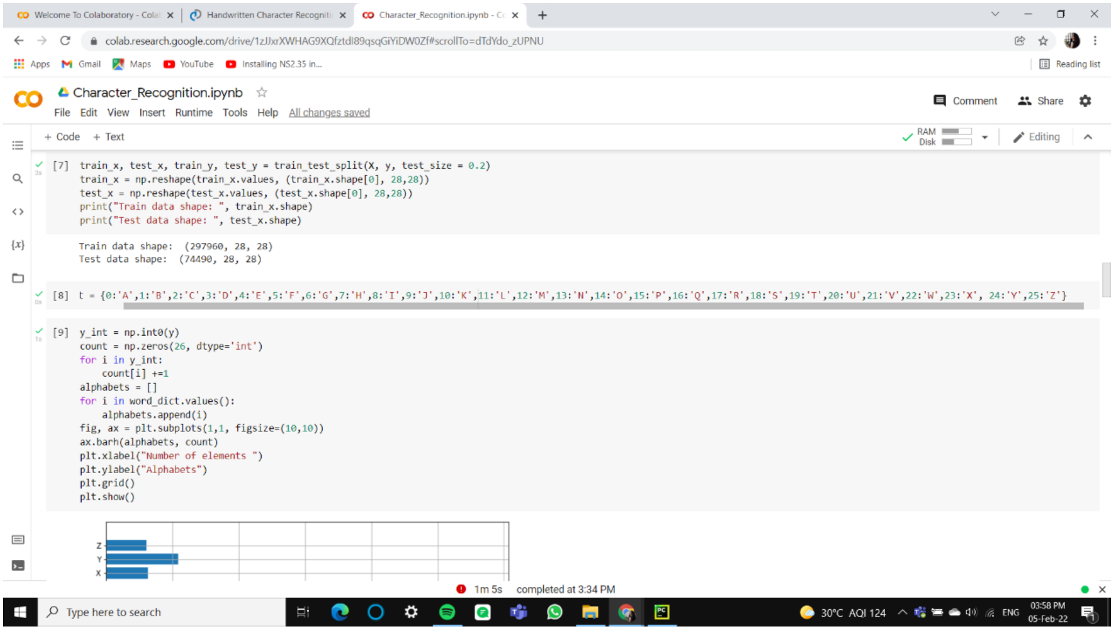

Fig 3.6: Output 6

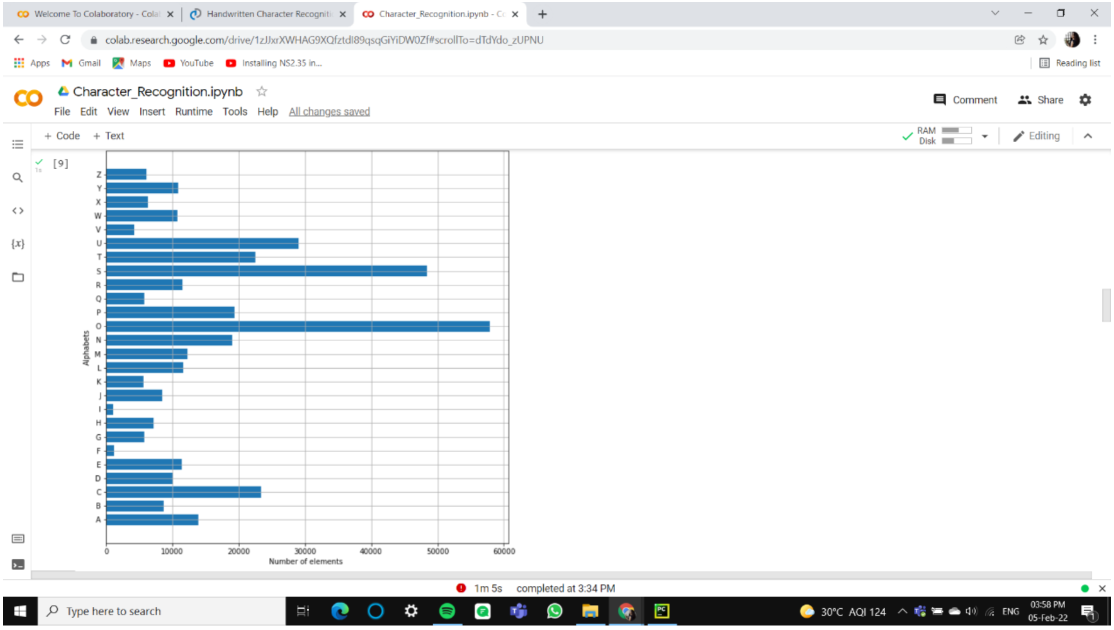

Fig 3.7: Output 7

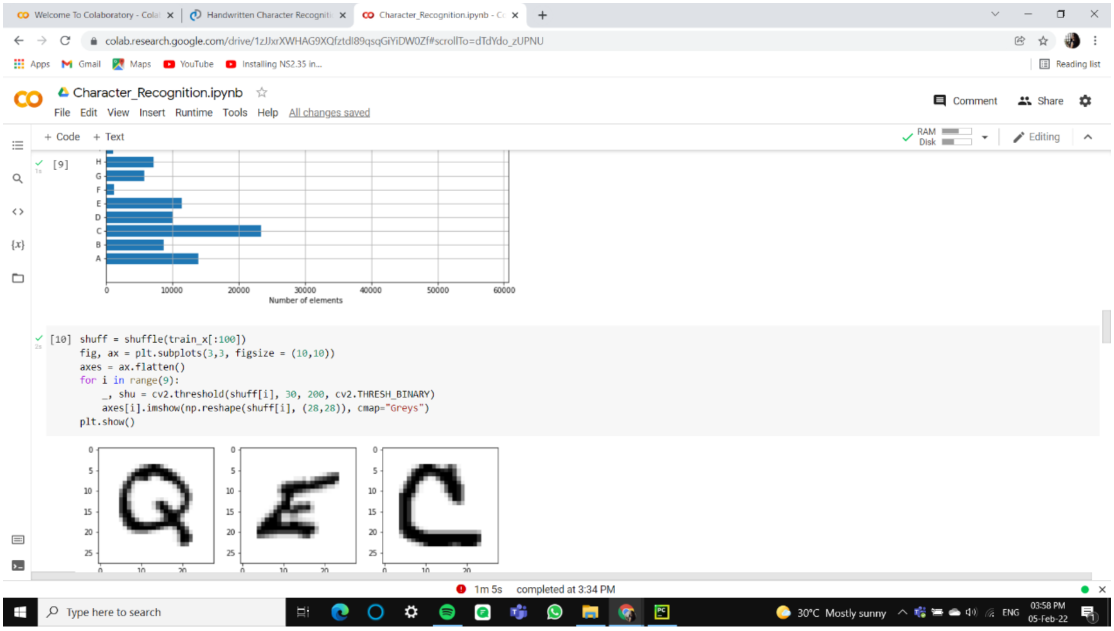

Fig 3.8: Output 8

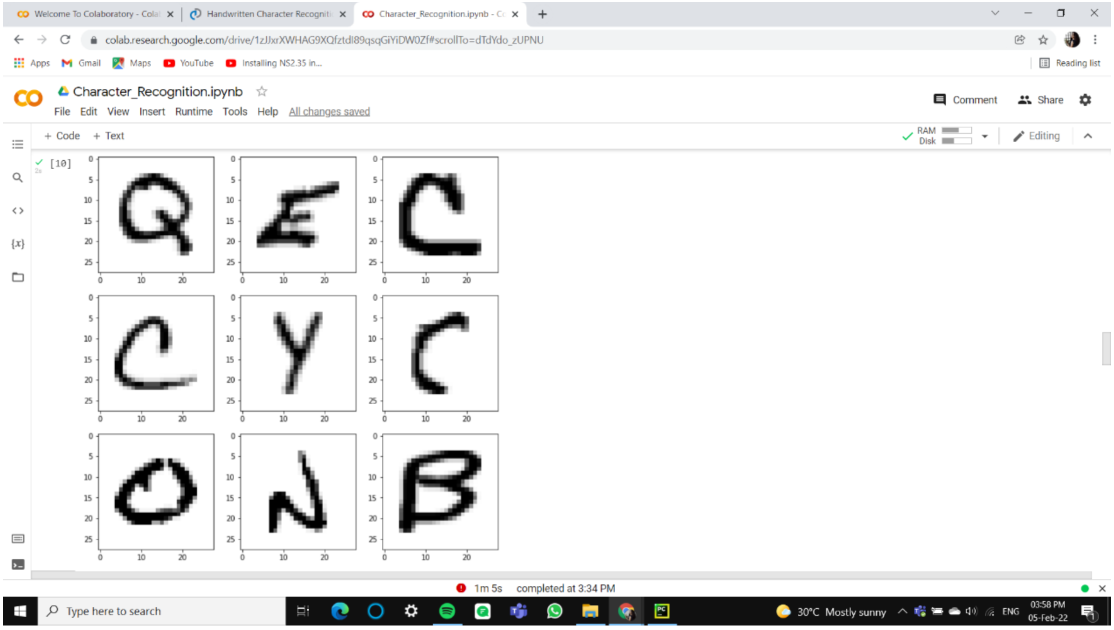

Fig 3.9: Output 9

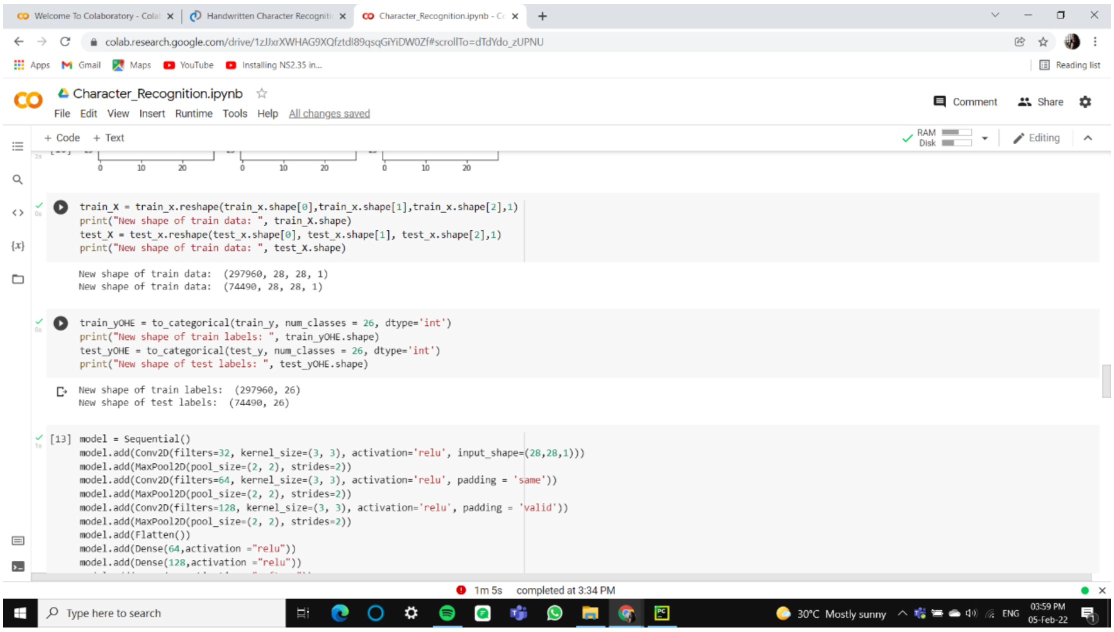

Fig 3.10: Output 10

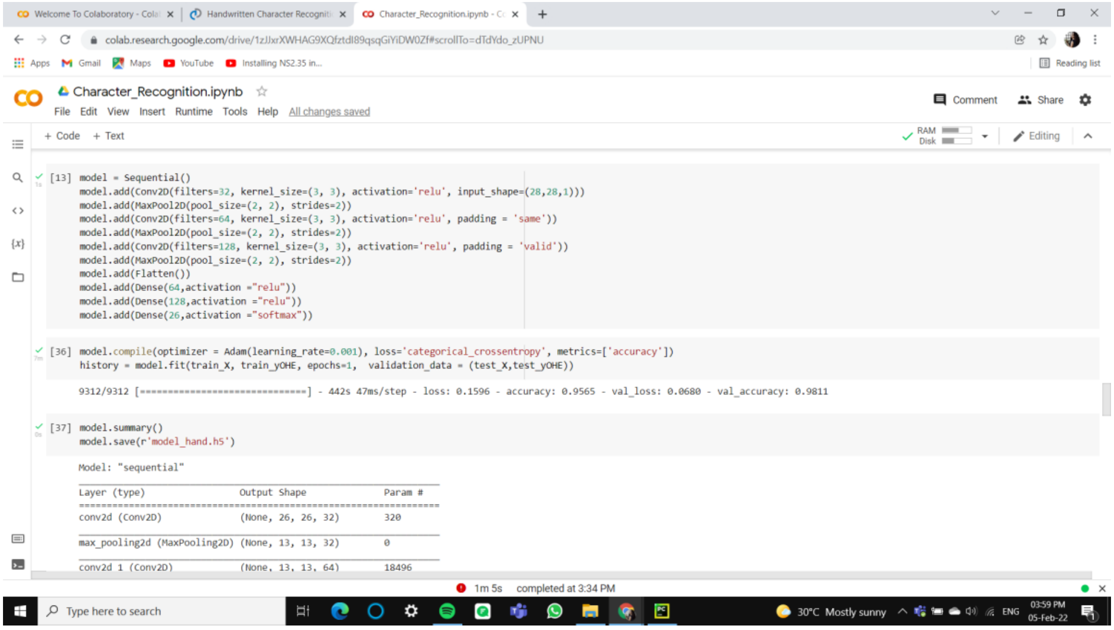

Fig 3.11: Output 11

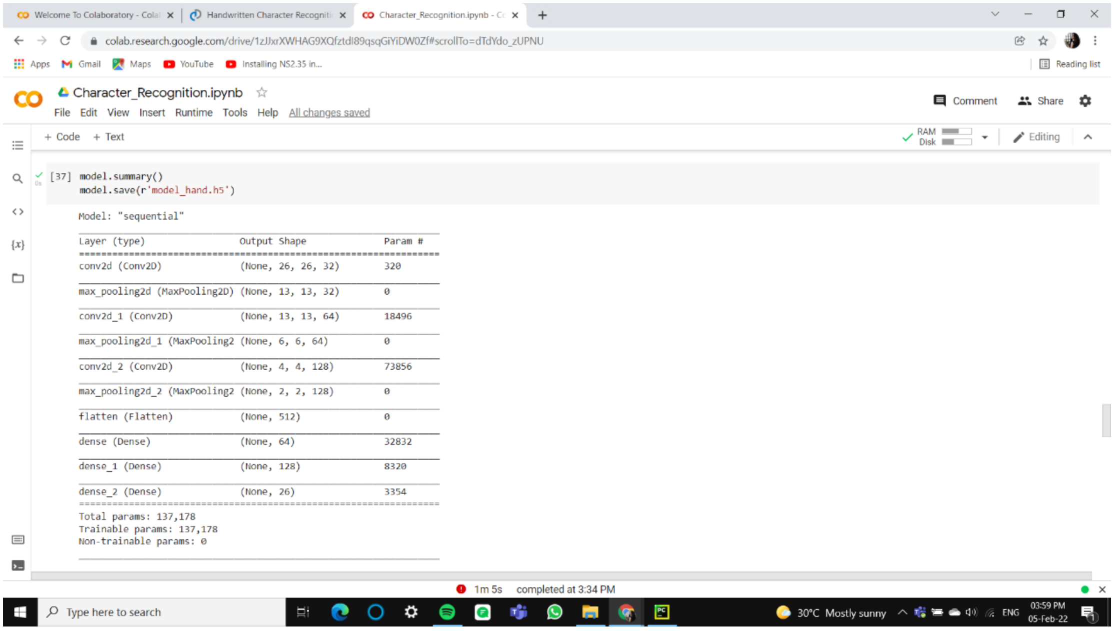

Fig 3.12: Output 12

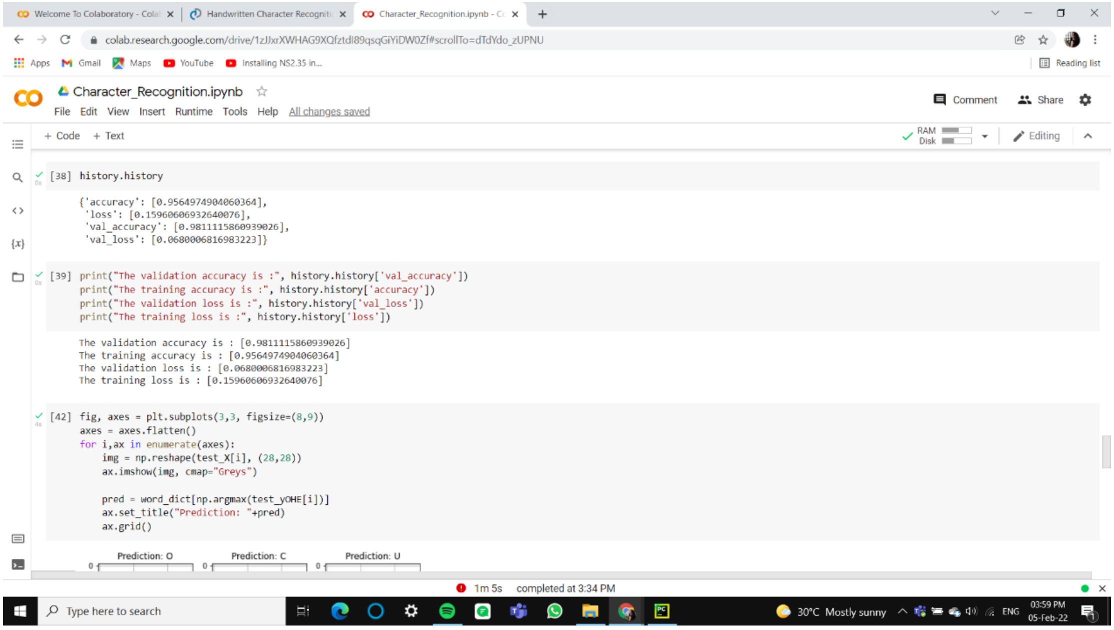

Fig 3.13: Output 13

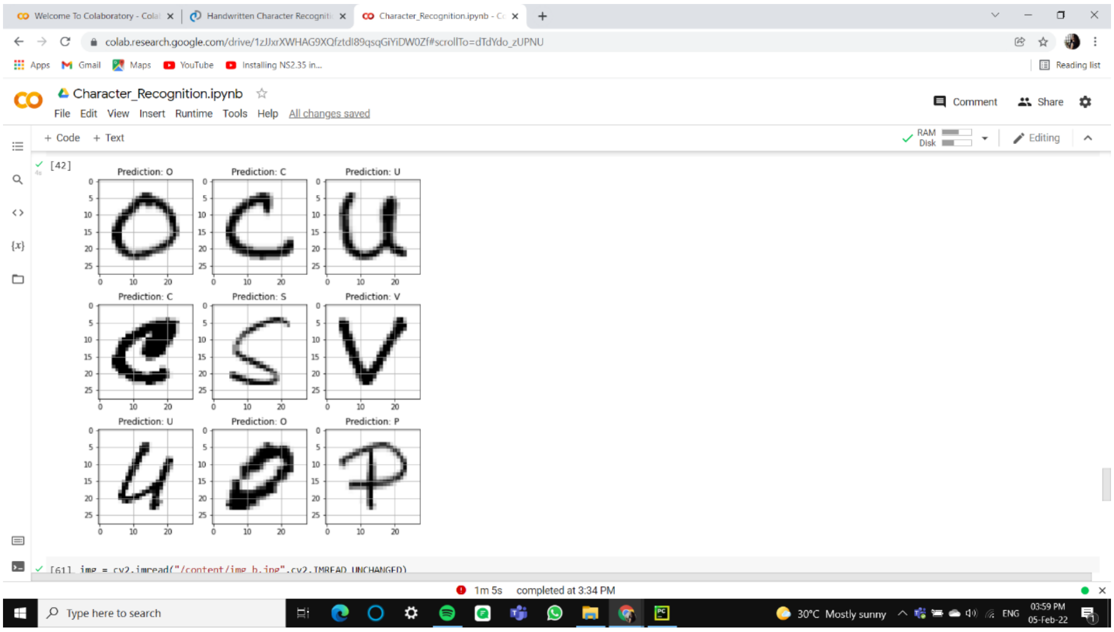

Fig 3.14: Output 14

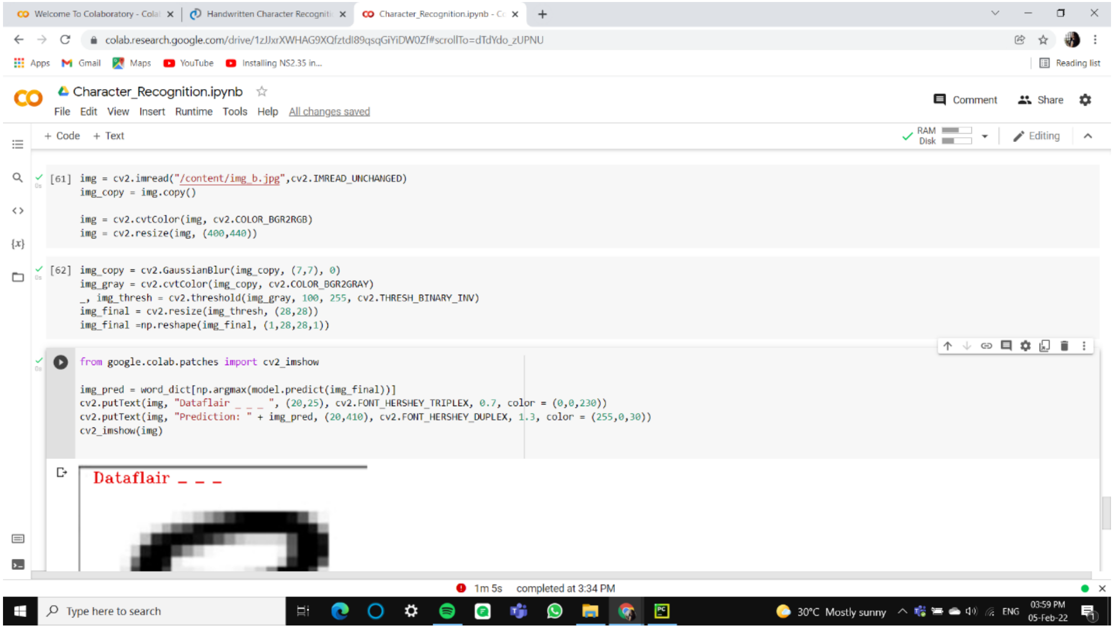

Fig 3.15: Output 15

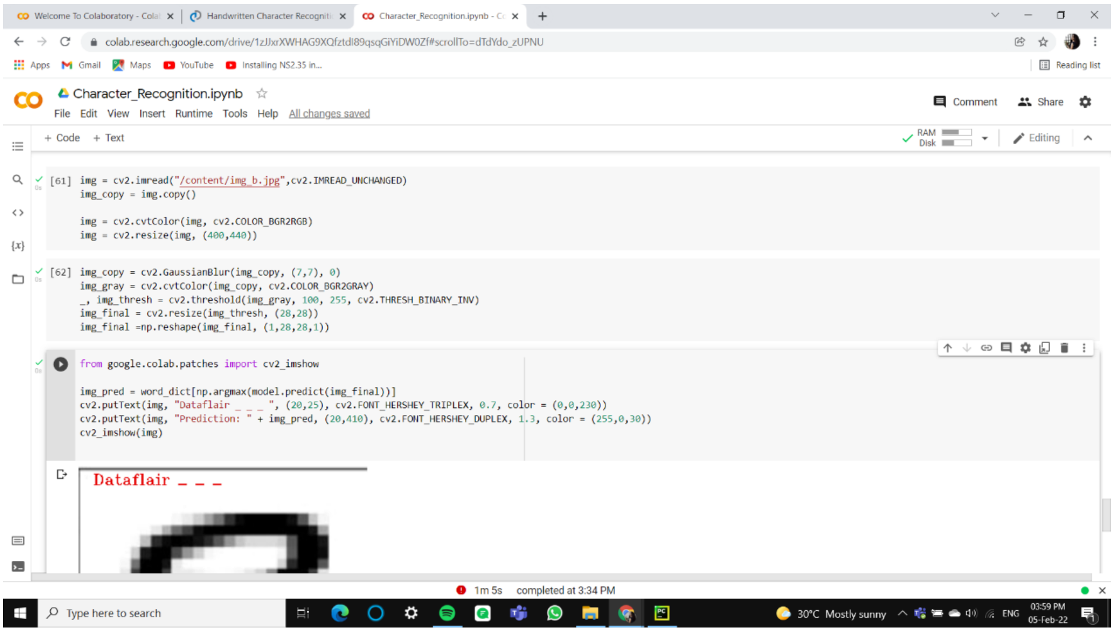

Fig 3.16: Output 16

3.3 Discussion

We have successfully developed Handwritten character recognition (Text Recognition) with Python, Tensorflow, and Machine Learning libraries.

Handwritten characters have been recognized with more than 97% test accuracy. This can be also further extended to identifying the handwritten characters of other languages too.

Chapter 4

RELATED WORK

Below table summarizes on the various works on character recognition methods by various people and their implementation method.

Sl No.	Author	Method	Classifier	Accuracy(%)
1.	Anshul Gupta, Manisha Srivastava and Chitralekha Mahanta	Support vector machine	Neural Network	98.86(trainin g)
				62.93(test)
2.	Aiquan Yuan, Gang Bai, Lijing Jiao and Yajie Liu	LetNet 5	Convolution al neural network	93.7(upper case)
				90.2(lower case)
3.	J.Pradeep, E.Srinivasan and S.Himavathi	Hybrid feature extraction	Feed forward	95.96
			NN Radial base function	93.82
			NN Nearest Neighbour	91.88
4.	N.M Noor,M.Razaz and P.Mahley	Geometry extraction	Geometric density	77.89
			Geometric feature	76.44
5	Rajib Lochan Das, Binod Kumar Prasad and Goutam Sanyal	Local and global feature extraction	HMM	98.26
6.	Rakesh Kumar Mandal and N R Manna	Row wise segmentation	Single ANN	80
7.	Huiqin Lin,Wennuan Ou,Tonglin Zhu	Direction element feature	Direction element feature	99.03
8	D.K.Patel,T.Som,M anoj Kumar Singh	Ecludian distance	ANN	92.31
9	Huihang Zaho,Dejian Zhou,Zhaohua Wu	Surface mount technology	BP neural network	98.6
10.	Peng Xu	Particle swarm method	PSO-BP neural network	86.8(Capital)
				85.3(Normal)
11.	P.M. Patil,P.S. Dhabe, U.V. Kulkarni and T.R. sontakke	Hyperline segment	FHLSSN	72.1
			MFHLSSN	72.55

The method proposed in this project has an accuracy of 97% and can further extended to be applied on other language characters too.

Chapter 5

FUTURE WORK

The method used to recognize characters in the project has given a better accuracy and results. It can be tested on external images too and not just the data sets which is divided as training sets and testing sets.

The advancements that can be incorporated to this project is:

- It can be applied to other regional languages.
- It can help recognizing the offline signatures for verification of fraudulent activities.
- It can also be used for online character recognitions.
- Can be helped to decode character messages with the prediction model.
- Can be used to encrypt and decrypt messages in important domains for maintenance of secrecy.

Chapter 6

CONCLUSION

The character recognition methods have developed amazingly in the last decade. A variety of techniques have emerged, influenced by developments in related fields such as image recognition and face recognition. In this paper we provide review of various techniques used in offline handwritten character recognition. These techniques provide better accuracy by use of different classifier. This review provides information about different classifier used in character recognition techniques. This comprehensive discussion will provide insight into the concepts involved, and perhaps provoke further advances in the area. The promise for the future is significantly higher performance for almost every character recognition technology area.

BIBLIOGRAPHY

- Aiquan Yuan, Gang Bai, Po Young, Yanni Guo, Xintig Zhao,"Handwritten english word recognition based on convolution neural networks", 2012 International conference on trontiers in handwriting recognition.
- Pranab k Charles,V. Harsh, M. Swath, CH. Deepth,"A review on the various techniques used for opticle character recognition", International journal of engineering research and application, vol.2, pp.659-662, jan-Feb 2012 .
- R. Plamondon and S.N. Srihari,"Online and offline handwritten character recognition: A comprehensive survey", IEEE transaction on pattern analysis and machine intelligence, vol22, no1, pp-63-84,2000.
- Nafiz Arica and Fatos T. Yarman," An overview ocharacter recognition focused on offline handwriting", IEEE transaction on system. Man and Cybeznetic, vol.31.no.2. May 2011.
- J.Pradeep, E.Srinivasan, S.Himawathi, "Perfomance analysis of hybrid feature extraction technique for recognizing english handwritten character",978-1-4673-4805-8/12/2012IEEE.
- Anshul Gupta, Manish Srivastva, Chitraleka Mahanta,"Offline handwritten character recognition using neural network", 2011 International conference on computer application and industrial electronics.
- Aiquan yuan, Gang Bai, Lijing Jiao, Yajie Liu,"Offline handwritten english character recognition based on convolution neural network, 2012, 10th IAPR International workshop on document analysis system.