PROJECT REPORT

A NOVEL METHOD FOR HANDWRITEN DIGIT RECOGNITION

submitted by

Team ID : PNT2022TMID07022

Kamalahasan A	**-**	**19181L07**
Shihaf Ahil M	**-**	**1918136**
Sugumar S	**-**	**1918138**
Varatharaj Manikandan S	**-**	**1918L18**

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

HANDWRITTEN digit recognition is the abilityof a computer system to recognize the handwritten inputs like digits,characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades.Some of the research areas include signatureverification, bank check processing, postal address interpretation from envelopes etc. Here comes the use of Deep Learning. In the past decade, deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machinelearning tools have been developedlike scikit-learn, scipy-image etc. and pybrains, Keras, Theano, Tensorflow by Google, TFLearn etc. for Deep Learning. Thesetools make the applications robust and therefore more accurate.

The Artificial Neural Networks can almost mimic the humanbrain and are a key ingredient in image processing field. For example, Convolutional Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for creating Art by learningfrom existing artist styles etc.. Handwriting Recognition has an active community of academics studying it.

Classification of images and patterns has been one of the major implementation of Machine Learning and Artificial Intelligence. People are continuously trying to make computersintelligent so that they can do almost all the work done by humans Handwriting recognition system is the most basic and an important step towards this huge and interesting area of Computer Vision.

1.2 PURPOSE

Digit recognition systems are able to identify numbers from a variety of sources, including emails, bank checks, papers, images, etc. They can also be used in a variety of real-world situations, such as online handwriting recognition on computer tablets orsystems, identifying vehiclelicence plates, processing bank cheque amounts, and reading numbers from formsthat have been filled out by hand (such as tax forms).

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

The fundamental problemwith handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbersbecause of similaritiesbetween numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally,the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES

Hermans et al. have addressedthe MNIST handwritten digit classification problem. In this context, 10 iterations are used for each image in the MNIST dataset; in other words, each input digit is repeated for 10 masking periods. In their experiments, the authors focusedon both an MNIST handwritten digit classification dataset, and a TIMIT phoneme classification dataset. In both MNIST and TIMT datasets, the authors found that optimizing the input encoding can make great improvements over random masks.

Mohapatra et al. proposed a new method for classifying MNIST handwritten digit images. In their new method, the authors used the discrete cosine space-frequency transform to extract image features and artificial neuralnetwork classifiers to solve the classification problem. In order to reduce the computational cost, the authors proposed to normalize all the images of the MNIST handwritten digitdataset and excludeundesirable boundary pixels.
Kussul and Baidyk proposed a new neural classifier limited receptive area (LIRA) for MNIST handwritten digit images classification. In the LIRA classifier, the sensor layer is followedwith the associative layer, and the trainable connections are used to connect the associative layer with the output layer.Experiments with MNISThandwritten digit imagesshow that the LIRA classifier has achieveda classification accuracyof 99.41%.

In order to classify MNIST handwritten digit images, Ahlawata and Choudharyb proposedto build a hybrid classification model by integrating convolutional neural networksand support vector machines (SVM). In this context, the authors used convolutional neural networks to extract the features of the image, while SVM was used as a binary classifier. Based on experimental results the authors have achieved a classification accuracy of 99.28%.Chazal et al. proposed to use identical network topologies to compare between two weight optimization methods using MNIST handwritten digit classification database. In the first weight optimization methods, the authors use the extreme learning machine algorithm. While backpropagation algorithm is used in the second weight optimization methods. Based on their experimental results, the authors found that the weight optimization method that uses the extreme learningmachine is much faster than the one that uses the backpropagation algorithm. Ma and Zhang adopted deep analysis with multi- feature extraction to build a handwritten digit classification method. In order to excludenegative information and maintain relevantfeatures, the images of various sizes were normalized, and projection features were extracted from pre-processed images. Distribution features and projection features are also used to classify MNIST handwritten digit datasets.

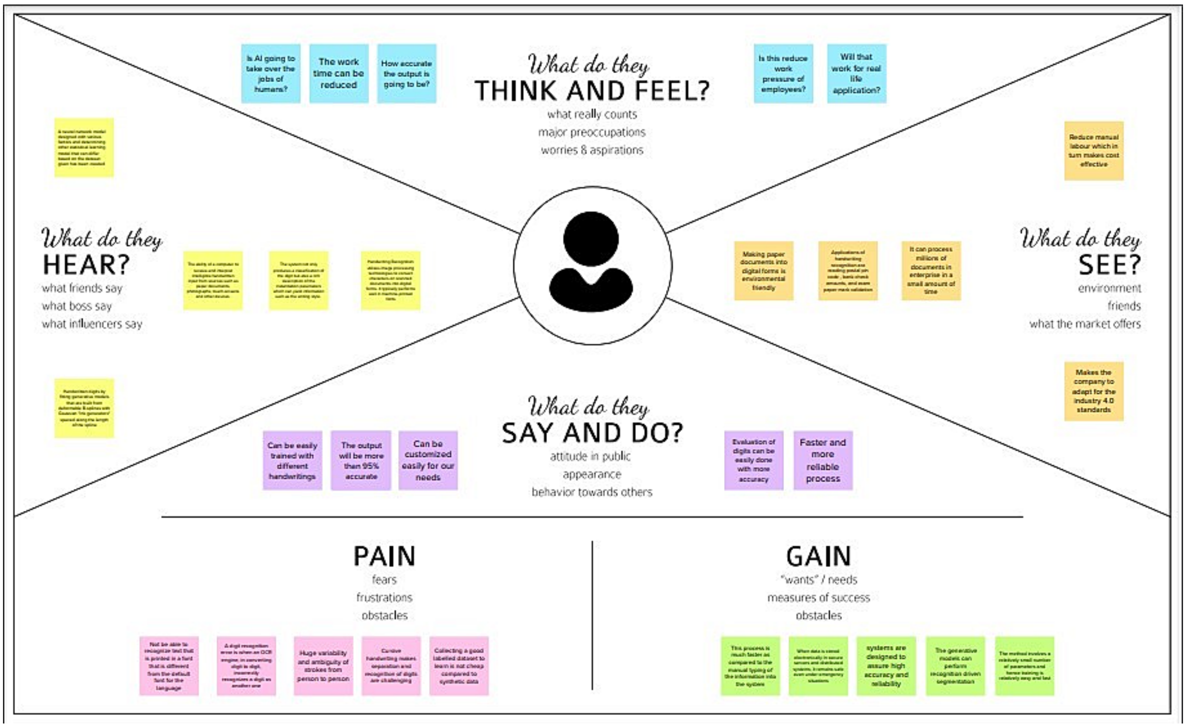
2.3 PROBLEM STATEMENT DEFINITION

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

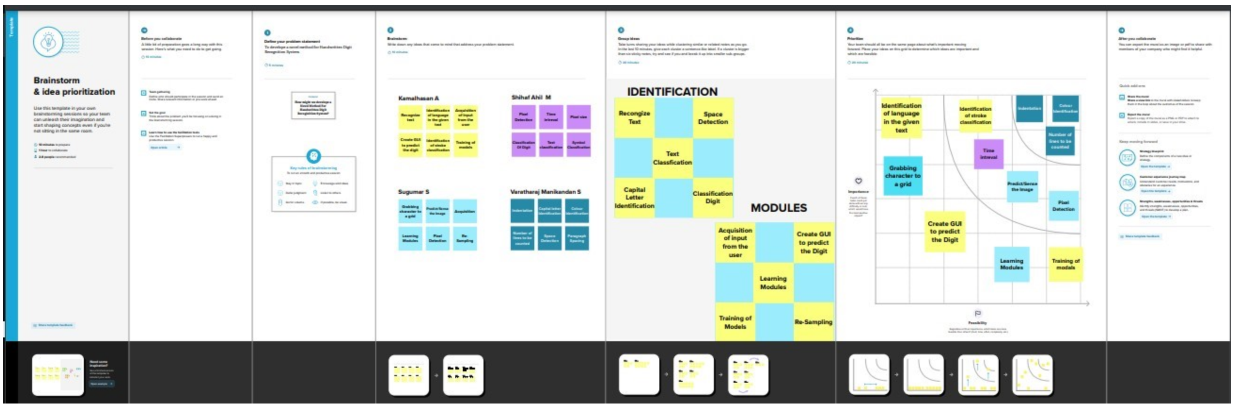
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Handwritten digit recognition is the ability of computer system to recognize the handwritten characters from wide variety of sources like emails, papers, images, etc. Manually written digits are of different sizes, styles, orientation,thickness and position. The model should be able to identify them and predict the output correctly.
2.	Idea / Solution description	To solve this problem, we are going to use Convolutional neural network. It is a type of Neural Networks and it is mainly used to identifying the image and speech recognition. It reduces the high dimensionality of image without losing of information. That’s why CNNs are especially suited for this use case.
3.	Novelty / Uniqueness	a. Training dataset contains more than40,000records. b. CNN is the model we have chosen. c. It recognises the digits with good accuracy.

4.	Social Impact / Customer Satisfaction	The main impact of this work is to reduce the errors that occurs in banking sectors. Due to the incorrect recognition of handwritten digits that are written in cheques and credit cards. So that this digit recognition will greatly improve the goodwill of the organization and customer satisfaction
5.	Business Model (RevenueModel)	Every one of us have different styles of writing and perception. Manually recognizing the handwritten digits are error prone due to various factors. So, if this digit recognition is done manually in business organizations, even if a single error occurs, it may cause severe damage to the organization. So here, we have proposed a solution to automate the digit recognition process. A deep learning model is trained with images of different styles, sizes, orientation and then the model is based to predict based on previous learning.
6.	Scalability of the Solution	We can extend this project into providing solutions to various other problems like solving handwritten mathematical equation by making some changes with the training data and final code. Organizations such as banks, revenue

		departments, accounting sectors are facing issues in recognizing written digits such as in cheques. This can be handled by our handwritten digit recognition project as they expand into different business domains withoutimpacting performance. Our proposed solution is thus scalable and can fit into different domains and solve different problems.

3.4 PROBLEM SOLUTION FIT

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No:	Functional Requirement and description
FR-1	**Image Data**: Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9). This has been a topic of boundless-research in the field of deep learning. In the realm ofdeep learning, this has been the subject of countless studies.
FR-2	**Website:** Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	**Digit_Classifier_Model:** To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training andvalidation data first.
FR-4	**MNIST dataset:** The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.
FR-5	databases, software, virtual storage, and networking, among others. In layman's terms, Cloud Computing is defined as a virtual platform that **allows you to store and access your data over the internet without any limitations**.

4.2 NON FUNCTIONAL REQUIREMENTS

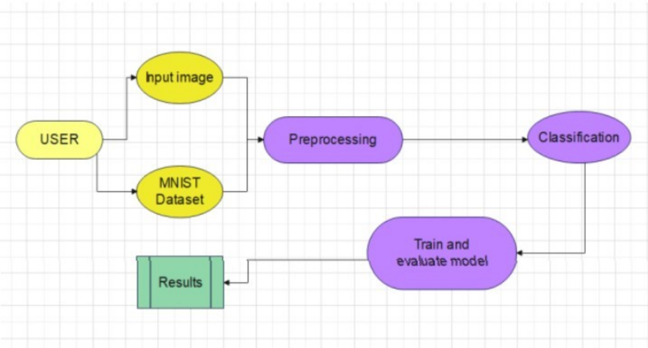
NFR No.	Non-Functional Requirement
NFR-1	**Usability:** Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc. One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.
NFR-2	**Reliability:** 1) The system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style. 2)The generative models can perform recognition driven segmentation. 3) The method involves a relative.
NFR-3	**Performance:** The neural network uses the examples to automatically infer rules forrecognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. There are a number of ways and algorithms to recognize handwritten digits, including Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees,Random Forests, etc.
NFR-4	**Accuracy:** Optical Character Recognition (OCR) technology provides higher than 99% accuracy with typed characters in high quality images. However, the diversity in human writing types, spacing differences, and irregularities of handwriting

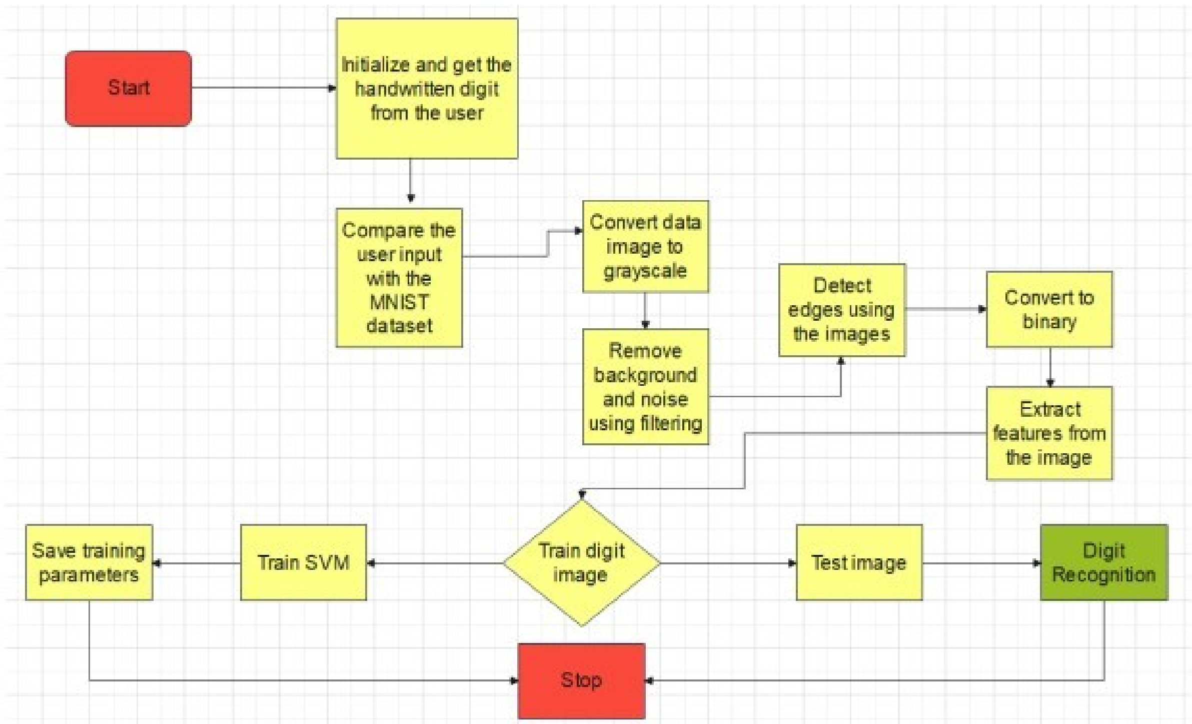
CHAPTER 5

PROJECT DESIGN

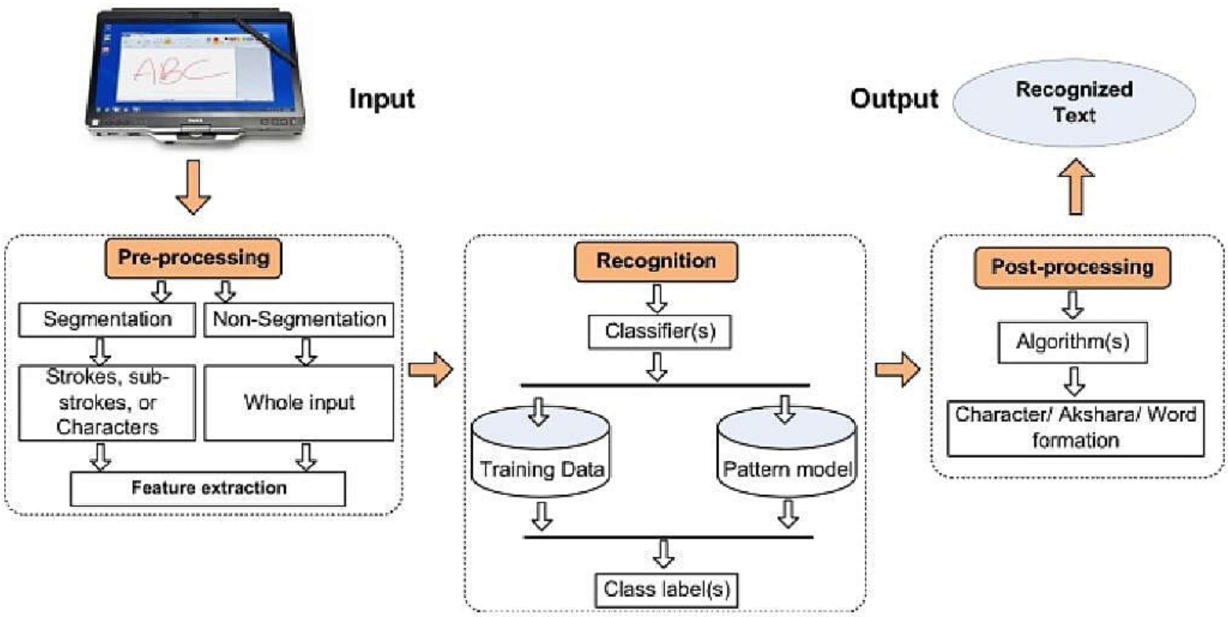
5.1 DATA FLOW DIAGRAM

Example: (Simplified)





5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Any common individual		USN-1	Receiving the digital form of the handwritten digits with a very high accuracy	Either write it on the webpage or scan the image of the written digit	High	Sprint-1
Bank officials	Separate registration	USN-2	Helps in understanding the amount and account number entered in demand draft and cheques in banks	Useful in characterizing the digits in banks	High	Sprint-2

Customer (Web user)	Home	USN-3	As a user, I can view the guide to use the webapp	I can view the awarenessof this application and its limitations.	Low	Sprint-1
		USN-4	As it is a web application, it is installation free	I can use it without the installation of the application or any software	Medium	Sprint-1
Any common person	Login	USN-5	As a user, I can log into the application byentering email & password		Low	Sprint-1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNINGAND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Kamalahasan A Varatharaj S
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Shihaf Ahil M Sugumar S
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Kamalahasan A Shihaf Ahil M Varatharaj S
Sprint-2	Add CNN	USN-4	Creating the	5	High	Shihaf Ahil M

	layers		model and adding the input, hidden, and output layersto it.			Sugumar S
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Sugumar S
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Kamalahasan A Shihaf Ahil M Sugumar S
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Kamalahasan A Varatharaj S
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High	Kamalahasan A Shihaf Ahil M Varatharaj S

Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Sugumar S
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Kamalahasan A Shihaf Ahil M
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High	Kamalahasan A Varatharaj S
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Kamalahasan A Sugumar S

6.2 SPRINT DELIVERY SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETE (AS ON PLANNED DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint - I	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint - II	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint - III	20	6 Days	07 Oct 2022	12 Nov 2022	20	12 Nov 2022
Sprint - IV	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

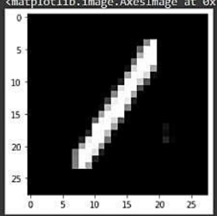
CHAPTER 7

CODING & SOLUTIONING

```
[1] from keras.datasets import mnist
    (X_train, y_train), (X_test, y_test) = mnist.load_data()

    Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
    11490434/11490434 [==============================] - 0s 0us/step
```

```
[2] print("Training Images:", len(X_train))
    print("Testing Images:", len(X_test))
    print("Shape:", X_train[0].shape)

    Training Images: 60000
    Testing Images: 10000
    Shape: (28, 28)
```

```
[3] import random
    import matplotlib.pyplot as plt
    random_image = random.choice(X_train)
    plt.imshow(random_image, cmap="gray")

    <matplotlib.image.AxesImage at 0x7eff1b680e90>
```



```
[4] X_train = X_train.reshape(X_train.shape[0], 28, 28, 1)
    X_test = X_test.reshape(X_test.shape[0], 28, 28, 1)
    X_train.shape

    (60000, 28, 28, 1)
```

```
[5] X_train = X_train / 255.
    X_test = X_test / 255.
```

```
[6] import numpy as np
    X_train = X_train.astype(np.float32)
    X_test = X_test.astype(np.float32)
```

```
[7] import tensorflow as tf
    from tensorflow.keras import layers

    model = tf.keras.Sequential([
      layers.Conv2D(filters=10,
                    kernel_size=3,
                    activation="relu",
                    input_shape=(28, 28, 1)),

      layers.Conv2D(10, 3, activation="relu"),
      layers.MaxPool2D(),

      layers.Conv2D(10, 3, activation="relu"),
      layers.Conv2D(10, 3, activation="relu"),
      layers.MaxPool2D(),

      layers.Flatten(),
      layers.Dense(10, activation="softmax")
    ])
```

```
[8] model.compile(loss="sparse_categorical_crossentropy",
                  optimizer=tf.keras.optimizers.Adam(),
                  metrics=["accuracy"])
```

```
[9]  model.fit(X_train, y_train, epochs=3)
     model.evaluate(X_test, y_test)

     Epoch 1/3
     1875/1875 [==============================] - 73s 39ms/step - loss: 0.2661 - accuracy: 0.9169
     Epoch 2/3
     1875/1875 [==============================] - 69s 37ms/step - loss: 0.0909 - accuracy: 0.9726
     Epoch 3/3
     1875/1875 [==============================] - 74s 39ms/step - loss: 0.0694 - accuracy: 0.9791
     313/313 [==============================] - 4s 13ms/step - loss: 0.0516 - accuracy: 0.9832
     [0.05156530812382698, 0.9832000136375427]
```

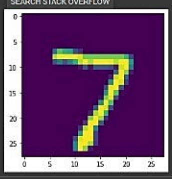
```
[10] !pip3 install --extra-index-url https://google-coral.github.io/py-repo/ tflite_runtime

     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/, https://google-coral.github.io/py-repo/
     Collecting tflite_runtime
       Downloading tflite_runtime-2.10.0-cp37-cp37m-manylinux2014_x86_64.whl (2.5 MB)
          |████████████████████████████████| 2.5 MB 5.2 MB/s
     Requirement already satisfied: numpy>=1.19.2 in /usr/local/lib/python3.7/dist-packages (from tflite_runtime) (1.21.6)
     Installing collected packages: tflite-runtime
     Successfully installed tflite-runtime-2.10.0
```

```
[11] converter = tf.lite.TFLiteConverter.from_keras_model(model)
     tflite_model = converter.convert()
     with open('model.tflite', 'wb') as f:
       f.write(tflite_model)

     WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showi
```

```
[12] import tflite_runtime.interpreter as tflite
     interpreter = tflite.Interpreter(model_path="model.tflite")
     interpreter.allocate_tensors()
```

```
[13] input_index = interpreter.get_input_details()[0]['index']
     output_index = interpreter.get_output_details()[0]['index']
```

```
plt.imshow(X_test[0].reshape(28, 28))
input_data = X_test[0].reshape(-1, 28, 28, 1)
interpreter.set_tensor(input_index, input_data)
interpreter.invoke()
output_data = interpreter.get_tensor(output_index)
pred = output_data.argmax()
print("Prediction:", pred);
print("Confidence:", output_data[pred]);

Prediction: 7
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-14-976e5afb57dc> in <module>
      6 pred = output_data.argmax()
      7 print("Prediction:", pred);
----> 8 print("Confidence:", output_data[pred]);

IndexError: index 7 is out of bounds for axis 0 with size 1

SEARCH STACK OVERFLOW
```



CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	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 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a nonimage file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if page redirects result page once input is given	The page should redirect to the results page	Working as expected	PASS
BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properlywork	Working as expected	PASS

M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model shouldrescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts thedigit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model shouldpredict the number in thecomplex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	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_003	UI	Result Page	Check if the result is displayed properly	The result should bedisplayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions shouldbe 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	Subtotal
By Design	0	4	2	3	9
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	20	0	1	0	21
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	2	2
Won't Fix	2	0	0	0	2
Totals	25	7	7	7	45

8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Login Page	10	3	0	7
User Interface	3	0	1	2
Model Endpoints	2	1	0	1
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	3	0	0	3

CHAPTER 9
RESULTS

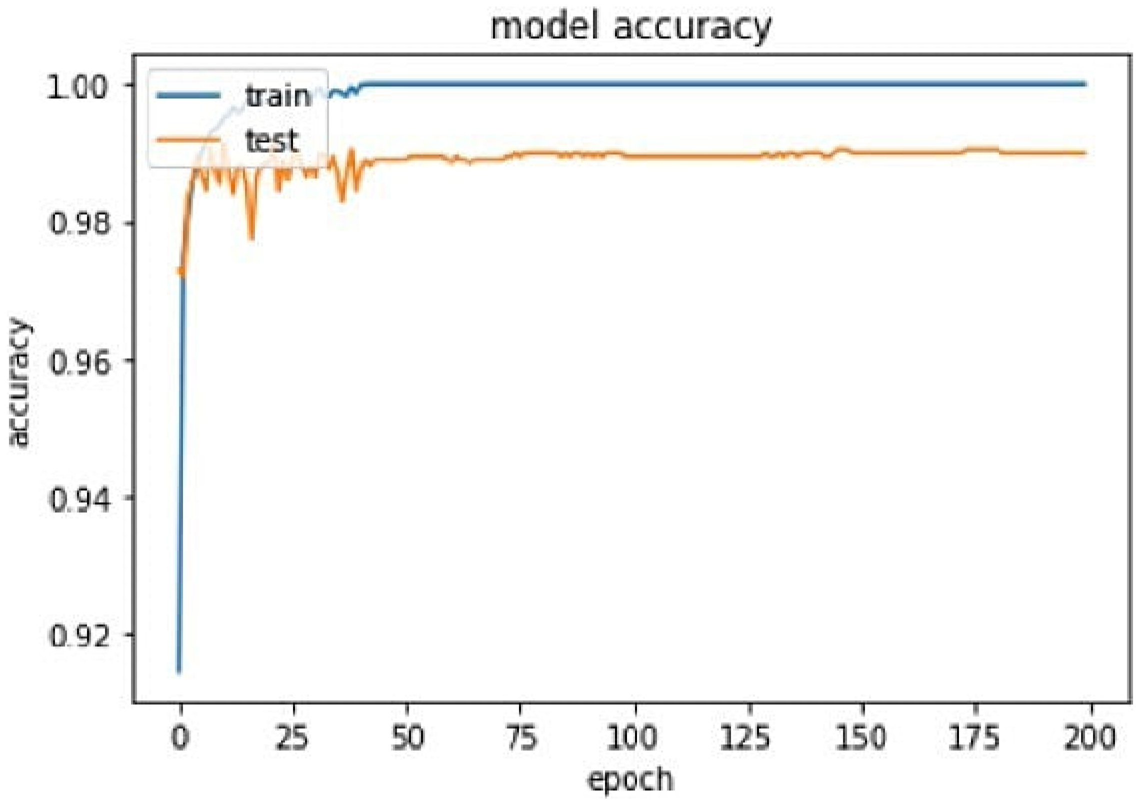
9.1 PERFORMANCE METRICS

9.1.1 MODEL SUMMARY

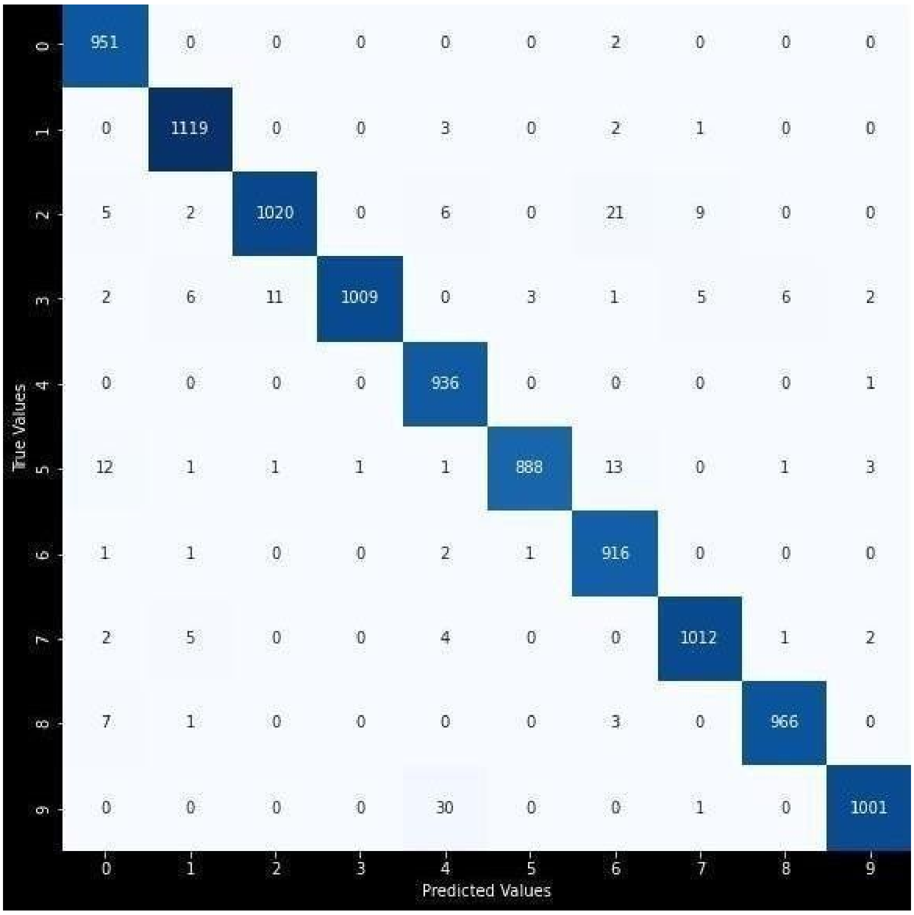
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 6)         156
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 6)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 10, 10, 16)        2416
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 400)               0
_________________________________________________________________
dense (Dense)                (None, 120)               48120
_________________________________________________________________
dense_1 (Dense)              (None, 84)                10164
_________________________________________________________________
dense_2 (Dense)              (None, 10)                850
=================================================================
Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0
_________________________________________________________________
```

9.1.2 ACCURACY

CONTENT	VALUE
Training Accuracy	99.2%
Validation Accuracy	99.8 %
Epochs Trained	200



9.1.3 CONFUSION MATRIX



9.1.4 CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	1.00	0.97	0.98	980
1	0.99	0.99	0.99	1135
2	0.96	0.99	0.97	1032
3	0.97	1.00	0.98	1010
4	1.00	0.95	0.98	982
5	0.96	1.00	0.98	892
6	0.99	0.96	0.97	958
7	0.99	0.98	0.99	1028
8	0.99	0.99	0.99	974
9	0.97	0.99	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

9.1.5 APPLICATION TEST REPORT

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/

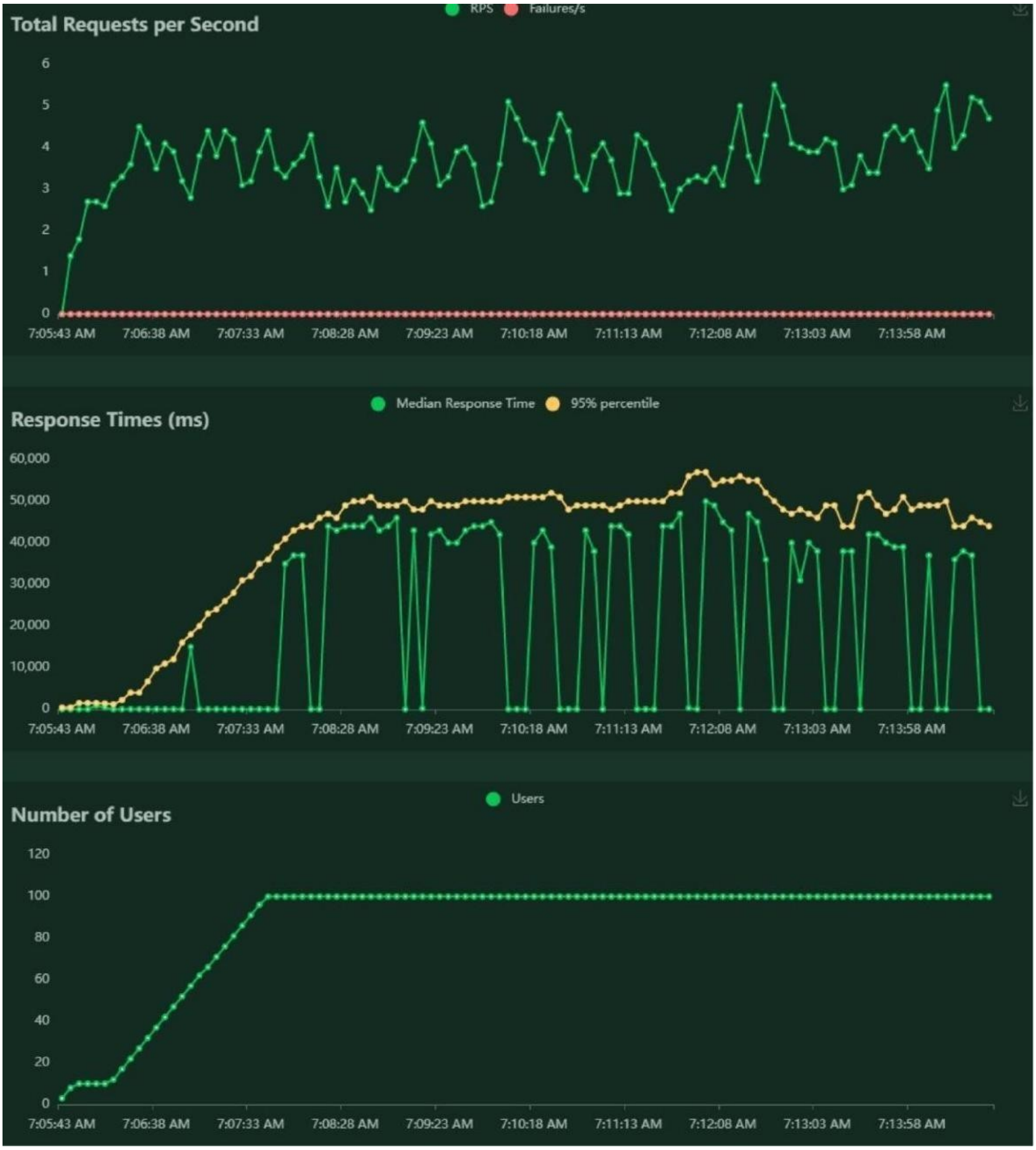
Script: locust.py

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**	**36000**	**43000**	**45000**	**48000**	**50000**	**54000**	**60000**


Total Requests per Second
RPS
Failures/s
0
1
2
3
4
5
6
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
0
10,000
20,000
30,000
40,000
50,000
60,000
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
0
20
40
60
80
100
120
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

1. Can be used anywhere from any device
2. Manual work can be reduced
3. Lot of data can beadded
4. Tends to be more accurate than humans

DISADVANTAGES

5. Complex data can’t behandled
6. Digital format is expected
7. Requires a fast server
8. Occasional errors might occur

CHAPTER 11

CONCLUSION

This project shows and suggests a web app that uses machine learning to identify handwritten numbers. The tech stack used here for the projectare Flask, HTML,CSS, JavaScript. CNN network model is used to predict the handwritten digits. The model achieved a 99.61% recognition rate during the testing. This project is scalable and caneasily handle a huge number of users.

This system is compatible with any device that can run a browser since it is a web application. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts,numeric entries in forms filled up by hand (tax forms) and so on.

In subsequent versions much more improvement can be made.

CHAPTER 12

FUTURE SCOPE

This projecthas a lot of room for improvement and improvements can be made inthe next versions. Some of the improvements that can be made to this projectare:

- Multiple digits detection support can be added.
- Model to detectdigits from complex imagescan be improved.
- Add support to detect from digits multipleimages and save the results
- Multilingual support can be added to help users from all over the world

Implementing this concept in the real world will benefit severalindustries and reducetheworkload on many workers, enhancing overall work efficiency. This system has endless advancement in the next versions and can alwaysbe improved to be betterthan this.

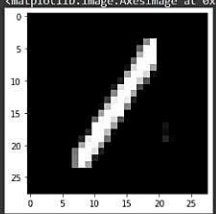
APPENDIX

SOURCE CODE

MODEL CREATION

```
[9]  model.fit(X_train, y_train, epochs=3)
     model.evaluate(X_test, y_test)

     Epoch 1/3
     1875/1875 [==============================] - 73s 39ms/step - loss: 0.2661 - accuracy: 0.9169
     Epoch 2/3
     1875/1875 [==============================] - 69s 37ms/step - loss: 0.0909 - accuracy: 0.9726
     Epoch 3/3
     1875/1875 [==============================] - 74s 39ms/step - loss: 0.0694 - accuracy: 0.9791
     313/313 [==============================] - 4s 13ms/step - loss: 0.0516 - accuracy: 0.9832
     [0.05156530812382698, 0.9832000136375427]

[10] !pip3 install --extra-index-url https://google-coral.github.io/py-repo/ tflite_runtime

     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/, https://google-coral.github.io/py-repo/
     Collecting tflite_runtime
       Downloading tflite_runtime-2.10.0-cp37-cp37m-manylinux2014_x86_64.whl (2.5 MB)
          |████████████████████████████████| 2.5 MB 5.2 MB/s
     Requirement already satisfied: numpy>=1.19.2 in /usr/local/lib/python3.7/dist-packages (from tflite_runtime) (1.21.6)
     Installing collected packages: tflite-runtime
     Successfully installed tflite-runtime-2.10.0

[11] converter = tf.lite.TFLiteConverter.from_keras_model(model)
     tflite_model = converter.convert()
     with open('model.tflite', 'wb') as f:
       f.write(tflite_model)

     WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showi
```

```
[3]  import random
     import matplotlib.pyplot as plt
     random_image = random.choice(X_train)
     plt.imshow(random_image, cmap="gray")

     <matplotlib.image.AxesImage at 0x7eff1b680e90>
```



```
[4]  X_train = X_train.reshape(X_train.shape[0], 28, 28, 1)
     X_test = X_test.reshape(X_test.shape[0], 28, 28, 1)
     X_train.shape

     (60000, 28, 28, 1)

[5]  X_train = X_train / 255.
     X_test = X_test / 255.
```

```
[6]  import numpy as np
     X_train = X_train.astype(np.float32)
     X_test = X_test.astype(np.float32)
```

```
[7]  import tensorflow as tf
     from tensorflow.keras import layers

     model = tf.keras.Sequential([
       layers.Conv2D(filters=10,
                     kernel_size=3,
                     activation="relu",
                     input_shape=(28, 28, 1)),

       layers.Conv2D(10, 3, activation="relu"),
       layers.MaxPool2D(),

       layers.Conv2D(10, 3, activation="relu"),
       layers.Conv2D(10, 3, activation="relu"),
       layers.MaxPool2D(),

       layers.Flatten(),
       layers.Dense(10, activation="softmax")
     ])
```

```
[8]  model.compile(loss="sparse_categorical_crossentropy",
                   optimizer=tf.keras.optimizers.Adam(),
                   metrics=["accuracy"])
```

```
[9]  model.fit(X_train, y_train, epochs=3)
     model.evaluate(X_test, y_test)

     Epoch 1/3
     1875/1875 [==============================] - 73s 39ms/step - loss: 0.2661 - accuracy: 0.9169
     Epoch 2/3
     1875/1875 [==============================] - 69s 37ms/step - loss: 0.0909 - accuracy: 0.9726
     Epoch 3/3
     1875/1875 [==============================] - 74s 39ms/step - loss: 0.0694 - accuracy: 0.9791
     313/313 [==============================] - 4s 13ms/step - loss: 0.0516 - accuracy: 0.9832
     [0.05156530812382698, 0.9832000136375427]
```

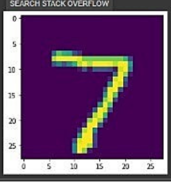
```
[10] !pip3 install --extra-index-url https://google-coral.github.io/py-repo/ tflite_runtime

     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/, https://google-coral.github.io/py-repo/
     Collecting tflite_runtime
       Downloading tflite_runtime-2.10.0-cp37-cp37m-manylinux2014_x86_64.whl (2.5 MB)
          |████████████████████████████████| 2.5 MB 5.2 MB/s
     Requirement already satisfied: numpy>=1.19.2 in /usr/local/lib/python3.7/dist-packages (from tflite_runtime) (1.21.6)
     Installing collected packages: tflite-runtime
     Successfully installed tflite-runtime-2.10.0
```

```
[11] converter = tf.lite.TFLiteConverter.from_keras_model(model)
     tflite_model = converter.convert()
     with open('model.tflite', 'wb') as f:
       f.write(tflite_model)

     WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showi
```

```
[12] import tflite_runtime.interpreter as tflite
     interpreter = tflite.Interpreter(model_path="model.tflite")
     interpreter.allocate_tensors()
```

```
[13] input_index = interpreter.get_input_details()[0]['index']
     output_index = interpreter.get_output_details()[0]['index']
```

```
plt.imshow(X_test[0].reshape(28, 28))
input_data = X_test[0].reshape(-1, 28, 28, 1)
interpreter.set_tensor(input_index, input_data)
interpreter.invoke()
output_data = interpreter.get_tensor(output_index)
pred = output_data.argmax()
print("Prediction:", pred);
print("Confidence:", output_data[pred]);

Prediction: 7
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-14-976e5afb57dc> in <module>
      6 pred = output_data.argmax()
      7 print("Prediction:", pred);
----> 8 print("Confidence:", output_data[pred]);

IndexError: index 7 is out of bounds for axis 0 with size 1

SEARCH STACK OVERFLOW
```



FLASK APP

app.py 1 × model.py 4

app.py > predict_digit

```python
from flask import Flask, request, render_template, jsonify
import base64, os, random
from uuid import uuid4
from model import predict

app = Flask(__name__)

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

@app.route("/predict-digit", methods=["POST", "GET"])
def predict_digit():
    image = request.get_json(silent=True)['image'].split(",")[1]
    image_data = base64.urlsafe_b64decode(image)

    prediction, confidence = predict(image_data)

    response = {
        "prediction": str(prediction),
        "confidence": str(confidence)
    }

    return jsonify(response)
```

RECOGNIZER

app.py 1 model.py 4 ×

model.py > ...

```python
import tflite_runtime.interpreter as tflite
from PIL import Image
import numpy as np
import os, io


interpreter = tflite.Interpreter(model_path="model.tflite")
interpreter.allocate_tensors()

input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

def model_predict(img):
    input_data = img.astype(np.float32)
    interpreter.set_tensor(input_details[0]['index'], input_data)
    interpreter.invoke()
    output_data = interpreter.get_tensor(output_details[0]['index'])
    return output_data.argmax(), output_data[0][output_data.argmax()]

def preprocess_image(image, img_type="file"):
    if img_type == "file":
        return (image.reshape(1, 28, 28, 1) / 255.).astype(np.float32)
    elif img_type == "df_row":
        return (image.to_numpy().reshape(1, 28, 28, 1) / 255.).astype(np.float32)

def predict(image_data):
    # Import image
    image = Image.open(io.BytesIO(image_data))

    # Convert the RGB image to grayscale image
    image = image.convert("L")

    # Resize the image to 28x28
    image = image.resize((28, 28))

    # Convert the image into numpy array
    image = np.array(image)

    # Reshape the image for the model
    image = image.reshape(1, 28, 28, 1)

    # Normalize the pixel values in image
    image = image / 255.

    # Set the datatype of image as float32
    image = image.astype(np.float32)

    # Make prediction on the image
    prediction, confidence = model_predict(image)

    return prediction, confidence
```

HOME PAGE (HTML)

```html
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>A Novel Method For Handwritten Digit Recognition System</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="static\style.css">

</head>
<body>
<main>
    <div class="container">
        <h1>A Novel Method For Handwritten Digit Recognition System</h1>
        <h2>Team ID : PNT2022TMID07022</h2>
        <div class="search-box">
            <div class="draw-title">
                Draw<br>a digit:
            </div>
            <div class="canvas-div">
                <canvas id="canvas" width="112px" height="112px"></canvas>
            </div>
            <div class="buttons">
                <button class="button" id="predictButton">Predict</button>
                <button class="button" id="clearButton">Clear</button>
            </div>
            <div class="prediction-container">
                <h3 class="prediction-heading">Prediction:</h3>
                <div class="prediction-label">
                    <div class="prediction" id="prediction">?</div>
           <div class="confidence" id="confidence"></div>
                </div>
            </div>
        </div>
    </div>
</main>
<script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/1.7.1/jquery.min.js'></script>
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script  src="/static/script.js"></script>

</body>
</html>
```

HOME PAGE (CSS)

```css
html, body {
    font-family: 'Poppins', sans-serif;
    color: #FFFFFF;
    width: 100%;
    max-width: 100%;
    height: 100%;
    padding: 0;
    margin: 0;
  }

  *, *:before, *:after {
    -webkit-tap-highlight-color: transparent;
    -webkit-tap-highlight-color: rgba(0,0,0,0);
    user-select: none;
    -webkit-user-select: none;
    -khtml-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    -o-user-select: none;
    box-sizing: border-box;
    -webkit-box-sizing: border-box;
    padding: 0;
    margin: 0;
  }
  a, a:visited, a:hover {
    color: inherit;
    text-decoration: none;
  }
  main {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(154,57,163,1);
    background: -moz-linear-gradient(-45deg, rgba(154,57,163,1) 0%, rgba(65,103,168,1) 100%);
    background: -webkit-gradient(left top, right bottom, color-stop(0%, rgba(154,57,163,1)), color-stop(100%, rgba(65,103,168,1)));
    background: -webkit-linear-gradient(-45deg, rgba(154,57,163,1) 0%, rgba(65,103,168,1) 100%);
    background: -o-linear-gradient(-45deg, rgba(154,57,163,1) 0%, rgba(65,103,168,1) 100%);
    background: -ms-linear-gradient(-45deg, rgba(154,57,163,1) 0%, rgba(65,103,168,1) 100%);
    background: linear-gradient(135deg, rgba(154,57,163,1) 0%, rgba(65,103,168,1) 100%);
  }
  h1 {
    display: block;
```

JAVASCRIPT FILE (JS):

```
app.py 1    model.py 4    home.html    style.css    script.js
script.js > ...
const predictButton = document.getElementById("predictButton");
const clearButton = document.getElementById("clearButton");
const predictionEl = document.getElementById("prediction");
const confidenceEl = document.getElementById("confidence");

let context, canvas;

$(document).ready(function() {
  initialize();
});

function getPosition(mouseEvent, sigCanvas) {
    var rect = sigCanvas.getBoundingClientRect();
    return {
      X: mouseEvent.clientX - rect.left,
      Y: mouseEvent.clientY - rect.top
    };
}


function initialize() {
  var sigCanvas = document.getElementById("canvas");
  canvas = sigCanvas;
  context = sigCanvas.getContext("2d");
  context.strokeStyle = "#fff";
  context.lineJoin = "round";
  context.lineWidth = 7;

  context.fillRect(0, 0, sigCanvas.width, sigCanvas.height);

  var is_touch_device = 'ontouchstart' in document.documentElement;

  if (is_touch_device) {
    var drawer = {
      isDrawing: false,
      touchstart: function(coors) {
        context.beginPath();
        context.moveTo(coors.x, coors.y);
        this.isDrawing = true;
      },
      touchmove: function(coors) {
        if (this.isDrawing) {
          context.lineTo(coors.x, coors.y);
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-3099-1658500984 Recognition System

PROJECT DEMO

https://drive.google.com/file/d/1aHQ_oQjg_c7DHKdnvAmmmldt8cEf7XeM/view?us
p=drivesd k