A Novel Method for Handwritten Digit Recognition System

USING AI

A Project report submitted in partial fulfilment of 7th semester in degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

Team ID: PNT2022TMID43993

BRINDHA.M (723719104016)

DEEPIKA.G.S (723719104018)

LAVANYA (723719104043)

MATHU MITHA.C (723719104045)

ANUDHARSHINI.R (723719104008)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600025

NOV-2022

VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS, COIMBATORE
(Affiliated College of Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report ”**A NOVEL METHOD FOR HANDWRITTEN RECOGNITION SYSTEM**” is the Bonafide record work done by **MS.BRINDHA.M**(723719104016),**MS.LAVANYA**(723719104043),**MS.ANUDHARSHINI**(723719104008),**MS.MATHUMITHA.C**(723719104045),**MS.DEEPIKA.G.S**(723719104018)for **IBM- NALAIYATHIRAN** in **VII** semester of B.E., degree course in **Computer Science and Engineering** branch during the academic year of 2022 - 2023.

STAFF INCHARGE

Ms. S. Dhrisya,M.E.,

EVALUATOR

Mr.B.Mari Kumar,M.E.,

HEAD OF THE DEPARTMENT

Mr. Dinesh Kumar P,M.E.,

ACKNOWLEDGEMENT

We express our breathless thanks to our **Dr.Velmurugan**., the principal(i\c), **VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS, COIMBATORE**, for giving constant motivation in succeeding in our goal.

We acknowledge our sincere thanks to Head of the Department (i/c) **Mr. P. Dinesh Kumar** for giving us valuable suggestion and help towards us throughout this Project.

We are highly grateful to thank our Project coordinator **DHRISYA.S** and our Project Evaluator **MARIKUMAR.B** ,**VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS,COIMBATORE** the coordinating us throughout this Project.

We are very much indebted to thank all the faculty members of Department of Computer science and Engineering in our Institute, for their excellent moral support and suggestions to complete our Project work successfully.

Finally, our acknowledgment does our parents, sisters, and friends those who had extended their excellent support and ideas to make our Project a pledge one.

BRINDHA.M

LAVANYA

ANUDHARSHINI.R

MATRHU MITH.C

DEEPIKA.G.S

ABSTRACT

The Handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person. The similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 7 etc. So, classifying between these numbers is also a major problem for computers. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits. Digit recognition plays an important role in the modern world. It can solve more complex problems and makes human job easier. This type of system can be widely used in the world to recognize zip code or postal code for mail sorting In Banking Sector too where more handwritten numbers are involved like account number, figure of cash and checks. Postal department and courier services can easily find the digits written. Old people who will have eye sight issues with handwritten digits. Baking sector and Postal sector by providing the services. Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. This image is analysed by the model and the detected result is returned on to UI.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Category: Artificial Intelligence

Skills Required: Python,CNN,IBM Cloud,

IBM Watson Studio,IBM Cloudant DB,Deep Learning,Python-Flask

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer applied science and artificial intelligence. With the use of deep learning , machine learning can be reduced in recognition, predictions and many more areas.

Handwritten Digit Recognition is the ability of computer systems to recognize handwritten digits from various sources, such as images, documents, among other examples. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

1.2 PURPOSE

Digit recognition systems are capable of recognizing the digits from different sources like emails,bank quench, papers,images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank quench amounts, numeric entries in forms filled up by hand (tax forms) and so on.

CHAPTER 2

LITERATURE SURVEY

TITLE : Handwritten Character Recognition

AUTHOR : Ayush Purohit

DESCRIPTION :

Handwriting recognition has gained a lot of attention in the fieldof pattern recognition and machine learning due to its application in various fields. Optical Character Recognition (OCR) and Handwritten Character Recognition (HCR) has specific domain to apply. Various techniques have been proposed to for character recognition in handwriting recognition system. Even though, sufficient studies and papers describes the techniques for converting textual content from a paper document into machine readable form. In coming days, character recognition system might serve as a key factor to create a paperless environment by digitizing and processing existing paper documents.

PUBLISHED IN : 2016

TITLE : Automatic Handwritten Digit Recognition

AUTHOR : Akkireddy Challa

DESCRIPTION :

The main purpose of this thesis is to build an automatic handwritten digit recognition method for the recognition of connected handwritten digit strings. To accomplish the recognition task, first, the digits were segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. In this study, different machine learning methods, which are SVM, ANN and CNN architectures are used to achieve high performance on the digit string recognition problem. In these methods, images of digit strings are trained with the SVM, ANN and CNN model with HOG feature vectors and Deep learning methods structure by sliding a fixed size window through the images labeling each sub-image as a part of a digit or not. After the completion of the segmentation, to achieve the complete recognition of handwritten digits.

PUBLISHED IN : 2019

TITLE : Handwritten Numeral Recognition

AUTHOR : Stuti Asthana

DESCRIPTION :

An extensive literature review on Neural Network based numeric recognition by describing the survey of some research articles have been involved for better analysis in order to enhance the system efficiency. Handwritten Numeric Recognition is very interesting area of Pattern Recognition and it deals with Offline Handwriting Recognition. Handwriting Recognition has kept on continuing as a method for correspondence, gathering, recording and transmitting data in everyday life since the hundreds of years even with the appearance of the new advancements. Machine recognition has numerous functional applications, perusing manually written postal envelopes, sum written in bank checks, bill handling, government records, business frames, signature confirmation, disconnected from the net archive acknowledgment and so on. This Paper portrays the bestin class study of the work accomplished for the Numeric recognition.

PUBLISHED IN : 2017

TITLE : Neural Network Based Handwritten Digit Recognition

AUTHOR : Ankit Sharma

DESCRIPTION :

Recognition of handwritten character is a difficult task in the field of image processing, artificial intelligence since the handwriting varies from person to person. In proposed paper, we are training the neural network to recognize the off-line strategies for the isolated handwritten character (0 to 9). This work improves the character recognition and pre-processing of the Character is done by image rendering, character extraction and training and testing steps. The proposed method is based on the use of linear regression algorithm to classify the characters and is used to train the given dataset. After training a network performance curve is generated along with the individual required characters. In given system, numerical character is represented by binary numbers that are used as input then they are fed to an ANN. Neural network followed by the linear regression, algorithm which compromises Training.

PUBLISHED IN : 2016

TITLE : Handwritten Optical Character Recognition

AUTHOR : Jamshed Memon

DESCRIPTION :

Given the ubiquity of handwritten documents in human transactions, Optical Character Recognition (OCR) of documents have invaluable practical worth. Optical character recognition is a science that enables to translate various types of documents or images into analyzable, editable and searchable data. During last decade, researchers have used artificial intelligence/machine learning tools to automatically analyze handwritten and printed documents in order to convert them into electronic format. The objective of this review paper is to summarize research that has been conducted on character recognition of handwritten documents and to provide research directions. In this Systematic Literature Review (SLR) we collected, synthesized and analyzed research articles on the topic of handwritten OCR (and closely related topics) which were published between year 2000 to 2019. We followed widely used electronic databases by following pre-defined review protocol. Articles were searched using keywords, forward reference searching and backward reference searching in order to search all the articles related to the topic. After carefully following study selection process 176 articles were selected for this SLR. This review article serves the purpose of presenting state of the art results and techniques on OCR and also provide research directions by highlighting research gaps.

PUBLISHED IN : 2020

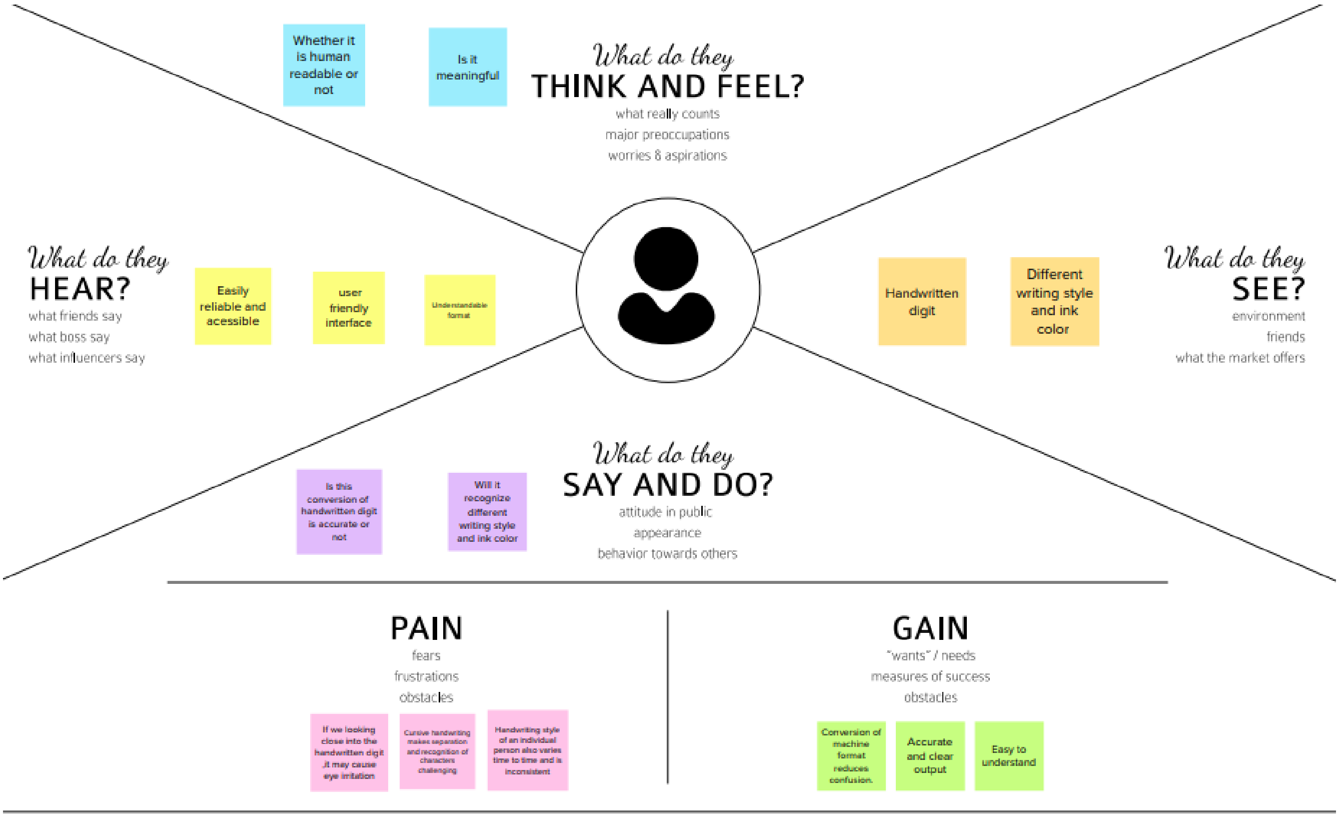
2.3PROBLEM STATEMENT DEFINITION

For years,the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible because for the average individual to write down the license plate of a reckless driver. Therefore, the goal of this project is to help the traffic department identify these offenders and reduce traffic violations as a result.

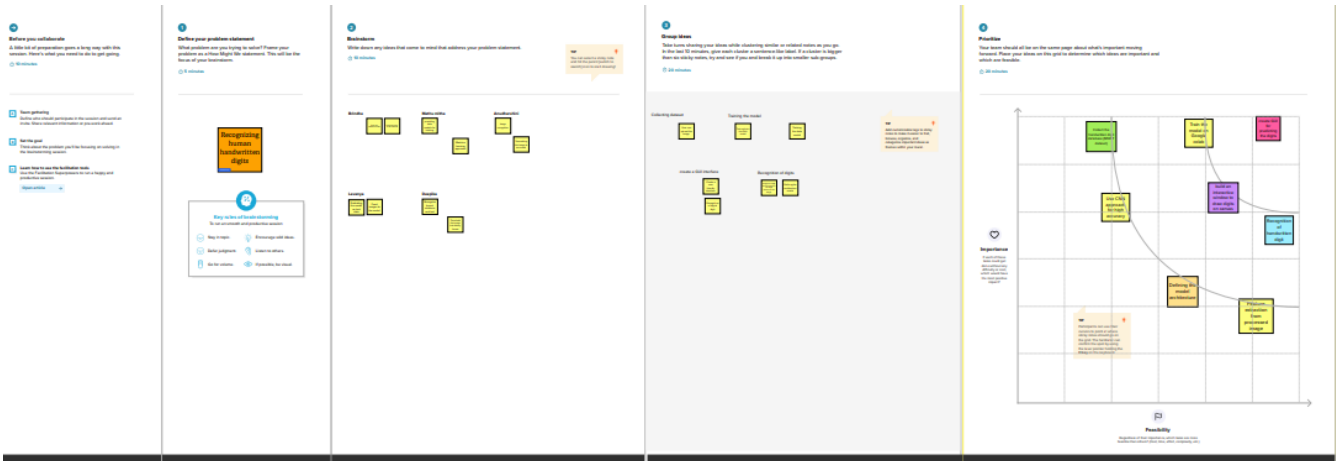
CHAPTER-3

IDEATION AND PROPOSEDSOLUTION

3.1 EMPATHY MAP CANVAS


Whether it is human readable or not
Is it meaningful
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
Easily reliable and acessible
user friendly interface
Understandable format
Handwritten digit
Different writing style and ink color
What do they
SEE?
environment
friends
what the market offers
Is this conversion of handwritten digit is accurate or not
Will it recognize different writing style and ink color
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
If we looking close into the handwritten digit ,it may cause eye irritation
Cursive handwriting makes separation and recognition of characters challenging
Handwriting style of an individual person also varies time to time and is inconsistent
GAIN
"wants" / needs
measures of success
obstacles
Conversion of machine format reduces confusion.
Accurate and clear output
Easy to understand

3.2 **IDEATION & BRAINSTORMING**


Before you collaborate
Define your problem statement
Recognizing human handwritten digits
Key rules of brainstorming
Brainstorm
Group ideas
Prioritize
Importance
Feasibility
Use CNN approach for high accuracy
Train the model on Google colab
Build an interactive window to draw digits on canvas
Recognition of handwritten digit
Defining the model architecture
Pixel extraction from processed image

3.3PROPOSED SOLUTION

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

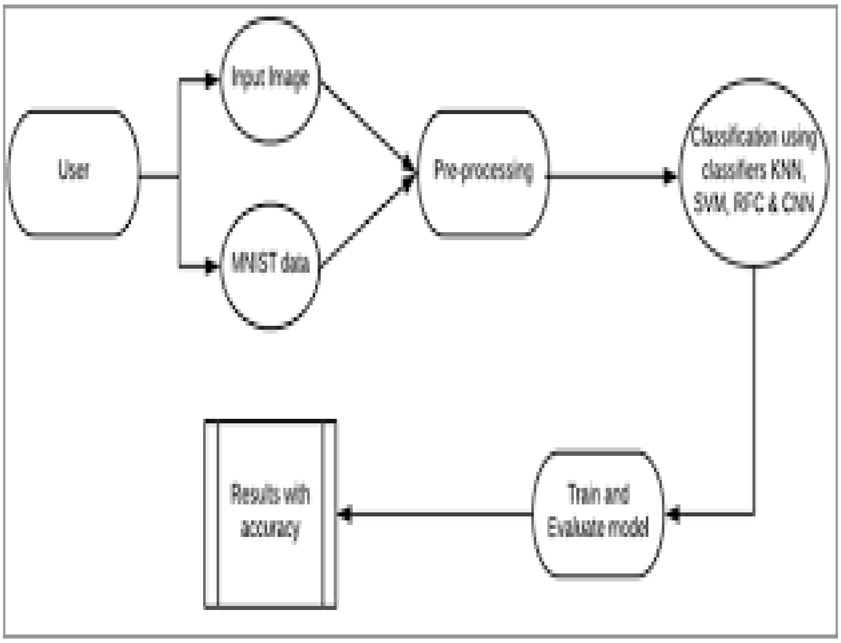
PROBLEM STATEMENT:

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits.

It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using **PyTorch library** over the **MNIST dataset** to **recognize handwritten digits**.

IDEA/SOLUTION DESCRIPTION:

- MNIST database contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing.
- We will create our CNN model. It works better for data that are represented as grid structures; this is the reason why CNN works well for image classification problems.


User
Input Image
MNIST data
Pre-processing
Classification using classifiers KNN, SVM, RFC & CNN
Train and Evaluate model
Results with accuracy

NOVELTY/UNIQUENESS:

- ✓ Handwritten digit recognition using MNIST dataset is a major project madewith the help of neural network. It basically detects the scanned images of handwritten digits.
- ✓ We have taken this a step further where are handwritten digit recognitionsystem not only detects the scanned images of handwritten digits but alsoallows writing digits on the screen with the help of an Integrated GUI forrecognition.

SOCIAL IMPACT/CUSTOMER SATISFACTION:

Digital Recognition is nothing other than recognizing or identifying digits in any document. The framework of digital recognition is simply the operation of the machine to prepare or interpret digits. Handwritten Digit Recognition is the power ofcomputers to translate handwritten digits from a variety of sources such as text messages, bank checks, papers, photos, etc. method

With the use of in-depth learning methods, human efforts can be reduced in perception, learning, perception and in too many regions. Using in-depth learning, the computer learns to perform distinctive functions in images or content anywhere accuracy, in addition to the performance of the human level. The digital recognitionmodel uses large data sets to detect digits from different sources.

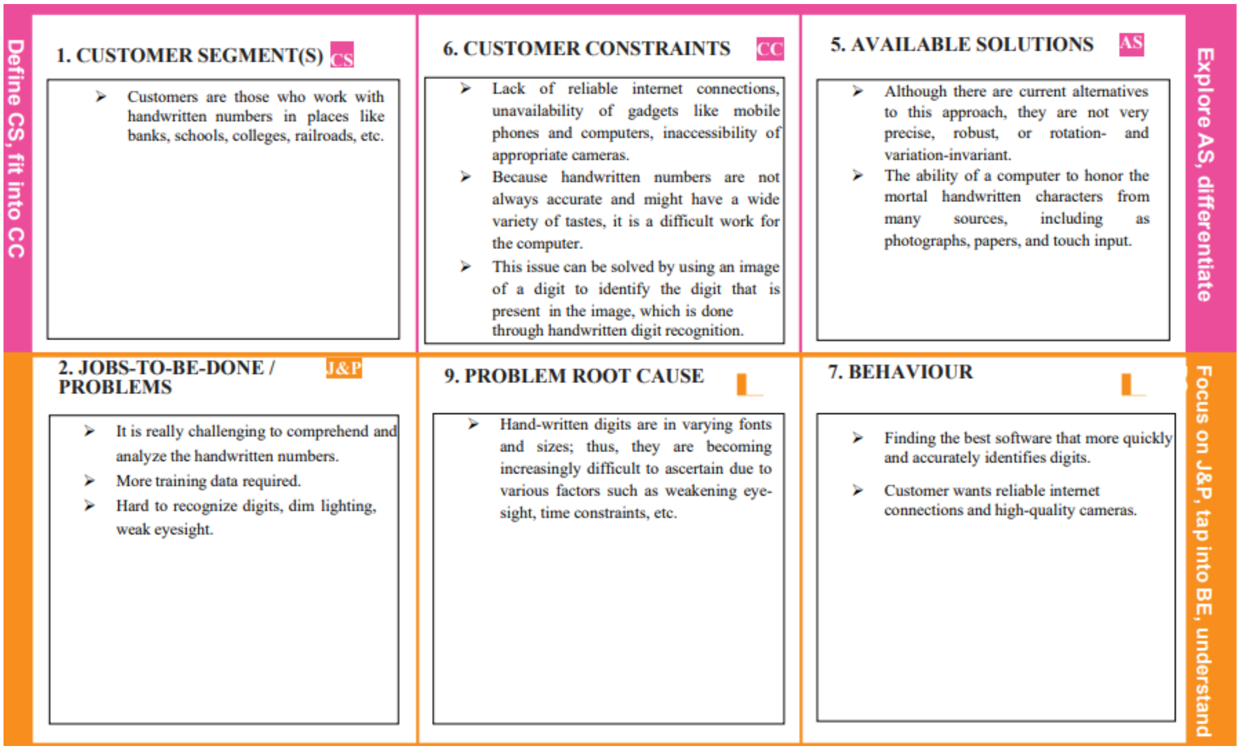
BUSINESS MODEL (FINANCIAL BENEFIT):

- ✓ Handwritten digit recognition refers to a model's (machine's) capacity to detectany handwritten digits from various sources, such as photographs, papers, and touch displays, and classify them into ten specified categories 0-9.
- ✓ Several ways and algorithms are used to recognize handwritten digits, such as Deep Learning/CNN, SVM (Support Vector Machine), Gaussian Naive Bayes,KNN (K-Nearest Neighbour), Decision Trees, Random Forests, etc.
- ✓ We used the CNN (Convolutional Neural network) algorithm to recognize handwritten digits in this project.

SCALABILITY OF SOLUTION:

- ✓ The variations of accuracies for handwritten digit were observed for 15 epochsby varying the hidden layers using CNN model and MNIST digit dataset.
- ✓ The maximum accuracy in the performance was found 99.64% and the totallowest test loss is 0.0239 approximately.

3.4 **PROBLEM SOLUTION FIT**



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Customers are those who work with handwritten numbers in places like banks, schools, colleges, railroads, etc.

6. CUSTOMER CONSTRAINTS CC

- Lack of reliable internet connections, unavailability of gadgets like mobile phones and computers, inaccessibility of appropriate cameras.
- Because handwritten numbers are not always accurate and might have a wide variety of tastes, it is a difficult work for the computer.
- This issue can be solved by using an image of a digit to identify the digit that is present in the image, which is done through handwritten digit recognition.

5. AVAILABLE SOLUTIONS AS

- Although there are current alternatives to this approach, they are not very precise, robust, or rotation- and variation-invariant.
- The ability of a computer to honor the mortal handwritten characters from many sources, including as photographs, papers, and touch input.

Explore AS, differentiate

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- It is really challenging to comprehend and analyze the handwritten numbers.
- More training data required.
- Hard to recognize digits, dim lighting, weak eyesight.

9. PROBLEM ROOT CAUSE

- Hand-written digits are in varying fonts and sizes; thus, they are becoming increasingly difficult to ascertain due to various factors such as weakening eye-sight, time constraints, etc.

7. BEHAVIOUR

- Finding the best software that more quickly and accurately identifies digits.
- Customer wants reliable internet connections and high-quality cameras.

Focus on J&P, tap into BE, understand

CHAPTER – 4

REQUIREMENT ANALYSIS

Solution Requirements (Functional & Non-functional)

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Model Creation	• Get access the MNIST dataset • Anaylze the data set • Define a CNN model • Train and test the model •
FR-2	Application Development	• Create a website to let the user recognize handwritten digits. • Create a homepage to upload images. • Create a result page to display the results.
FR-3	Input image upload	• Let users upload the images of various formats. • Let users upload the images of various sizes. • Prevent users from uploading unsupported image formats. • Pre-process the image to use it on the model. • Create a database to store all the input images.
FR-4	Display results	• Display the result form result. • Display input image. • Display accuracy of the result.

4.2 NON-FUNCTIONAL REQUIREMENTS:

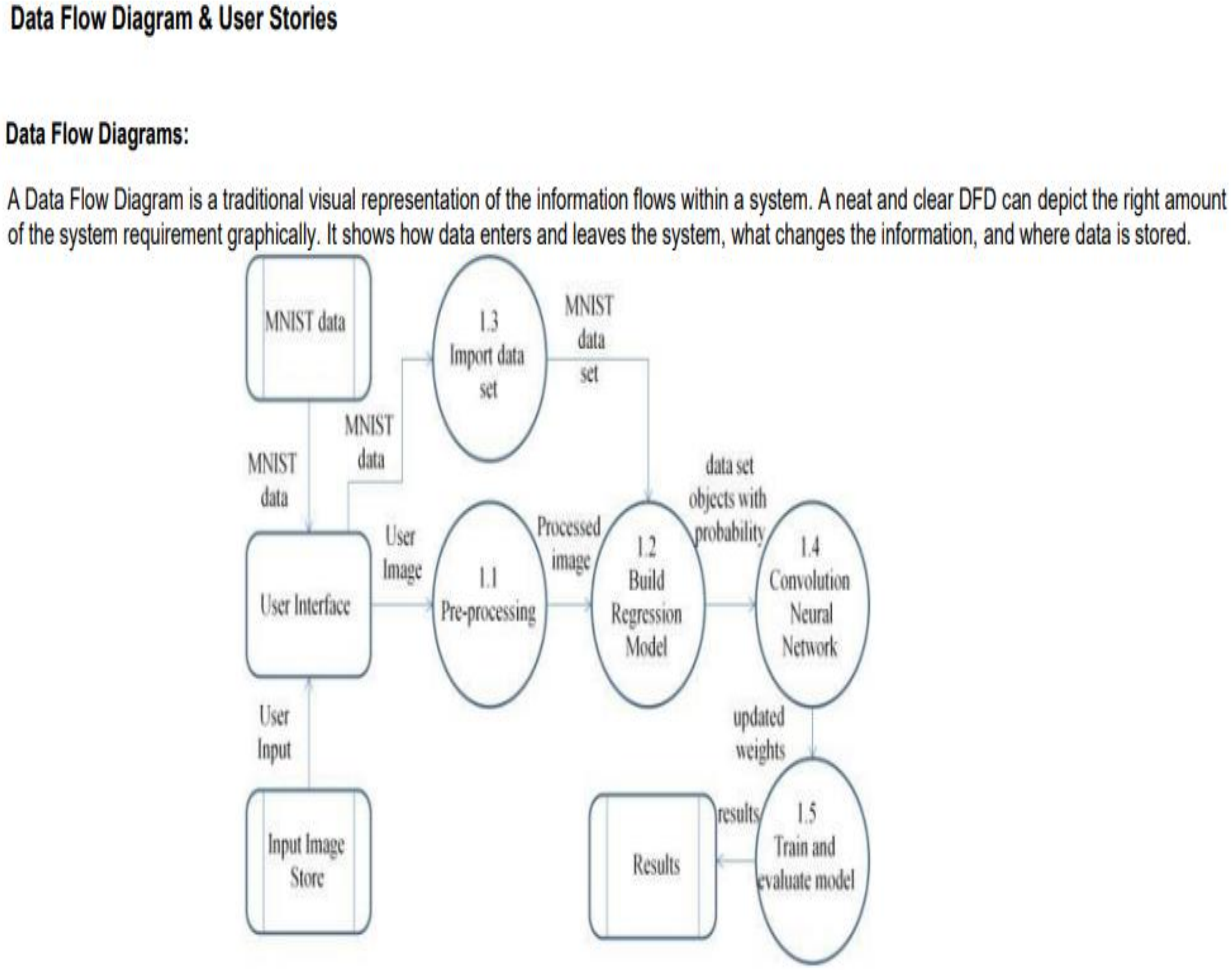
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application must be usable in all devices.
NFR-2	**Protection**	The application must protect user uploaded images.
NFR-3	**Reliability**	The application must give an accurate result as much as possible.
NFR-4	**Performance**	The application must be fast and quick to load up.
NFR-5	**Handiness**	The application must be available to use all the time.
NFR-6	**Scalability**	The application must be scale along with the user base.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



Data Flow Diagram & User Stories

Data Flow Diagrams:

A Data Flow Diagram is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Input correlation	Image Correlation is a technique used to recognize characters from images.
FR-2	Data Preparation	Collecting data and prepare it for training

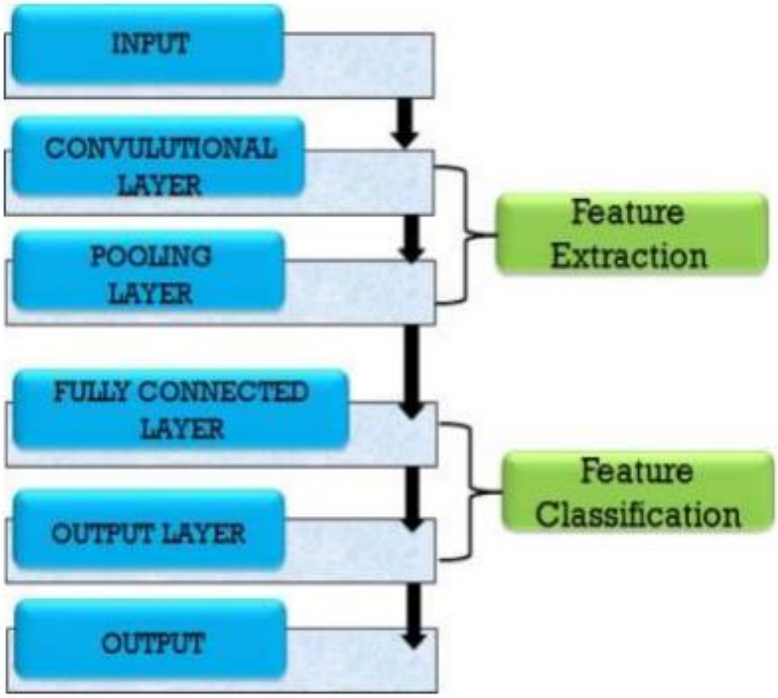
FR-3	Feature extraction	Feature extraction is analyzing the images and derive some characteristics from these images that identify each specific element
FR-4	Character classification	During the classification phase, the attributes of the data in the picture are compared to the classes in the database to determine which class the picture belongs to.

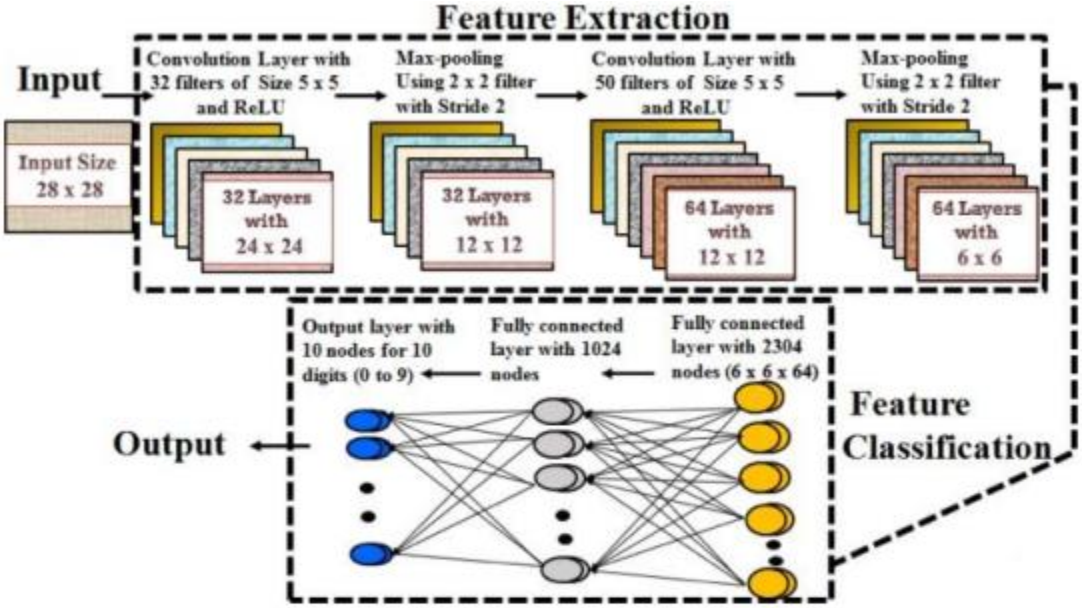
NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The software is very easy to use and reduces the learning work. To recognize the digits from bankcheque, papers, numeric entry in forms etc.
NFR-2	**Security**	The handwritten digit recognition can be used by banking sector where it can be used to maintain the security pin numbers, it can be also used for blind People by using sound output.
NFR-3	**Reliability**	This software will work reliably for low resolution images and not for graphical images.
NFR-4	**Performance**	Handwritten characters in the input image will be recognized with an accuracy of about 90% and more.
NFR-5	**Availability**	This system will retrieve the handwritten text regions only if the image contains written text in it.
NFR-6	**Scalability**	It contains thousands of handwritten digits that have been used in the development of programs .

Technology Architecture for Handwritten Digit Recognition System




Feature Extraction
Input
Input Size 28 x 28
Convolution Layer with 32 filters of Size 5 x 5 and ReLU
32 Layers with 24 x 24
Max-pooling Using 2 x 2 filter with Stride 2
32 Layers with 12 x 12
Convolution Layer with 50 filters of Size 5 x 5 and ReLU
64 Layers with 12 x 12
Max-pooling Using 2 x 2 filter with Stride 2
64 Layers with 6 x 6
Output layer with 10 nodes for 10 digits (0 to 9)
Fully connected layer with 1024 nodes
Fully connected layer with 2304 nodes (6 x 6 x 64)
Feature Classification
Output

5.3 USER STORIES

User type	Functional Requirements	User story number	User story/Task	Acceptance criteria	Priority	Release
Customer	Accessing the application	USN 1	As a user, I should be able to access the application from anywhere and use on any devices.	User can access the application using the browser on any device.	High	Sprint-4
	Uploading image	USN-2	As a user, I should be able to upload images to predict the digits	User can upload image	High	Sprint-3
	Viewing the result	USN-3	As a user, I should be able to view the result	The result of the prediction is displayed	Medium	Sprint-3

	Viewing other prediction	USN-4	As a user, I should be able to see other close prediction	The accuracy of other values must be displayed	Medium	Sprint-4
	Usage instruction	USN-5	As a user, I should have a usage instruction to know how to use the application	The usage instruction is displayed on the home page	Medium	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the applicationby entering my email, password, and confirming my password.	2	High	Brindha.M
Sprint-1	Login	USN-2	As a user, I can log into the application byentering email & password	1	High	Lavanya
Sprint-2	Upload Image of digital document	USN-3	As a user, I can able to input the images of digital documents to the application	2	Medium	Deepika.G.S
Sprint-2	Prediction	USN-4	As a user, I can predict the word	1	Medium	Anudharshini.R
Sprint-3	Upload Image of Handwritten document	USN-5	As a user, I can able to input the images ofthe handwritten documents or images to the application	2	High	Mathumitha.C
Sprint-3	Recognize text	USN-6	As a user, I can able to choose the font ofthe text to be displayed	1	Medium	Deepika.G.S
Sprint-4	Recognize digit	USN-7	As a user I can able to get the recognized digit as output from the images of digital documents or images	1	Medium	Brindha.M
Sprint-4	Recognize digit	USN-8	As a user I can able to get the recognized digit as output from the images ofhandwritten	2	High	Lavanya

			documents or images			

PROJECT TRACKER, VELOCITY & BURNDOWN CHART:

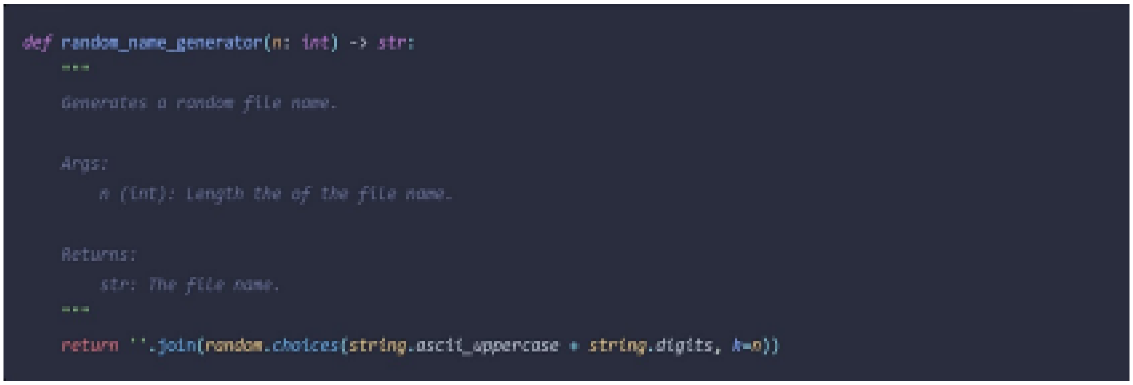
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	2	6 Days	24 Oct 2022	29 Oct 2022	2	29 Oct 2022
Sprint-2	2	6 Days	31 Oct 2022	05 Nov 2022	2	05 Nov 2022
Sprint-3	2	6 Days	07 Nov 2022	12 Nov 2022	2	12 Nov 2022
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	19 Nov 2022

6.2 SPRINT DELIVERY SCHEDULE

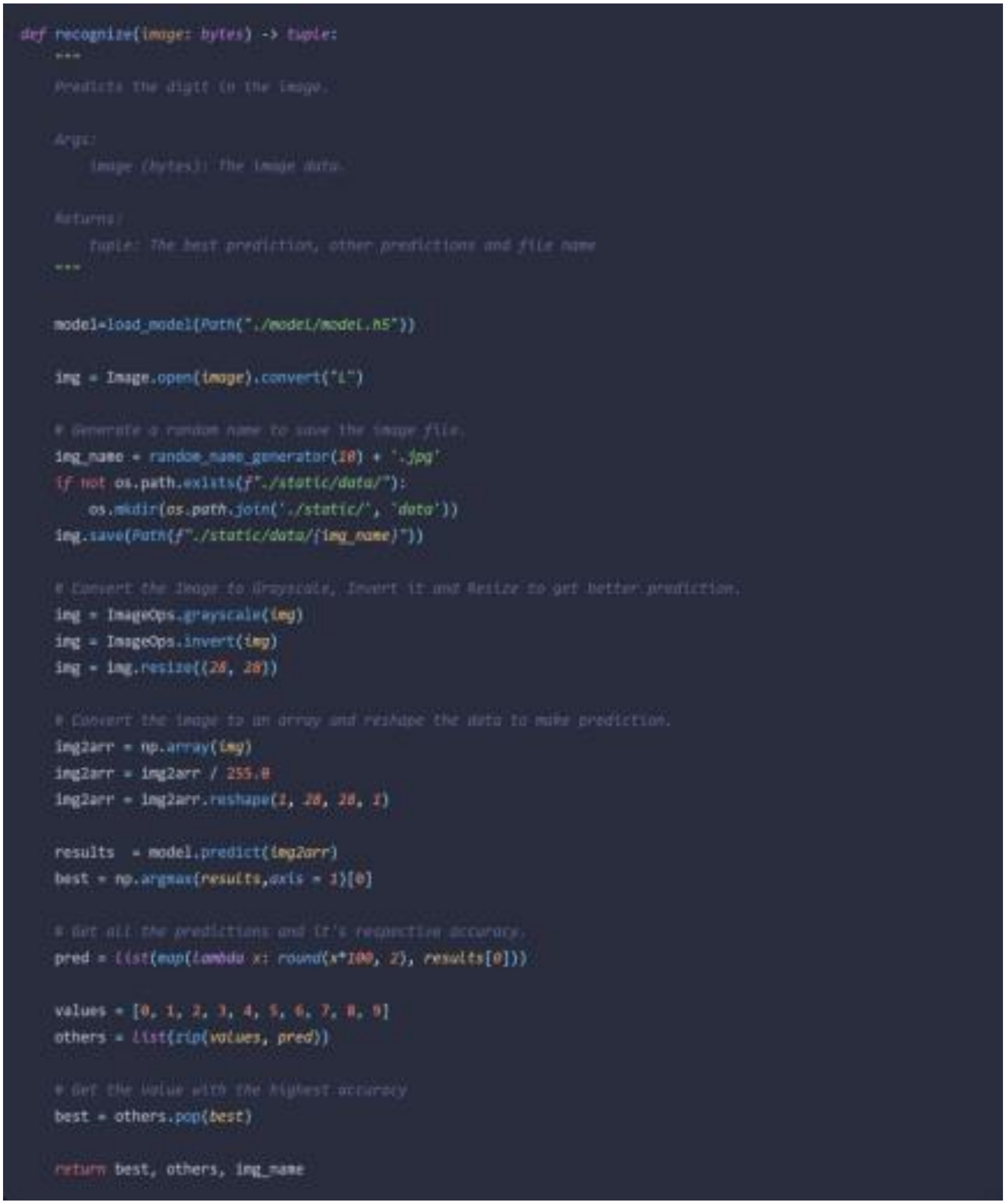
Sprint	Total story points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story points Completed(As on planed Date)	Sprint Released Date(Actual)
Sprint-1	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	9	6 Days	31 Oct 2022	05 Nov 2022	9	05 Nov 2022
Sprint-3	10	6 Days	07 Oct 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	9	6 Days	14 Nov 2022	19 Nov 2022	9	19 Nov 2022

CHAPTER 7

CODING & SOLUTION



```python
def random_name_generator(n: int) -> str:
    """
    Generates a random file name.

    Args:
        n (int): Length the of the file name.

    Returns:
        str: The file name.
    """
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))
```



```python
def recognize(image: bytes) -> tuple:
    """
    Predicts the digit in the image.

    Args:
        image (bytes): The image data.

    Returns:
        tuple: The best prediction, other predictions and file name
    """

    model=load_model(Path("./model/model.h5"))

    img = Image.open(image).convert("L")

    # Generate a random name to save the image file.
    img_name = random_name_generator(10) + '.jpg'
    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    # Convert the Image to Grayscale, Invert it and Resize to get better prediction.
    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))

    # Convert the image to an array and reshape the data to make prediction.
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]

    # Get all the predictions and it's respective accuracy.
    pred = list(map(lambda x: round(x*100, 2), results[0]))

    values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))

    # Get the value with the highest accuracy
    best = others.pop(best)

    return best, others, img_name
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_0 01	U I	Home Page	Verify UI Elements in the home page	The home page must be displayed properly	Working as expected	PASS
HP_TC_0 02	U I	Home Page	Check if the UI elements are displayed properly in different screen sizes	The home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x1801 and 768x630	FAIL
HP_TC_0 03	Functional	Home Page	Check if the user can upload their file	The input images should be uploaded to application successfully	Working as expected	PASS
HP_TC_0 04	Functional	Home Page	Check if the user can upload their un supported file	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_0 05	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the result page	Working as expected	PASS
BE_TC_0 1	Functional	Back end	Check if all the routs are working properly	All the routs should work properly	Working as expected	PASS
M_TC_0 1	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the	Working as expected	PASS

				result		
	M_TC_002 Functional	Model	Check if the model predict the digit	The model should predict the number	Working as expected	PASS
	M_TC_003 Functional	Model	Check if the model can handle complex input images	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_0 01	UI	Result page	Verify UI elements in the result page	The result page must be displayed properly	Working as expected	PASS
RP_TC_0 02	UI	Result page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_0 03	UI	Result page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2

Duplicate	0	0	0	0	0
External	0	0	3	0	3
Fixed	3	2	0	2	7
Not Reproduced	0	1	0	0	1
Skipped	0	0	0	0	0
Won’t Fix	1	0	1	1	3
Total	5	3	5	3	16

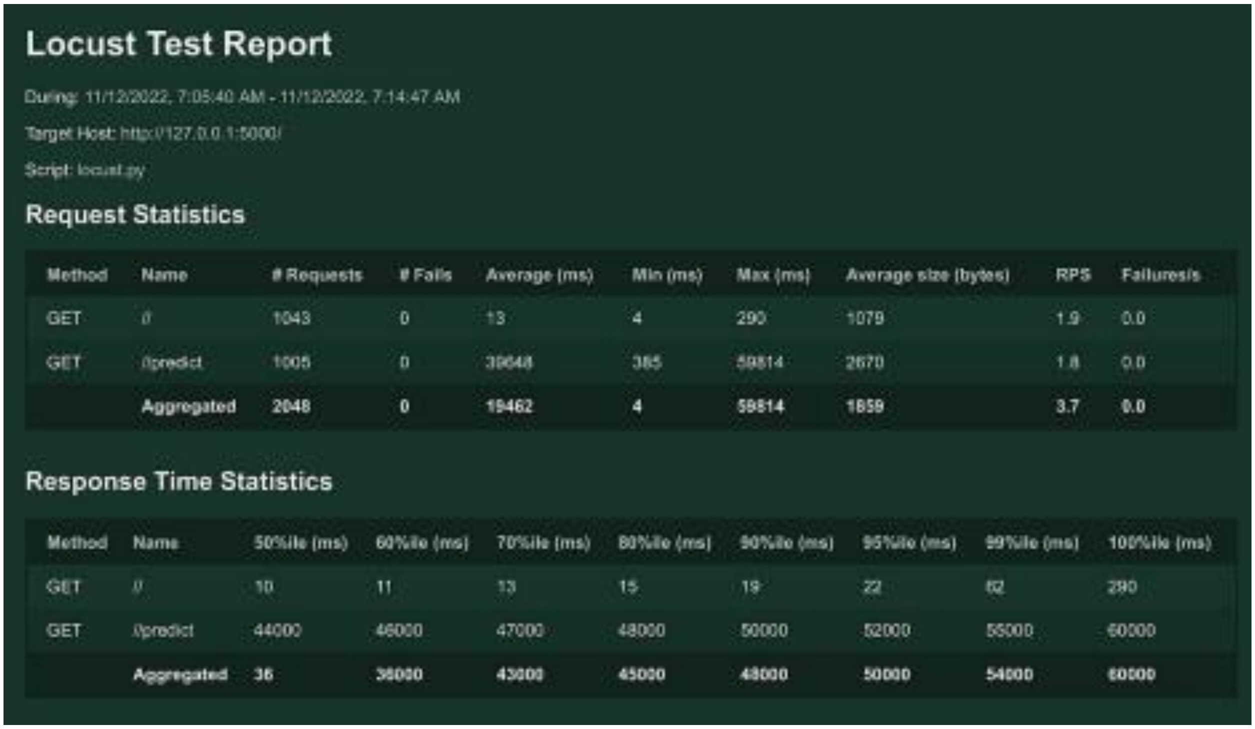
8.2.2 TEST CASE ANALYSIS

Section Total	**Not cases**	**Tested**	**Fail**	**Pass**
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Report	2	0	0	2

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS



Locust Test Report

During: 11/12/2022, 7:05:40 AM - 11/12/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

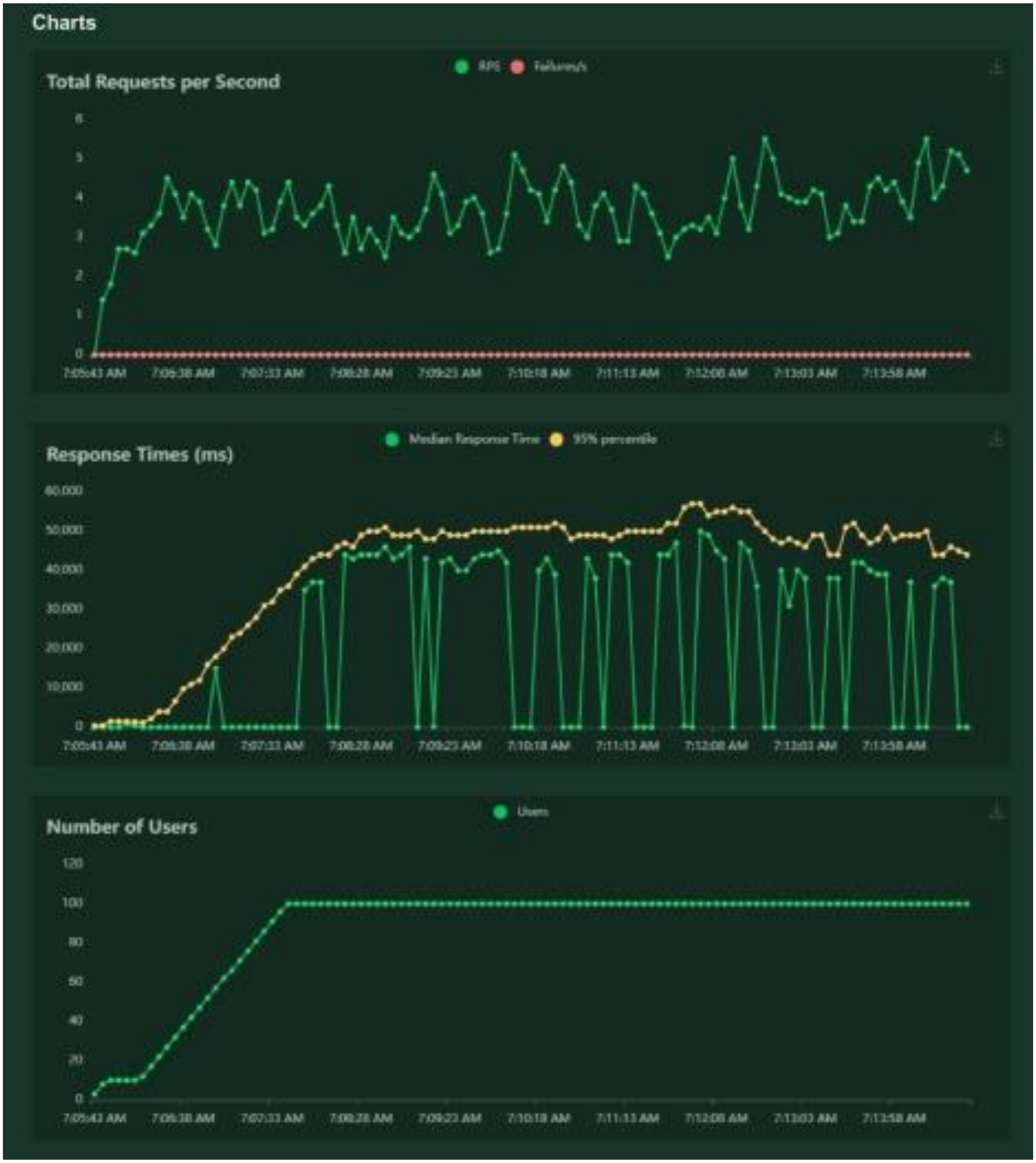
Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	1043	0	13	4	290	1079	1.9	0.0
GET	//predict	1005	0	39648	385	59814	2670	1.8	0.0
	Aggregated	**2048**	**0**	**19462**	**4**	**59814**	**1859**	**3.7**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	10	11	13	15	19	22	62	290
GET	//predict	44000	46000	47000	48000	50000	52000	55000	60000
	Aggregated	**36**	**38000**	**43000**	**45000**	**48000**	**50000**	**54000**	**60000**


Charts
Total Requests per Second
RPS
Failures/s
6
5
4
3
2
1
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Response Times (ms)
Median Response Time
95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Number of Users
Users
120
100
80
60
40
20
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Decreases manual work
- More accurate prediction
- Ability to handle bulk of data
- Portable and Scalable

DISADVANTAGES

- Difficulty in handling complex data
- Data must be in digital format
- Need of high performance server for faster predictions

CHAPTER 11

CONCLUSION

This project demonstrated a web application that uses machine learning to recognize handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is useful in real-world scenarios such as recognizing numberplates of vehicles, processing bank quench amounts, numeric entries in forms filledup by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12

FUTURE SCOPE

This project is far from complete and there is a substantial amount of room foimprovement.

Some of the improvements that can be made to this project are as follows:

● Add support to detect from digits multiple images and save the results ● Add support to detect multiple digits

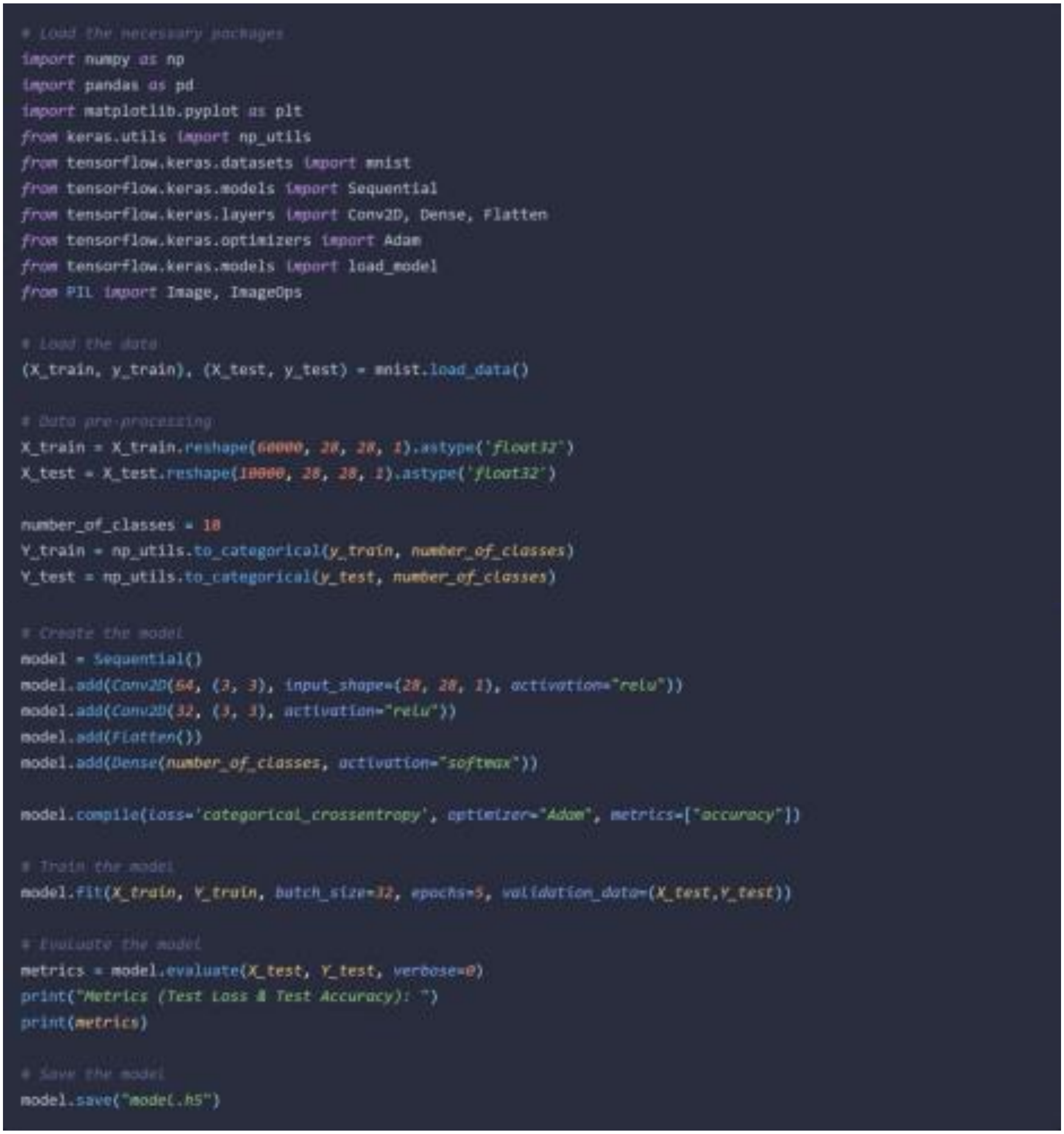
● Improve model to detect digits from complex images

● Add support to different languages to help users from all over the world

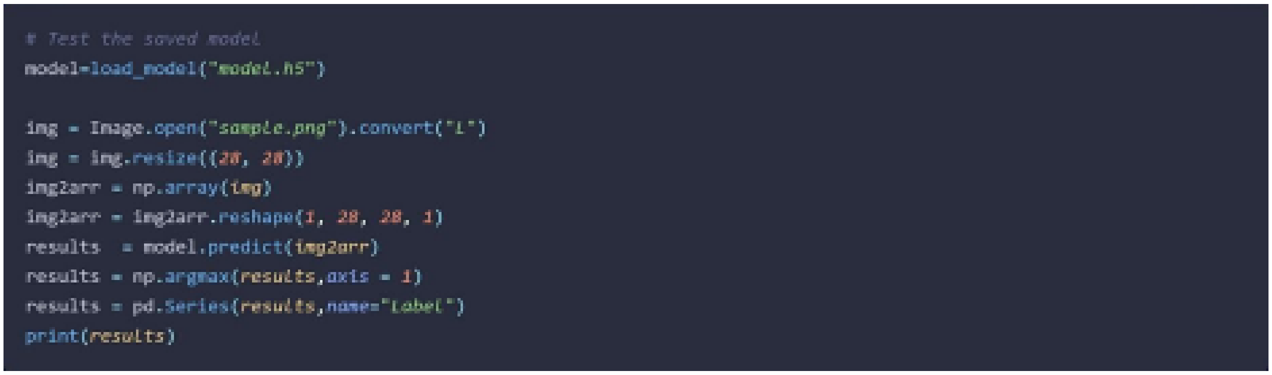
This project has endless potential and can always be enhanced to become better.Implementing this concept in the real world will benefit several industries and reduce the workload on many workers,enhancing overall work efficiency.

APPENDIX

SOURCE CODE

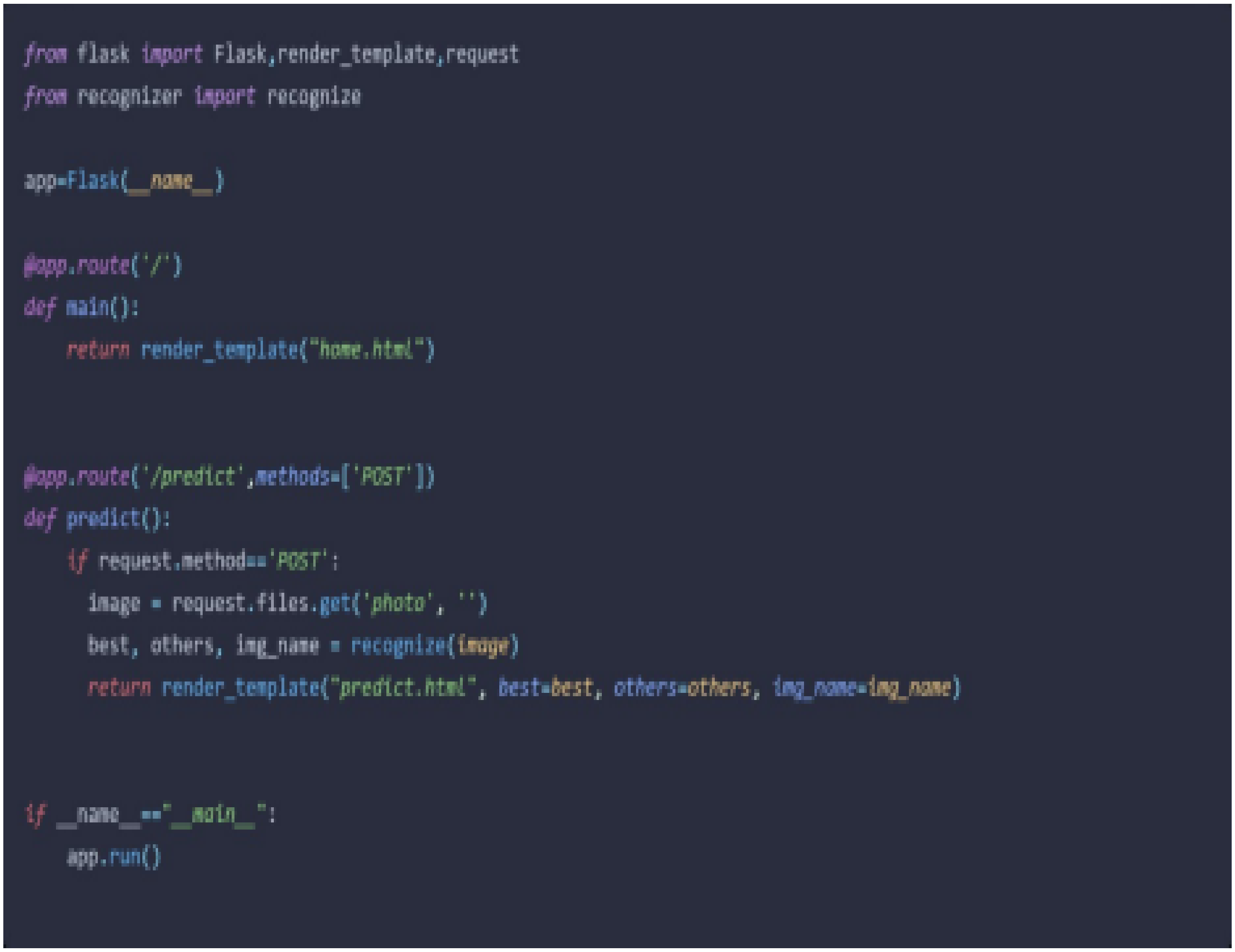


```python
# Load the necessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps

# Load the data
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Data pre-processing
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')

number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)

# Create the model
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))

model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])

# Train the model
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))

# Evaluate the model
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)

# Save the model
model.save("model.h5")
```



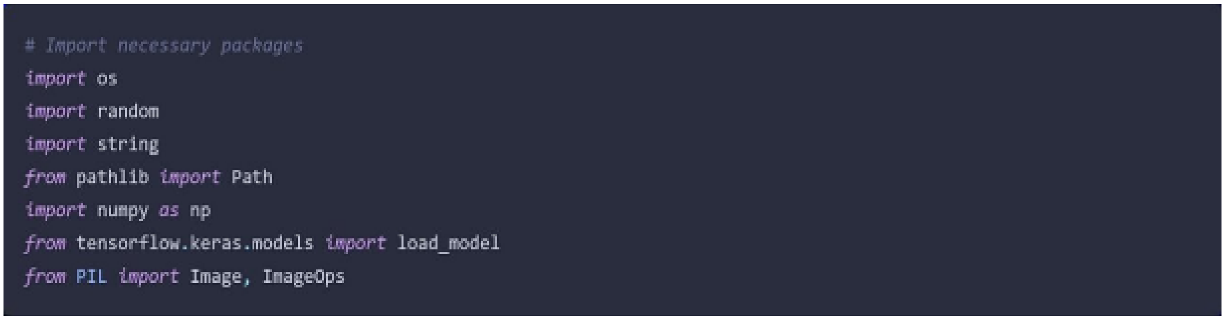
```
# Test the saved model
model=load_model("model.h5")

img = Image.open("sample.png").convert("L")
img = img.resize((28, 28))
img2arr = np.array(img)
img2arr = img2arr.reshape(1, 28, 28, 1)
results  = model.predict(img2arr)
results = np.argmax(results,axis = 1)
results = pd.Series(results,name="Label")
print(results)
```

FLASK APP

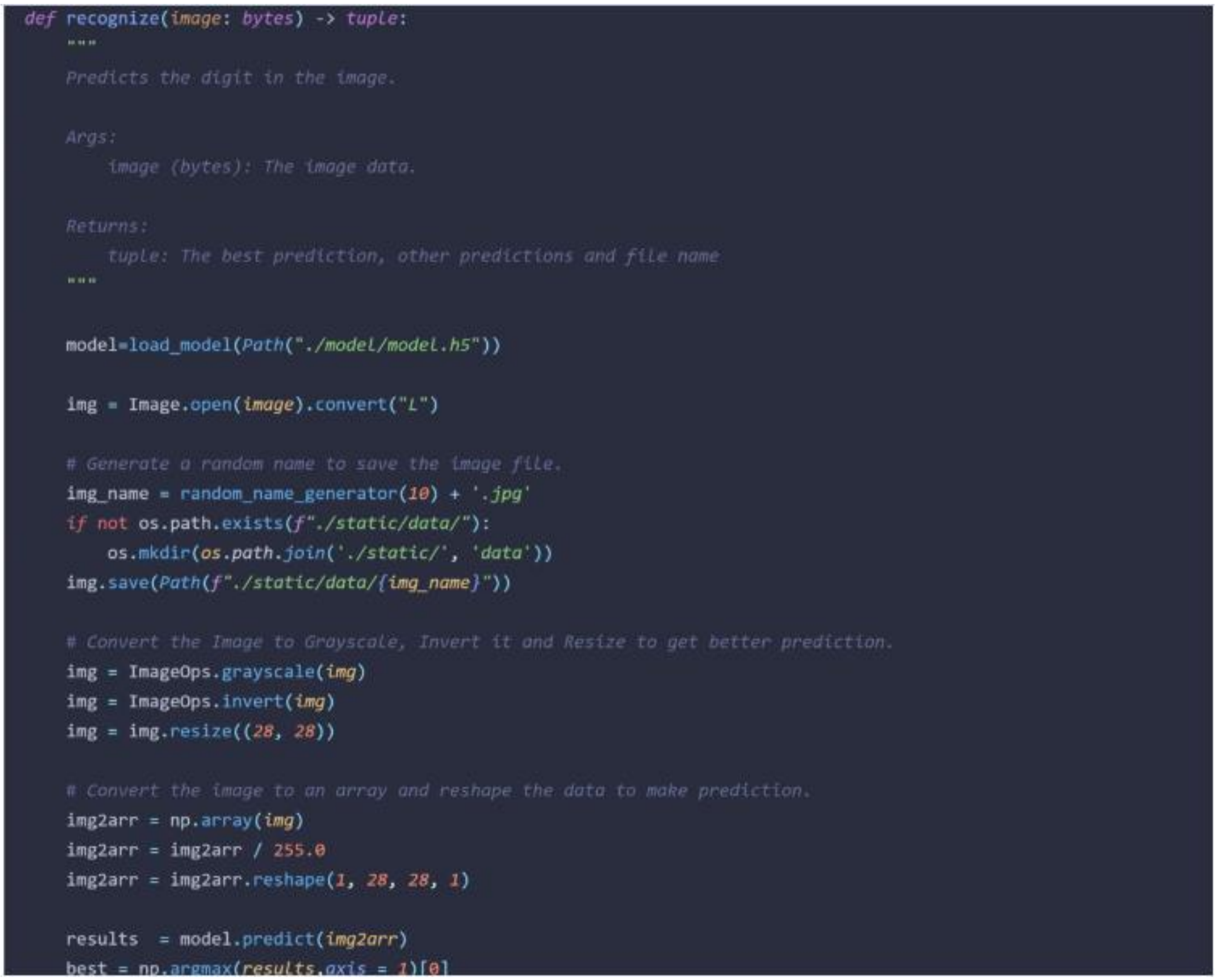


```
from flask import Flask,render_template,request
from recognizer import recognize

app=Flask(__name__)

@app.route('/')
def main():
    return render_template("home.html")



@app.route('/predict',methods=['POST'])
def predict():
    if request.method=='POST':
      image = request.files.get('photo', '')
      best, others, img_name = recognize(image)
      return render_template("predict.html", best=best, others=others, img_name=img_name)



if __name__=="__main__":
    app.run()
```

RECOGNIZOR



```python
# Import necessary packages
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
```



```python
def recognize(image: bytes) -> tuple:
    """
    Predicts the digit in the image.

    Args:
        image (bytes): The image data.

    Returns:
        tuple: The best prediction, other predictions and file name
    """

    model=load_model(Path("./model/model.h5"))

    img = Image.open(image).convert("L")

    # Generate a random name to save the image file.
    img_name = random_name_generator(10) + '.jpg'
    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    # Convert the Image to Grayscale, Invert it and Resize to get better prediction.
    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))

    # Convert the image to an array and reshape the data to make prediction.
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]
```

HTML HOMEPAGE

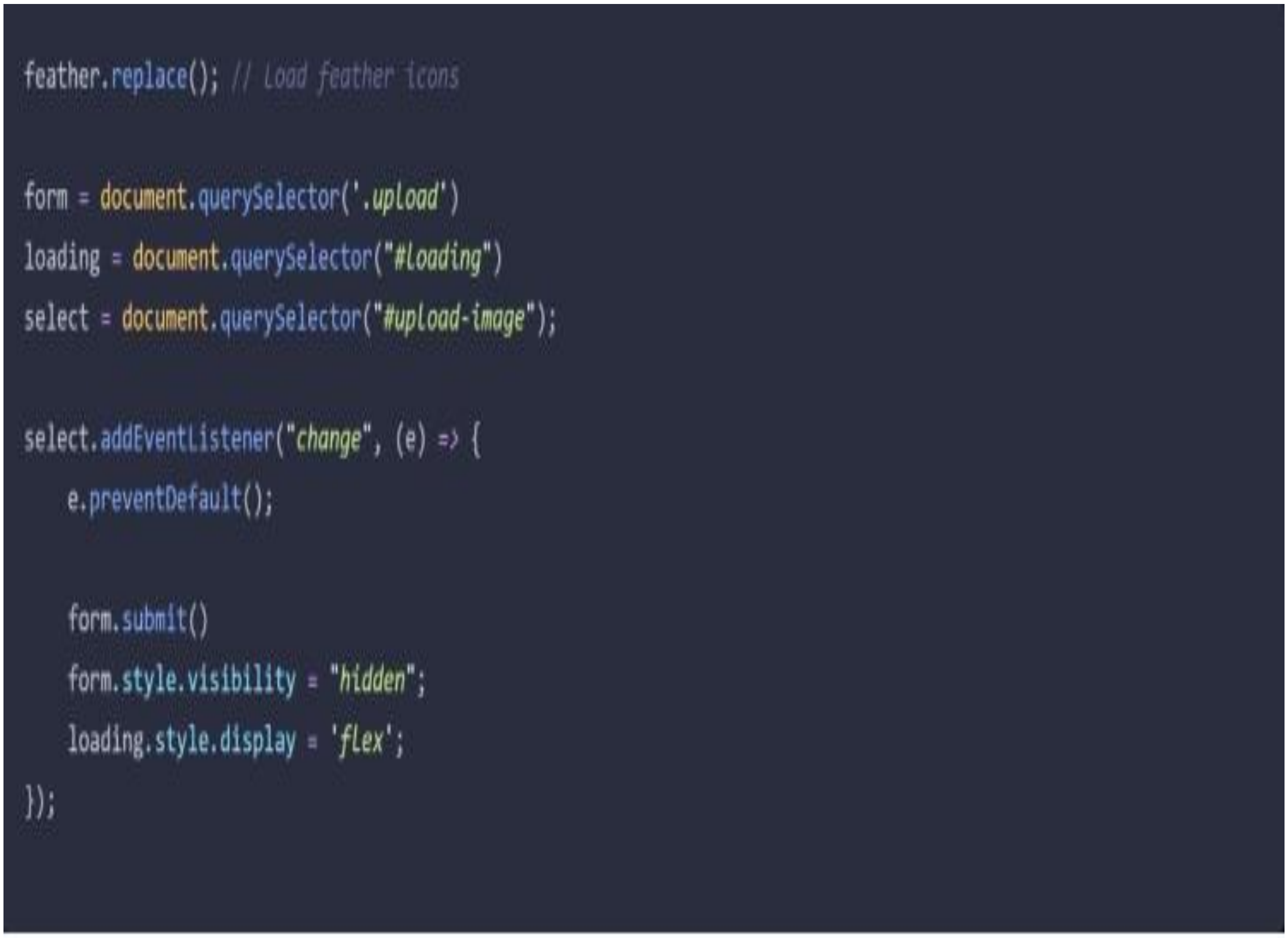
```
<html>
<head>
  <title>Handwritten digit recognition</title>
  <meta name="viewport" content="width=device-width">
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display
=swap" rel="stylesheet">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "style.css">
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>
</head>
<style>
    body{
     background-image: url('backimg.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }
</style>
```

```
<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}
    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
        });
</script>
<body>
    <h1>HandWritten Digit Recognition System</h1>
        <div class="container p-3 my-3 bg-black text-white">
            <p>Handwritten Digit Recognition is a technology that is much needed in this
world as of Today.This Digit Recognition System is used to recognize the digits from
different sources like email, posts, cheque etc. Before proper implementation of this
technology we have relied on writing text with our own hands which can result in error.It's
difficult to store and access physical data with efficiency.The project presents in
representing the recognization of handwritten digits (0 - 9) from the famous MNIST dataset.
Here we will be using AlexNet which is an architecture of Convolutional Neural Network.</p>
        </div>
        <section id="content">
            <div class="leftside">
            <form action="/predict" method="POST" enctype="multipart/form-data">
            <label>Select a image:</label>
            <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
              <img id="frame" width="100px" height="100px"/>
              <div class="buttons_div">
                <button type="submit" class="btn btn-light"
onclick="myfunc()">Predict</button>
                <script>
                    function myfunc()
                    {
                        window.location.href="C:\IBM project\predict.html";
                    }
                </script>
                <button type="button" class="btn btn-light">&nbsp Clear &nbsp</button>
              </div>
            </form>
            </div>
      </section>
</body>
</html>
```

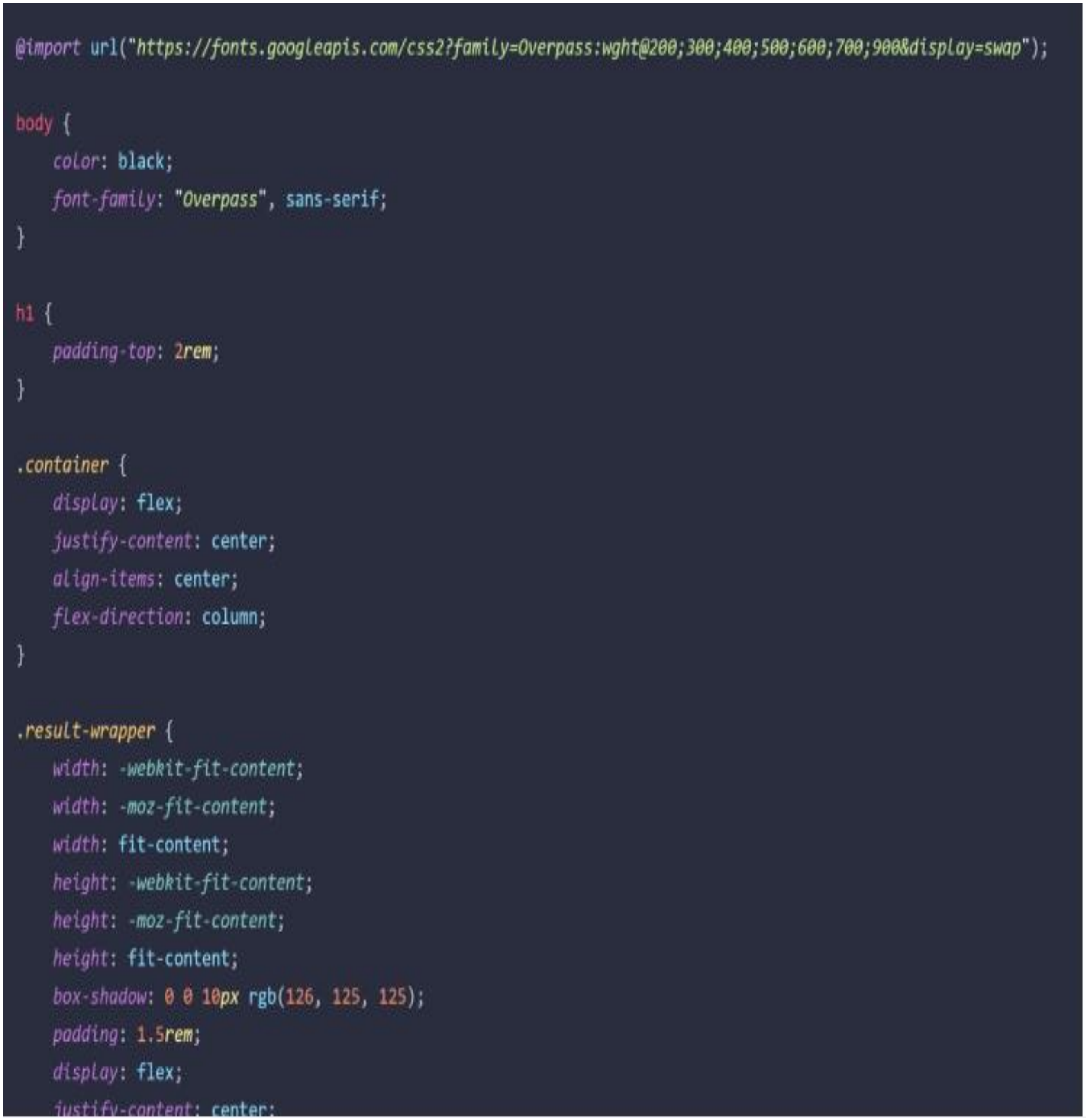
PREDICT.HTML

```
<!DOCTYPE
html>
            <html lang="en">
            <head>
                <meta charset="UTF-8">
                <title>Prediction</title>
            </head>
            <style>
                body{
                 background-image: url('backimg.jpg');
                 background-repeat: no-repeat;
                 background-size: cover;
                }
                #rectangle{
                 width:400px;
                 height:150px;
                 background-color: #000000;
                 border-radius: 15px;
                 position:absolute;
                 box-shadow: 0px 0px 10px 5px grey;
                 top:25%;
                 left:50%;
                 transform:translate(-50%,-50%);
                }
                #head{
              text-align: center;
              font-size: 30px;
              margin: 0 auto;
              padding: 3% 5%;
              font-family: Arial, Helvetica, sans-serif;
              color: white;
                }
                #num{
                    font-size: 50px;
                }
            </style>
            <body>

                <div id="rectangle">
                    <h1 id="head">Predicted Number : <br><center
            id="num">{{num}}</center></h1>
                </div>
            </body>
            </html>
```

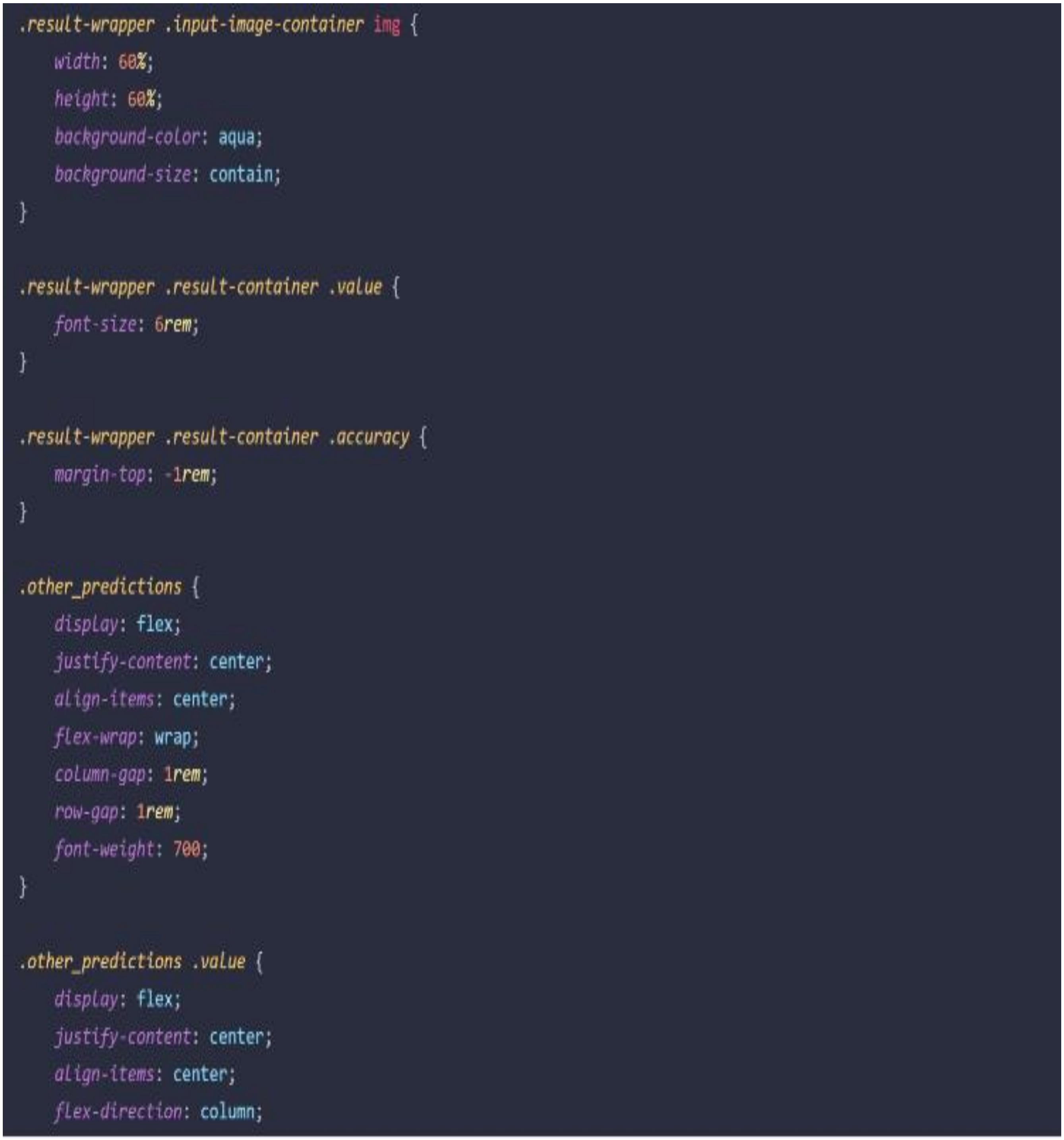
HOME PAGE (JS)



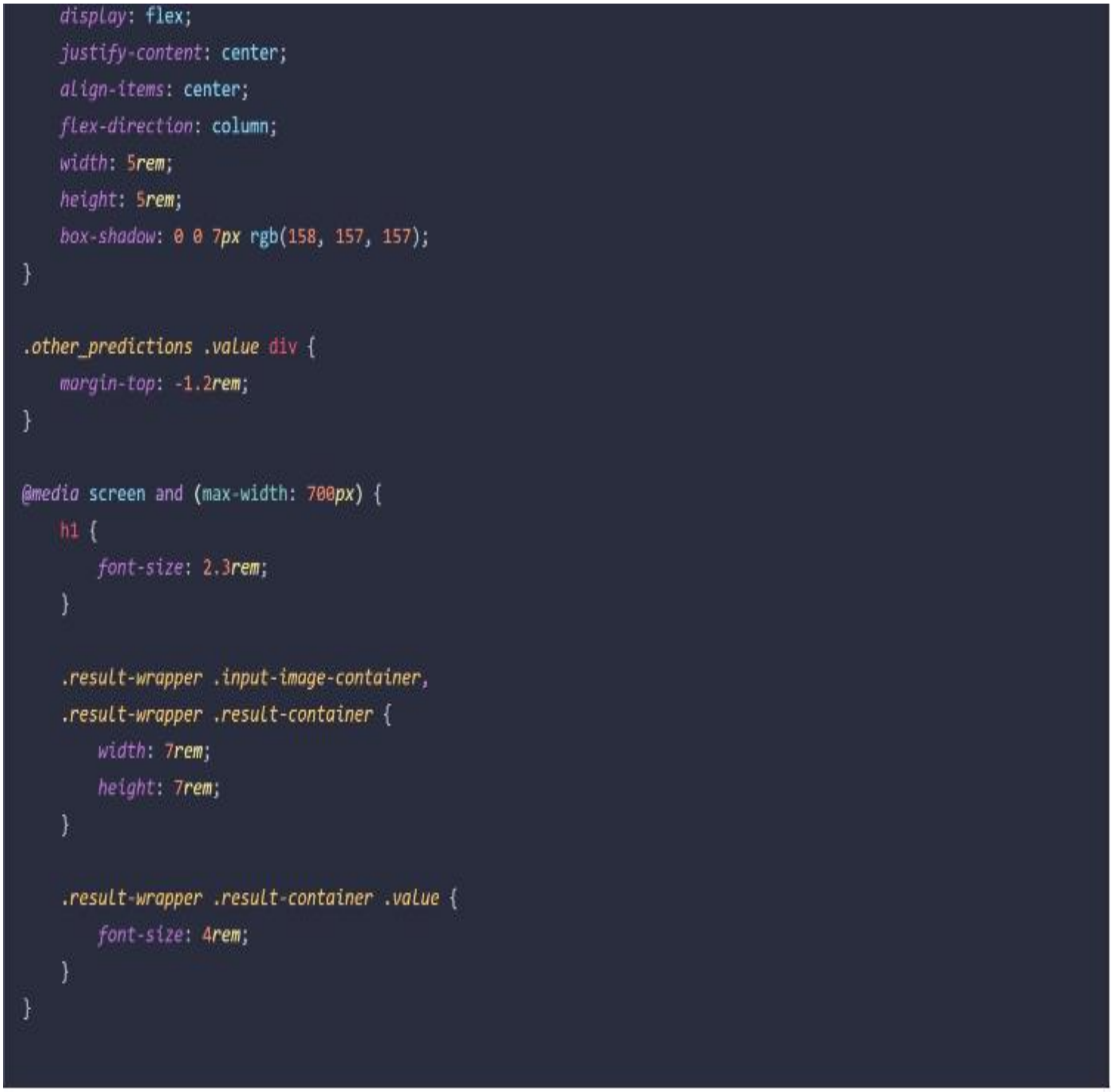
```
feather.replace(); // Load feather icons


form = document.querySelector('.upload')
loading = document.querySelector("#loading")
select = document.querySelector("#upload-image");


select.addEventListener("change", (e) => {
    e.preventDefault();


    form.submit()
    form.style.visibility = "hidden";
    loading.style.display = 'flex';
});
```



```css
@import url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700;900&display=swap");

body {
    color: black;
    font-family: "Overpass", sans-serif;
}

h1 {
    padding-top: 2rem;
}

.container {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}

.result-wrapper {
    width: -webkit-fit-content;
    width: -moz-fit-content;
    width: fit-content;
    height: -webkit-fit-content;
    height: -moz-fit-content;
    height: fit-content;
    box-shadow: 0 0 10px rgb(126, 125, 125);
    padding: 1.5rem;
    display: flex;
    justify-content: center;
```



```
.result-wrapper .input-image-container img {
    width: 60%;
    height: 60%;
    background-color: aqua;
    background-size: contain;
}

.result-wrapper .result-container .value {
    font-size: 6rem;
}

.result-wrapper .result-container .accuracy {
    margin-top: -1rem;
}

.other_predictions {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
    column-gap: 1rem;
    row-gap: 1rem;
    font-weight: 700;
}

.other_predictions .value {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
```



```
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    width: 5rem;
    height: 5rem;
    box-shadow: 0 0 7px rgb(158, 157, 157);
}

.other_predictions .value div {
    margin-top: -1.2rem;
}

@media screen and (max-width: 700px) {
    h1 {
        font-size: 2.3rem;
    }

    .result-wrapper .input-image-container,
    .result-wrapper .result-container {
        width: 7rem;
        height: 7rem;
    }

    .result-wrapper .result-container .value {
        font-size: 4rem;
    }
}
```

APP.PY PAGE:

```
Importnumpyasnp
                  import os
                  from PIL import Image
                  from flask import Flask, request, render_template, url_for
                  from werkzeug.utils import secure_filename, redirect
                  #from gevent.pywsgi import WSGIServer
                  from keras.models import load_model
                  from keras.preprocessing import image
                  from flask import send_from_directory
                  UPLOAD_FOLDER = 'C:\IBM project\data'
                  app = Flask(__name__)
                  app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
                  model = load_model("./models/mnistCNN.h5")
                  @app.route('/')
                  def index():
                      return render_template('home.html')
                  @app.route('/predict', methods=['GET', 'POST'])
                  def upload():
                      if request.method == "POST":
                          f = request.files["image"]
                          filepath = secure_filename(f.filename)
                          f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
                          upload_img = os.path.join(UPLOAD_FOLDER, filepath)
                          img = Image.open(upload_img).convert("L")  # convert image to
                  monochrome
                          img = img.resize((28, 28))  # resizing of input image
                          im2arr = np.array(img)  # converting to image
                          im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our
                  requirement
                          pred = model.predict(im2arr)
                          num = np.argmax(pred, axis=1)  # printing our Labels
                          return render_template('predict.html', num=str(num[0]))
                  if __name__ == '__main__':
```

```
    app.run(debug=True, threaded=False)import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
#from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
UPLOAD_FOLDER = 'C:\IBM project\data'
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
model = load_model("./models/mnistCNN.h5")
@app.route('/')
def index():
    return render_template('home.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to
monochrome
        img = img.resize((28, 28))  # resizing of input image
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our
requirement
        pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1)  # printing our Labels
        return render_template('predict.html', num=str(num[0]))
if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

STYLE.CSS

```
#clear_button{
                       margin-left: 20px;
                       font-weight: bold;
                       color: rgb(14, 137, 195);
                     }
                     #confidence{
                       font-family: cursive;
                       margin-top: 7.5%;
                     }
                     #content{
                       margin: 0 auto;
                       padding: 2% 15%;
                       padding-bottom: 0;
                     }
                     .welcome{
                        text-align: center;
                        position: relative;
                        color: rgb(0, 32, 112);
                        background-color: rgb(106, 173, 199);
                        padding-top: 1%;
                        padding-bottom: 1%;
                        font-weight: bold;
                        font-family: cursive;
                     }
                     #predict_button{
                       margin-right: 20px;
                       color: rgb(53, 75, 150);
                       font-weight: bold;
                     }
                     #prediction_heading{
                       font-family: cursive;
                       margin-top: 7.5%;
                     }
                     #result{
                       font-size: 5rem;
                     }
                     #title{
                       padding: 1.5% 15%;
                       margin: 0 auto;
                       text-align: center;
                     }
                     .btn {
```

```
    font-size: 15px;
    padding: 10px;
    /* -webkit-appearance: none; */
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}
.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}
.heading{
  font-family:cursive;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}
.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;

}
#frame{
  margin-right: 10%;
}
.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;

}
h1{
  text-align: center;
  color: aliceblue;
  padding: 100px 50px 65px 100px;
}
@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}
```

FINAL CODE:

Import required libraries*

```
from keras.datasets import mnist
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,Dense,Flatten
from tensorflow.keras.optimizers import Adam
```

Loading the dataset

```
 (X_train,y_train),(X_test,y_test) =mnist.load_data()
print(X_train.shape)
print(X_test.shape)
print(y_test.shape)
print(y_train.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-data
sets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
(60000, 28, 28)
(10000, 28, 28)
(10000,)
(60000,)
```

Fetch the data from the dataset

```
print("The label value is ",y_test[9]) #Value in y_test
plt.imshow(X_test[9])
```

```
The label value is  9
```

```
print("The label value is ",y_test[10]) #Value in y_test
plt.imshow(X_test[10])
```

```
The label value is  0
```

```
print("The label value is ",y_test[23]) #Value in y_test
plt.imshow(X_test[23])
```

```
The label value is  5
```

Applying one hot encoding

```
X_train.shape
```

```
 (60000, 28, 28)
```

```
X_test.shape
```

```
 (10000, 28, 28)
```

```
X_train1 = X_train.reshape(60000, 28, 28, 1).astype('float32')

X_test1 = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

```
number_of_classes= 12
y_train1 = np_utils.to_categorical(y_train,number_of_classes)
y_test1 = np_utils.to_categorical(y_test,number_of_classes)
```

Encoding the value

```
print("After encoding the value",y_test[9] ,"become", y_test1[9])
```

```
After encoding the value 9 become [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
```

```
print("After encoding the value",y_test[10] ,"become", y_test1[10])
```

```
After encoding the value 0 become [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
print("After encoding the value",y_test[23] ,"become", y_test1[23])
```

```
After encoding the value 5 become [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
```

Add CNN layers

```
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
```

Compile the model

```
model.compile(loss='categorical_crossentropy', optimizer="Adam",
metrics=["accuracy"])
```

Train the model

```
model.fit(X_train1, y_train1, batch_size=32, epochs=10,
validation_data=(X_test1,y_test1))
```

```
Epoch 1/10
1875/1875 [==============================] - 194s 103ms/step - loss: 0.2549 -
accuracy: 0.9504 - val_loss: 0.0902 - val_accuracy: 0.9730
Epoch 2/10
1875/1875 [==============================] - 194s 104ms/step - loss: 0.0726 -
accuracy: 0.9782 - val_loss: 0.0820 - val_accuracy: 0.9742
Epoch 3/10
1875/1875 [==============================] - 194s 103ms/step - loss: 0.0496 -
accuracy: 0.9848 - val_loss: 0.0817 - val_accuracy: 0.9759
Epoch 4/10
1875/1875 [==============================] - 195s 104ms/step - loss: 0.0383 -
accuracy: 0.9880 - val_loss: 0.0899 - val_accuracy: 0.9785
Epoch 5/10
1875/1875 [==============================] - 195s 104ms/step - loss: 0.0313 -
accuracy: 0.9907 - val_loss: 0.1074 - val_accuracy: 0.9761
Epoch 6/10
1875/1875 [==============================] - 194s 104ms/step - loss: 0.0244 -
accuracy: 0.9928 - val_loss: 0.1156 - val_accuracy: 0.9773
Epoch 7/10
1875/1875 [==============================] - 193s 103ms/step - loss: 0.0218 -
accuracy: 0.9936 - val_loss: 0.1221 - val_accuracy: 0.9771
Epoch 8/10
1875/1875 [==============================] - 192s 102ms/step - loss: 0.0196 -
accuracy: 0.9946 - val_loss: 0.1727 - val_accuracy: 0.9778
Epoch 9/10
```

```
1875/1875 [==============================] - 192s 103ms/step - loss: 0.0171 -
accuracy: 0.9953 - val_loss: 0.1468 - val_accuracy: 0.9785
Epoch 10/10
1875/1875 [==============================] - 193s 103ms/step - loss: 0.0144 -
accuracy: 0.9962 - val_loss: 0.1704 - val_accuracy: 0.9777
```

Observing the metrics

```
metrics = model.evaluate(X_test1, y_test1, verbose=0)
print("Checking the Metrics (Test Loss & Test Accuracy): ")
print(metrics)
```

```
Checking the Metrics (Test Loss & Test Accuracy):
[11.306961059570312, 0.12229999899864197]
```

Test the model

```
prediction = model.predict(X_test1[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 112ms/step
[[5.0968147e-06 3.2904151e-08 2.4547335e-08 3.8771137e-09 9.9999297e-01
  2.1400561e-12 9.0379384e-09 1.9089430e-06 2.7502803e-10 2.1564152e-10
  1.3407317e-11 2.5973085e-08]
 [1.0000000e+00 2.0193573e-12 1.2437545e-10 3.0768805e-12 1.9168457e-14
  6.3709477e-10 1.7837687e-10 2.4965596e-14 3.3803925e-13 1.5835364e-13
  1.1105061e-16 9.6047545e-12]
 [2.0305255e-05 1.8551295e-04 2.5913024e-03 1.0359057e-05 1.3580263e-04
  2.6764979e-05 1.2820570e-02 6.5554171e-03 8.3878607e-02 8.9356083e-01
  3.7450151e-05 1.7707223e-04]
 [7.9626665e-07 2.8583373e-09 1.5453403e-09 3.7636035e-04 8.5368520e-06
  9.6965458e-08 9.9961424e-01 1.3939068e-12 9.4559267e-09 9.6343879e-14
  7.8885800e-18 1.2240818e-09]]
```

```
import numpy as np
print(np.argmax(prediction, axis=1))
print(y_test1[:4])
```

```
[4 0 9 6]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

Save the model

```
model.save("digit.h5")
```

```
from tensorflow.keras.models import load_model
model=load_model("digit.h5")
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640       
                                                                 
 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464     
                                                                 
 flatten (Flatten)           (None, 18432)             0         
                                                                 
 dense (Dense)               (None, 12)                221196    
```

```
=================================================================
Total params: 240,300
Trainable params: 240,300
Non-trainable params: 0
_________________________________________________________________
```

```
# Saving in tar

!tar -zcvf digit_recognition.tgz digit.h5
```

```
digit.h5
```

```
!pip install watson-machine-learning-client
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB
)
     |████████████████████████████████| 538 kB 4.4 MB/s
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-pack
ages (from watson-machine-learning-client) (0.8.10)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages
(from watson-machine-learning-client) (4.64.1)
Collecting lomond
  Downloading lomond-0.3.3-py2.py3-none-any.whl (35 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-pack
ages (from watson-machine-learning-client) (2.23.0)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packa
ges (from watson-machine-learning-client) (1.24.3)
Collecting ibm-cos-sdk
  Downloading ibm-cos-sdk-2.12.0.tar.gz (55 kB)
     |████████████████████████████████| 55 kB 3.5 MB/s
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packag
es (from watson-machine-learning-client) (1.3.5)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packa
ges (from watson-machine-learning-client) (2022.9.24)
Collecting boto3
  Downloading boto3-1.26.9-py3-none-any.whl (132 kB)
     |████████████████████████████████| 132 kB 53.2 MB/s
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 6.4 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting botocore<1.30.0,>=1.29.9
  Downloading botocore-1.29.9-py3-none-any.whl (9.9 MB)
     |████████████████████████████████| 9.9 MB 45.8 MB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/
python3.7/dist-packages (from botocore<1.30.0,>=1.29.9->boto3->watson-machine
-learning-client) (2.8.2)
Collecting urllib3
  Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)
     |████████████████████████████████| 140 kB 45.8 MB/s
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-pack
ages (from python-dateutil<3.0.0,>=2.1->botocore<1.30.0,>=1.29.9->boto3->wats
on-machine-learning-client) (1.15.0)
Collecting ibm-cos-sdk-core==2.12.0
  Downloading ibm-cos-sdk-core-2.12.0.tar.gz (956 kB)
     |████████████████████████████████| 956 kB 55.8 MB/s
```

```
Collecting ibm-cos-sdk-s3transfer==2.12.0
  Downloading ibm-cos-sdk-s3transfer-2.12.0.tar.gz (135 kB)
     |████████████████████████████████| 135 kB 53.0 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting requests
  Downloading requests-2.28.1-py3-none-any.whl (62 kB)
     |████████████████████████████████| 62 kB 1.2 MB/s
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/pyt
hon3.7/dist-packages (from requests->watson-machine-learning-client) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->watson-machine-learning-client) (2.10)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist
-packages (from pandas->watson-machine-learning-client) (1.21.6)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas->watson-machine-learning-client) (2022.6)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-co
s-sdk-s3transfer
  Building wheel for ibm-cos-sdk (setup.py) ... done
  Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.12.0-py3-none-any.whl
size=73931 sha256=828cd7ebe3989eb3f0f89d8aa8b2672fdfedbacff67110754e1186bc114
462b3
  Stored in directory: /root/.cache/pip/wheels/ec/94/29/2b57327cf00664b661430
4f7958abd29d77ea0e5bbece2ea57
  Building wheel for ibm-cos-sdk-core (setup.py) ... done
  Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.12.0-py3-no
ne-any.whl size=562962 sha256=e3e83fbd43e20a5e9f729519f4f078ad1ddd5e749e91026
173e51feee7d799e8
  Stored in directory: /root/.cache/pip/wheels/64/56/fb/5cd6f4f40406c828a5289
b95b2752a4d142a9afb359244ed8d
  Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done
  Created wheel for ibm-cos-sdk-s3transfer: filename=ibm_cos_sdk_s3transfer-2
.12.0-py3-none-any.whl size=89778 sha256=45c0dc69fa9821741f923f81f07af8872d68
c74bf9cbc2ee0dd7a6237a07a3d2
  Stored in directory: /root/.cache/pip/wheels/57/79/6a/ffe3370ed7ebc00604f9f
76766e1e0348dcdcad2b2e32df9e1
Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer
Installing collected packages: urllib3, requests, jmespath, ibm-cos-sdk-core,
botocore, s3transfer, ibm-cos-sdk-s3transfer, lomond, ibm-cos-sdk, boto3, wat
son-machine-learning-client
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.24.3
    Uninstalling urllib3-1.24.3:
      Successfully uninstalled urllib3-1.24.3
  Attempting uninstall: requests
    Found existing installation: requests 2.23.0
    Uninstalling requests-2.23.0:
      Successfully uninstalled requests-2.23.0
Successfully installed boto3-1.26.9 botocore-1.29.9 ibm-cos-sdk-2.12.0 ibm-co
s-sdk-core-2.12.0 ibm-cos-sdk-s3transfer-2.12.0 jmespath-0.10.0 lomond-0.3.3
requests-2.28.1 s3transfer-0.6.0 urllib3-1.26.12 watson-machine-learning-clie
nt-1.0.391
```

In [ ]:

```
!pip install ibm_watson_machine_learning
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
```

```
Collecting ibm_watson_machine_learning
  Downloading ibm_watson_machine_learning-1.0.257-py3-none-any.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 4.3 MB/s
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python
3.7/dist-packages (from ibm_watson_machine_learning) (1.3.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-pac
kages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-pack
ages (from ibm_watson_machine_learning) (2.28.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packa
ges (from ibm_watson_machine_learning) (2022.9.24)
Collecting ibm-cos-sdk==2.7.*
  Downloading ibm-cos-sdk-2.7.0.tar.gz (51 kB)
     |████████████████████████████████| 51 kB 630 kB/s
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packa
ges (from ibm_watson_machine_learning) (1.26.12)
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-pack
ages (from ibm_watson_machine_learning) (0.8.10)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7
/dist-packages (from ibm_watson_machine_learning) (4.13.0)
Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packag
es (from ibm_watson_machine_learning) (0.3.3)
Collecting ibm-cos-sdk-core==2.7.0
  Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)
     |████████████████████████████████| 824 kB 46.8 MB/s
Collecting ibm-cos-sdk-s3transfer==2.7.0
  Downloading ibm-cos-sdk-s3transfer-2.7.0.tar.gz (133 kB)
     |████████████████████████████████| 133 kB 39.6 MB/s
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/pytho
n3.7/dist-packages (from ibm-cos-sdk==2.7.*->ibm_watson_machine_learning) (0.
10.0)
Collecting docutils<0.16,>=0.10
  Downloading docutils-0.15.2-py3-none-any.whl (547 kB)
     |████████████████████████████████| 547 kB 54.5 MB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/
python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.*->ib
m_watson_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2022.6)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist
-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.21.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-pack
ages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.7.0->ibm-cos-sdk=
=2.7.*->ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/pyt
hon3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->ibm_watson_machine_learning) (2.10)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-pac
kages (from importlib-metadata->ibm_watson_machine_learning) (3.10.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/pyt
hon3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (
4.1.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/pyt
hon3.7/dist-packages (from packaging->ibm_watson_machine_learning) (3.0.9)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-co
s-sdk-s3transfer
```

```
  Building wheel for ibm-cos-sdk (setup.py) ... done
  Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.7.0-py2.py3-none-any.
whl size=72563 sha256=659267c434e8e7c27acc7dda571c4454f1a639f6511dd150da1952a
79c21e6cf
  Stored in directory: /root/.cache/pip/wheels/47/22/bf/e1154ff0f5de93cc477ac
d0ca69abfbb8b799c5b28a66b44c2
  Building wheel for ibm-cos-sdk-core (setup.py) ... done
  Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.7.0-py2.py3
-none-any.whl size=501013 sha256=4df31bb57b8cc5edbe1054ca45f259583c0bedd53a63
f1bdffa5b6207432b6e9
  Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17c
fd02c13369281f3d232aaf5902c19
  Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done
  Created wheel for ibm-cos-sdk-s3transfer: filename=ibm_cos_sdk_s3transfer-2
.7.0-py2.py3-none-any.whl size=88622 sha256=b0c77e9f333bbc5f59f67f5d8f8755168
4769077c751076e77c542812d38847e
  Stored in directory: /root/.cache/pip/wheels/5f/b7/14/fbe02bc1ef1af890650c7
e51743d1c83890852e598d164b9da
Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer
Installing collected packages: docutils, ibm-cos-sdk-core, ibm-cos-sdk-s3tran
sfer, ibm-cos-sdk, ibm-watson-machine-learning
  Attempting uninstall: docutils
    Found existing installation: docutils 0.17.1
    Uninstalling docutils-0.17.1:
      Successfully uninstalled docutils-0.17.1
  Attempting uninstall: ibm-cos-sdk-core
    Found existing installation: ibm-cos-sdk-core 2.12.0
    Uninstalling ibm-cos-sdk-core-2.12.0:
      Successfully uninstalled ibm-cos-sdk-core-2.12.0
  Attempting uninstall: ibm-cos-sdk-s3transfer
    Found existing installation: ibm-cos-sdk-s3transfer 2.12.0
    Uninstalling ibm-cos-sdk-s3transfer-2.12.0:
      Successfully uninstalled ibm-cos-sdk-s3transfer-2.12.0
  Attempting uninstall: ibm-cos-sdk
    Found existing installation: ibm-cos-sdk 2.12.0
    Uninstalling ibm-cos-sdk-2.12.0:
      Successfully uninstalled ibm-cos-sdk-2.12.0
Successfully installed docutils-0.15.2 ibm-cos-sdk-2.7.0 ibm-cos-sdk-core-2.7
.0 ibm-cos-sdk-s3transfer-2.7.0 ibm-watson-machine-learning-1.0.257
```

Cloud deployment

```
from ibm_watson_machine_learning import APIClient

wml_credentials = {
                   "url": "https://us-south.ml.cloud.ibm.com",  # example:
"https://eu-gb.ml.cloud.ibm.com"
                   "apikey":"Dt-EkyRgxXR--1mhO8JnCjRGR_AvzoUpJQqbzFnWklU1"
                  }

client = APIClient(wml_credentials)
client
```

```
Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future
release. Use Python 3.9 framework instead.
```

```
client.spaces.get_details()
```

```
{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:u
s-south:a/d74a81b5072a47ea932088f3c95b3d8d:ab0faf12-e097-475c-b555-79f9a13b44
0d::',
      'guid': 'ab0faf12-e097-475c-b555-79f9a13b440d',
      'name': 'Watson Machine Learning-lz',
      'type': 'machine_learning'}],
    'description': '',
    'name': 'digit_deploy',
    'scope': {'bss_account_id': 'd74a81b5072a47ea932088f3c95b3d8d'},
    'stage': {'production': False},
    'status': {'state': 'active'},
    'storage': {'properties': {'bucket_name': 'dede02b9-9740-4319-881c-f10ec6
202dce',
      'bucket_region': 'us-south',
      'credentials': {'admin': {'access_key_id': '9bfe67bd39f14cf5a8666e6188b
02143',
        'api_key': '50PMGAm3eSnX_G1VpNG6_XJkwa-veWNCSyyru5ksZsWB',
        'secret_access_key': 'b63dd4e1b1ecefdbdb32478174a66d411cd7a98519c8565
b',
        'service_id': 'ServiceId-cf7956f9-5d6e-4fde-9bf9-c2d7d324d3d3'},
       'editor': {'access_key_id': '9e76c7cc5b2c438396b834aaeda87df4',
        'api_key': 'EzZkGCey-46EuCVz3IztC8mnBFtuaD40Srufvm_hFBUz',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:globa
l:a/d74a81b5072a47ea932088f3c95b3d8d:b81cecb9-1689-4f8e-87d7-c70c72300b4e::',
        'secret_access_key': '00cbee74cb48d75ca43d688108297703eea7ec26903a04c
d',
        'service_id': 'ServiceId-725da56e-c4c0-4ecb-9d36-ea58872bbcf3'},
       'viewer': {'access_key_id': '238ea99d20354b55b78c557fdb973972',
        'api_key': 'im-71co9LWBLEb295LCJlWx4AOejZgzJAxpq1SB9P5N9',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:globa
l:a/d74a81b5072a47ea932088f3c95b3d8d:b81cecb9-1689-4f8e-87d7-c70c72300b4e::',
        'secret_access_key': 'e3ca34240ce3757c166469ac364c6df4e20f464cbbad5d7
a',
        'service_id': 'ServiceId-ca7069ff-0f0b-479e-af7f-4127e8cd1703'}},
      'endpoint_url': 'https://s3.us-south.cloud-object-storage.appdomain.clo
ud',
      'guid': 'b81cecb9-1689-4f8e-87d7-c70c72300b4e',
      'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d7
4a81b5072a47ea932088f3c95b3d8d:b81cecb9-1689-4f8e-87d7-c70c72300b4e::'},
     'type': 'bmcos_object_storage'}},
   'metadata': {'created_at': '2022-11-13T07:31:19.376Z',
    'creator_id': 'IBMid-666002J5U4',
    'id': '0d542d58-0e93-4b26-a2c6-156ce46c2f36',
    'updated_at': '2022-11-13T07:31:32.819Z',
    'url': '/v2/spaces/0d542d58-0e93-4b26-a2c6-156ce46c2f36'}}]}
```

```
def guid_space_name(client,digit_deploy):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if
item['entity']['name']==digit_deploy)['metadata']['id'])
```

```
space_uid = guid_space_name(client,'digit_deploy')
space_uid
```

```
'0d542d58-0e93-4b26-a2c6-156ce46c2f36'
client.set.default_space(space_uid)
```

```
'SUCCESS'
client.software_specifications.list()
```

```
-----------------------------  ------------------------------------  ----
NAME                           ASSET_ID                              TYPE
default_py3.6                  0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12      020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt     069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6        09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12     09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9      0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6          0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                     0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod   1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6              10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl      111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10        125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9             12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6        154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                   1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6         1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6           1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt  1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6           1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod   217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda        26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                       295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8            2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6          2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9          2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6              2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt     32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37           36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10        396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6             39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt     40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10     40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9         4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                 42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                   493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6          49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6             4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6         50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8            52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11     55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                 5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1              5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                     5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7            632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8         634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
-----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' param
eter.
software_space_uid =
client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
```

```
software_space_uid
```

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
model_details =
client.repository.store_model(model='digit_recognition.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"DigitRecognition Model",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

```
model_details
```

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_2.7'},
 'metadata': {'created_at': '2022-11-15T06:32:10.093Z',
  'id': '892f9dba-862a-4094-8701-f063b6fd66da',
  'modified_at': '2022-11-15T06:32:14.285Z',
  'name': 'DigitRecognition Model',
  'owner': 'IBMid-666002J5U4',
  'resource_key': '0961989d-65f0-4052-9429-70ed03c421fb',
  'space_id': '0d542d58-0e93-4b26-a2c6-156ce46c2f36'},
 'system': {'warnings': []}}
model_id = client.repository.get_model_id(model_details)
model_id
```

```
'892f9dba-862a-4094-8701-f063b6fd66da'
client.repository.download(model_id,'Digit_Recognition_Model.tar.gb')
```

```
Successfully saved model content to file: 'Digit_Recognition_Model.tar.gb'
'/content/Digit_Recognition_Model.tar.gb'
```

-----------------THE END-----------------

GITHUB

https://github.com/IBM-EPBL/IBM-Project-32124-1660208167

DEMO LINK

https://drive.google.com/file/d/1riaT9_A3yVC7456nf7b0Hs36mxHs7Op1/view?usp=drivesdk