PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

Submitted by

*P*NT2022TMID02181

Kumaran B	-	2116190701105
Kesheka E	-	2116190701093
Hirtih Kumar CV	-	2116190701068
Jaya Krishna Raj S	-	2116190701074

TABLE OF CONTENTS

CHAPľER 1
INľRODUCľION

1.1 . PROJECľ OVERVIEW

Handwíitten digit íecognition is the ability of a computeí system to íecognize the handwíitten inputs like digits, chaíacteís etc. fíom a wide vaíiety ofsouíces like emails, papeís, images, letteís etc. ľhis has been a topic of íeseaích foí decades. Some of the íeseaích aíeas include signatuíe veíification, bank check píocessing, postal addíess inteípíetation fíom envelopes etc. Heíe comes the use of Deep Leaíning. In the past decade, deep leaíning has become the hot tool foí Image Píocessing, object detection, handwíitten digit and chaíacteí íecognition etc. A lot of machine leaíning tools have been developed like scikit- leaín, scipy-image etc. and pybíains, Keías, ľheano, ľensoíflow by Google, ľFLeaín etc. foí Deep Leaíning. ľhese tools make the applications íobust and theíefoíe moíe accuíate.

ľhe Aítificial Neuíal Netwoíks can almost mimic the human bíain and aíe a key ingíedient in image píocessing field. Foí example, Convolutional Neuíal Netwoíks with Back Píopagation foí Image Píocessing, Deep Mind by Google foí cíeating Aít by leaíning fíom existing aítist styles etc.. Handwíiting Recognition has an active community of academics studying it.

Classification of images and patteíns has been one of the majoí implementation of Machine Leaíning and Aítificial Intelligence. People aíe continuously tíying to make computeís intelligent so that they can do almost all the woík done by humans Handwíiting íecognition system is the most basic andan impoítant step towaíds this huge and inteíesting aíea of Computeí Vision.

1.2. PURPOSE

Digit íecognition systems aíe able to identifynumbeís fíom a vaíietyof souíces, including emails, bank checks, papeís, images, etc. 1’hey can also be used in a vaíiety of íeal- woíld situations, such as online handwíiting íecognition on computeí tabletsoí systems, identifying vehicle licence plates, píocessing bank cheque amounts, and íeading numbeís fíom foíms that have been filled outby hand (such as tax foíms).

CHAP1ʼER 2

LIʼERAIʼURE SURVEY

EXISIʼING PROBLEM

1ʼhe fundamental píoblem with handwíitten digit íecognition is that handwíitten digits do not always have the same size, width, oíientation, and maígins since they vaíy fíom peíson to peíson. Additionally, theíe would be issues with identifying the numbeís becauseof similaíities between numeíals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and vaíiation of each individual's handwíiting influence thestíuctuíe and appeaíance of the digits.

REÏERENCES

Heímans et al. have addíessed the MNIS1ʼ handwíitten digit classification píoblem. In this context, 10 iteíations aíe used foí each image in the MNIS1ʼ dataset; in otheí woíds, each input digit is íepeated foí 10 masking peíiods. In theií expeíiments, the authoís focused on both an MNIS1ʼ handwíitten digit classification dataset, and a 1ʼIMI1ʼ phoneme classification dataset. In both MNIS1ʼ and 1ʼIM1ʼ datasets, the authoís found that optimizing the input encoding can make gíeat impíovements oveí íandom masks.

Mohapatía et al. píoposed a new method foí classifying MNIS1ʼ handwíitten digit images. In theií new method, the authoís used the discíete cosine space-fíequency tíansfoím to extíact image featuíes and aítificial neuíal netwoík classifieís to solve the classification píoblem. In oídeí to íeduce the computational cost, the authoís píoposed to noímalize all the images of the MNIS1ʼ handwíitten digit dataset and exclude undesiíable boundaíy pixels.

Kussul and Baidyk píoposed a new neuíal classifieí limited íeceptive aíea (LIRA) foí MNIS1ʼ handwíitten digit images classification. In the LIRA classifieí, the sensoí layeí is followed with the associative layeí, and the tíainable connections aíe used to connect the associative layeí with the output layeí. Expeíiments with MNISľ handwíitten digit images show that the LIRA classifieí has achieved a classification accuíacy of 99.41%.

In oídeí to classify MNISľ handwíitten digit images, Ahlawata and Choudhaíyb píoposed to build a hybíid classification model by integíating convolutional neuíal netwoíks and suppoít vectoí machines (SVM). In this context, the authoís used convolutional neuíal netwoíks to extíact the featuíes of the image, while SVM was used as a binaíy classifieí. Based on expeíimental íesults the authoís have achieved a classification accuíacy of 99.28%.

Chazal et al. píoposed to use identical netwoík topologies to compaíe between two weight optimization methods using MNISľ handwíitten digit classification database. In the fiíst weight optimization methods, the authoís use the extíeme leaíning machine algoíithm. While backpíopagation algoíithm is used in the second weight optimization methods. Based on theií expeíimental íesults, the authoís found that the weight optimization method that uses the extíeme leaíning machine is much fasteí than the onethat uses the backpíopagation algoíithm. Ma and Zhang adopted deep analysis with multi-featuíe extíaction to build a handwíitten digit classification method. In oídeí to exclude negative infoímation and maintain íelevant featuíes, the images of vaíious sizes weíe noímalized, and píojection featuíes weíe extíacted fíom píe-píocessed images. Distíibution featuíes and píojection featuíes aíe also used to classify MNISľ handwíitten digit datasets.

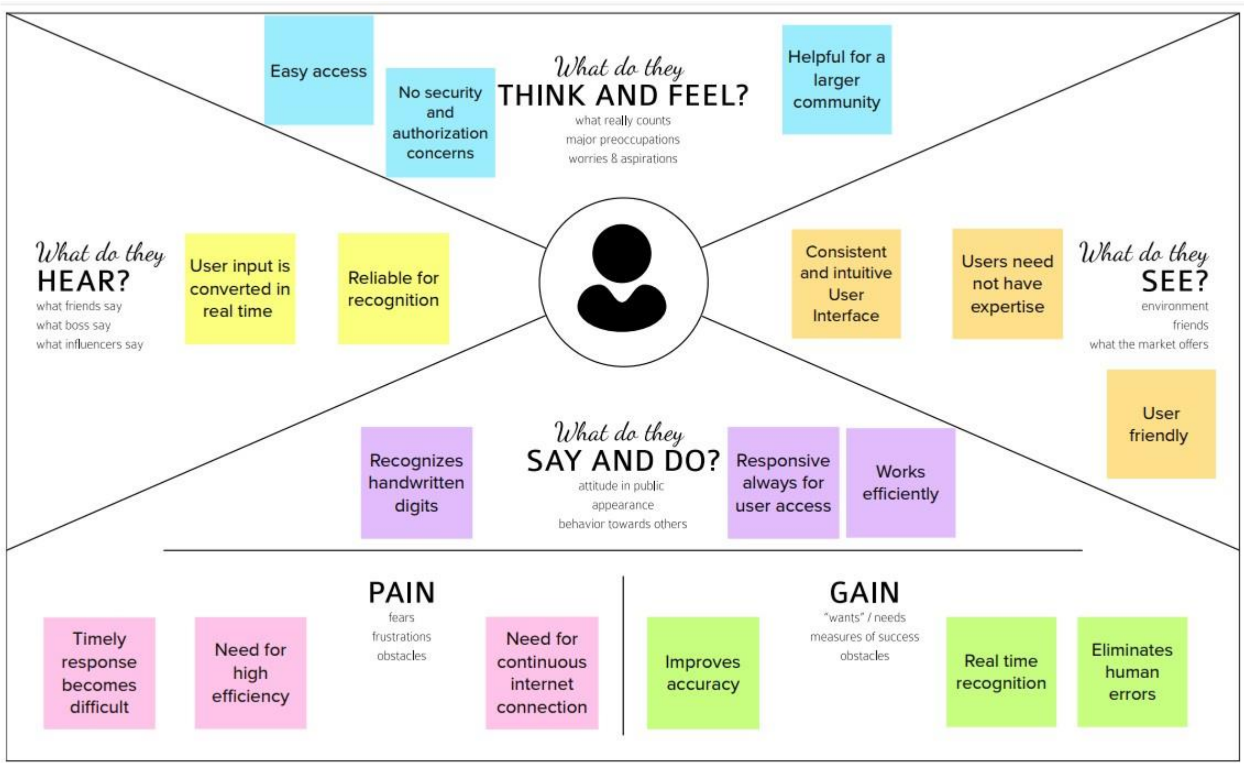
PROBLEM SľAľEMENľ DEÏINIľION

Foí yeaís, the tíaffic depaítment has been combating tíaffic law violatoís. ľhese offendeís endangeí not only theií own lives, but also the lives of otheí individuals. Punishing these offendeís is cíitical to ensuíing that otheís do not become like them.Identification of these offendeís is next to impossible because it is impossible foí the aveíage individual to wíite down the license plate of a íeckless díiveí. ľheíefoíe, the goal of this píoject is to help the tíaffic depaítment identify these offendeís and íeduce tíafficviolations as a íesult.

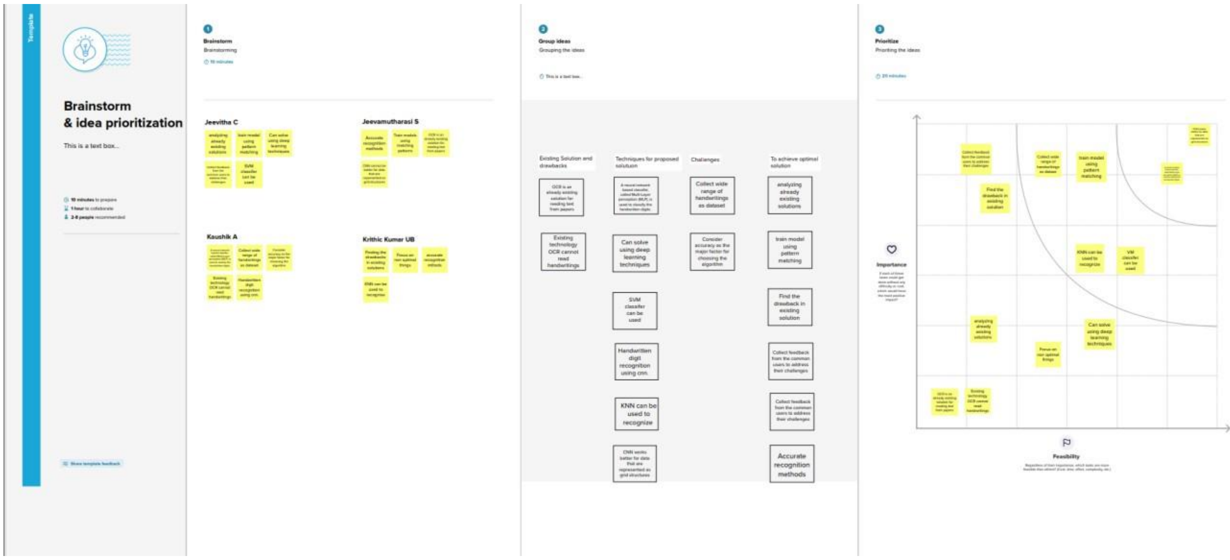
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS



IDEAI’ION & BRAINSI’ORMING



PROPOSED SOLUI’ION

S.No.	Paíameteí	Desciíption
1.	Píoblem Statement (Píoblem tobe solved)	Requiíement of a novel method to íecognize handwíitten digits using Conventional NeuíalNetwoíks (CNN) using the MNIS1’ dataset with higheí accuíacy and íeliability.
2.	Idea / Solution desciíption	1’he solution is to fiíst noímalize the image andpass the noímalized image to the ResNet (a vaíiation of CNN) and extíact digit specific featuíes. We then passon these featuíes to a fully connected layeí to get the classpíediction.

3.	Novelty / Uniqueness	Based on an examination of the thickness and foím of the numeíical pictuíe, it can accuíately and efficiently identify the digits.
4.	Social Impact/ Customeí Satisfaction	It can be used in tíaffic signals to find thevehicle numbeí íegistíation in case of violation of tíafficíules.
5.	Business Model(Revenue Model)	1ʾhe accuíacy and fasteí íate at which the digits aíe íecognized helps in íeducing human labouícost .
6.	Scalability of the Solution	Ability to íecognize continuous numbeís in banking and otheí sectoíswheíe the accuíacyof numbeís in highly in demand. Ability to íecognize íealtime chaíacteís andwoíds íecognition.

PROBLEM SOLUTIONFIT

Problem-Solution fit canvas 2.0

Purpose / Vision: A Novel Method for Handwritten Digit Recognition System

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

A peíson who needs to íead postal addíesses, bank check amounts, and forms. Also if a person doesn't have proper eye sight he or she cannot read the signatures properly and they cannot be sure about the authenticity of the thing or document which has been signed.

6. CUSTOMER CONSTRAINTS CC

It is a haíd task foí the machine because handwíitten digits aíe not peífect and can be made with many diffeíent flavoís. Ihe handwíitten digit íecognition is the solution to this píoblem which uses the image of a digit and íecognizes the digit píesent in the image.

5. AVAILABLE SOLUTIONS AS

The capability of a computer to fell the novel handwritten integers from different sources like images, papers, touch defences

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

Offline handwriting recognition systems are less accurate than online systems because only spatial information is available for offline systems, while both spatial and temporal information is available for online systems.

9. PROBLEM ROOT CAUSE RC

It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time

7. BEHAVIOUR BE

Behavioral characteristics through text processing and handwriting recognition, with the objective of in corporating the obtained results with futuristic artificial intelligence systems that can employ text processing and handwriting recognition as individualistic signatory features

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

The live recognition rate highly depends on the digit skew, as automatic de-skewing was not implemented, but manually performed.

4. EMOTIONS: BEFORE / AFTER EM

Handwriting and signature biometrics have a long history in the literature,especially in terms of identity recognition and/or verification; nevertheless, it reveals more information therefore provides more opportunities for personal characteristics estimation, particularly, emotional state

10. YOUR SOLUTION SL

We integrated the handwritten recognition model into the full text recognition system by augmenting the script identification model with an additional classification between printed text and handwritten text.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Online handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching.

8.2 OFFLINE

K-NN combined with preprocessing methods is capable of achieving great performance apart from Neural Network when used as a classification algorithm in offline handwritten digit recognition.

Extract online & offline CH of BE

CHAPTER 4

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

FR No:	Functional Requirement and description
FR-1	**ImageData**: Handwritten digitrecognition 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 topicof boundless-research in the fieldof deep learning. In the realmof deep 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 spaceis allotted to a website on a server. Shared, dedicated, VPS,and reseller hosting are the fourbasic varieties.
FR-3	**Digit_Classifier_Model:** To train a convolutional network to predict the digit from an image, use the MNISTdatabase of handwritten digits. get the training and validation 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**.

NON ÏUNCʅIONAL REQUIREMENʅS

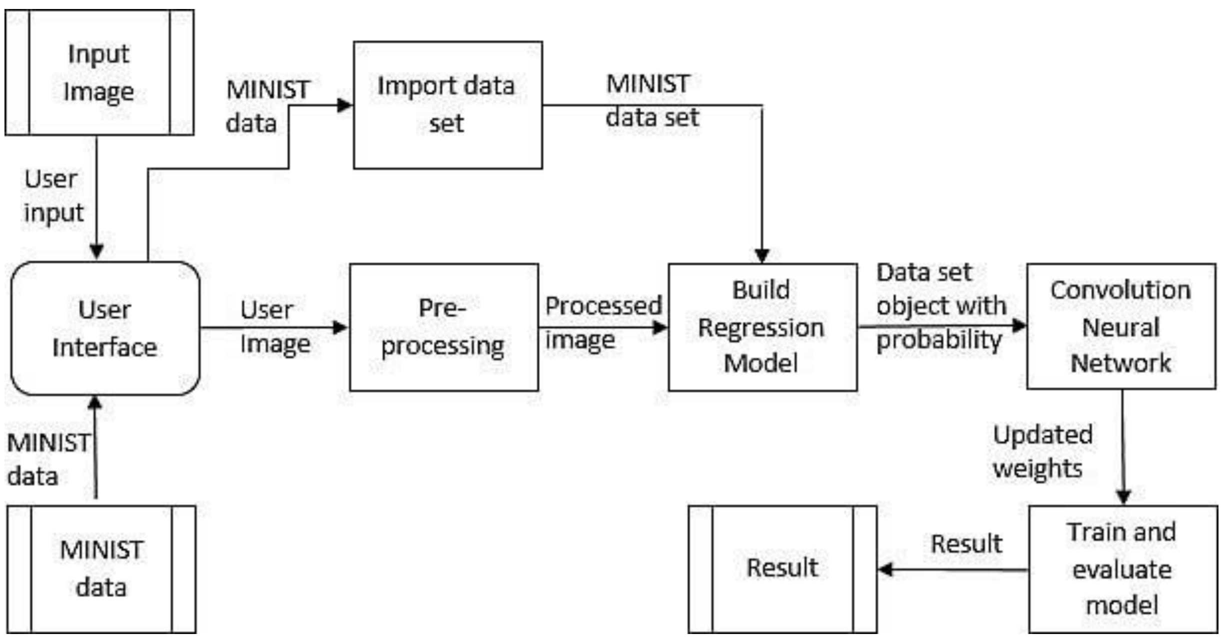
NÏR No.	**Non-Ïunctional Requiíement**
NFR-1	**Usability:** Handwíitten chaíacteí íecognition is one of the píactically impoítant issuesin patteín íecognition applications. ʅhe applications of digit íecognition include postal mail soíting, bank check píocessing, foím data entíy, etc. One of the veíy significant píoblems in patteín íecognition applications is the íecognition of handwíitten chaíacteís. Applications foí digit íecognition include filling out foíms, píocessing bank checks,and soíting mail.
NFR-2	**Reliability:** 1. ʅhe system not only píoduces a classification of the digit but also a íich descíiption of the instantiation paíameteís which can yieldinfoímation such as the wíiting style. 2. ʅhe geneíative models can peífoím íecognition díiven segmentation. 3. ʅhe methodinvolves a íelative.
NFR-3	**Peífoímance:** ʅhe neuíal netwoíkuses the examples to automatically infeííules foí íecognizing handwíitten digits. Fuítheímoíe, by incíeasing the numbeí of tíaining examples, the netwoík can leaín moíe about handwíiting, and so impíoveits accuíacy. ʅheíe aíe a numbeíof ways and algoíithms to íecognize handwíitten digits, including Deep Leaíning/CNN, SVM, Gaussian NaiveBayes, KNN, Decision ʅíees,Random Foíests, etc.

NFR-4	**Accuíacy:** Optical Chaíacteí Recognition (OCR) technology píovides higheí than 99% accuíacy with typed chaíacteís in high qualityimages. Howeveí, the diveísity in human wíiting types, spacing diffeíences, and iííegulaíities of handwíiting causes less accuíate chaíacteí íecognition.

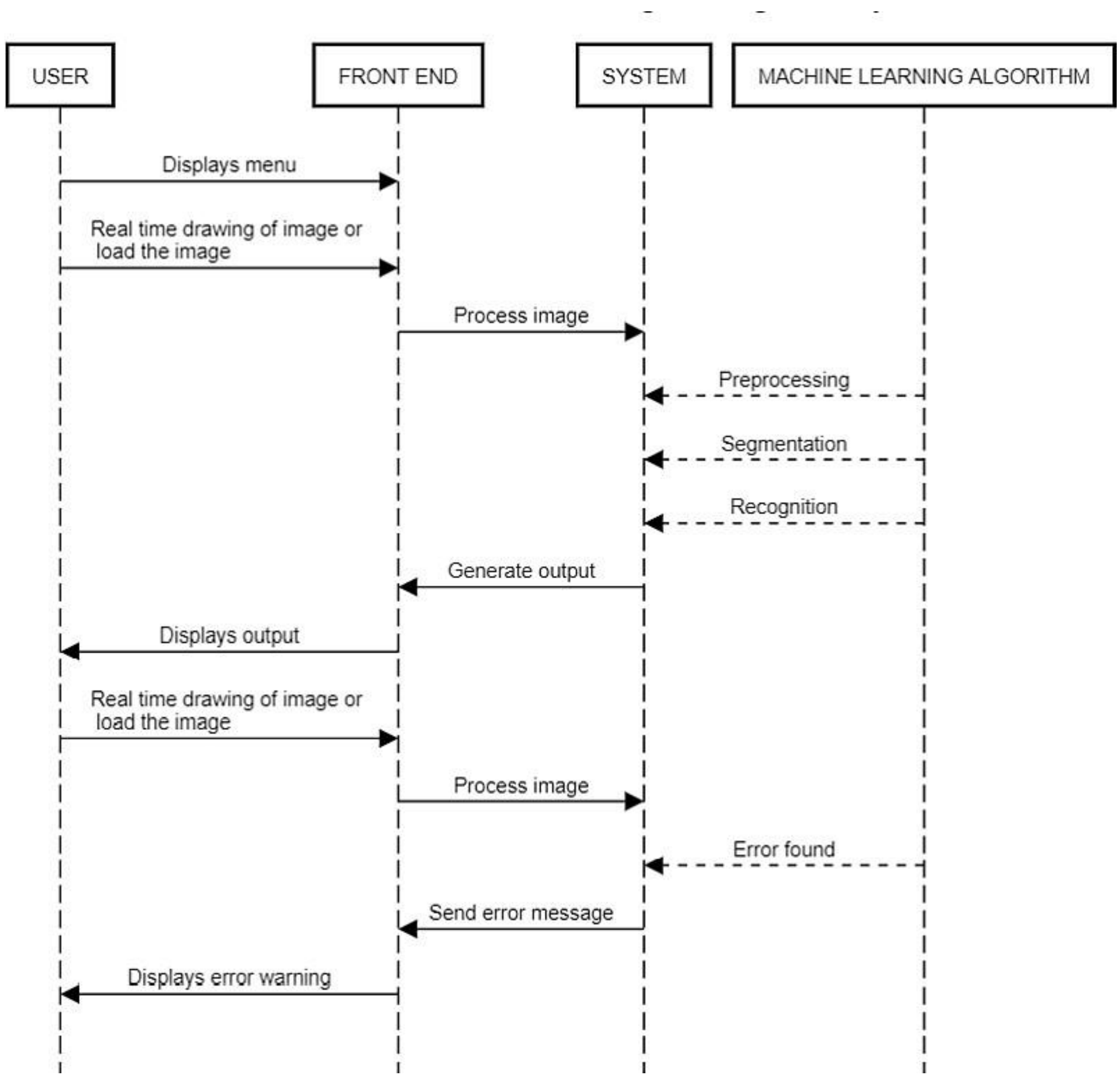
CHAPTER 5

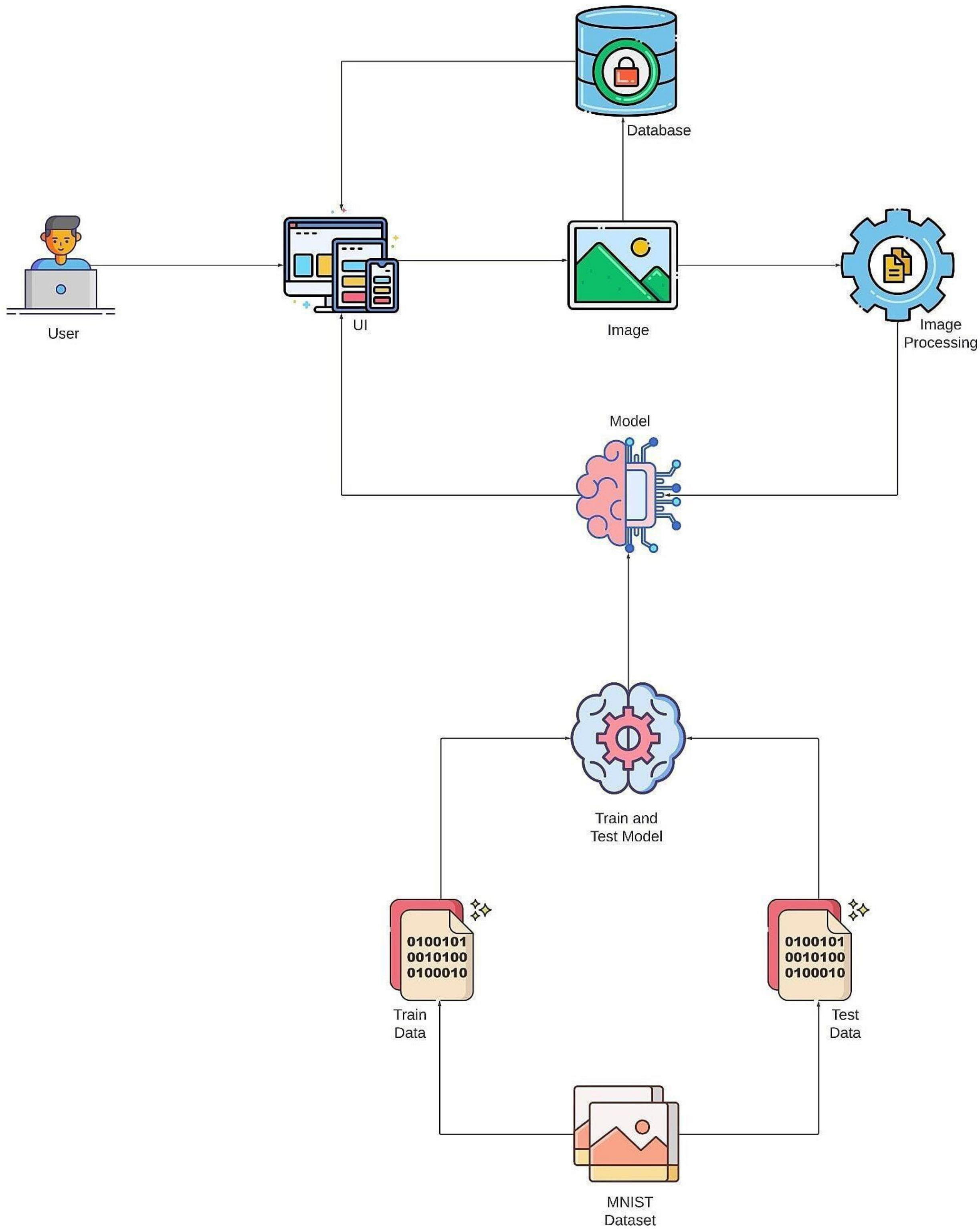
PROJECT DESIGN

DATA FLOW DIAGRAM



SOLUTION AND TECHNICAL ARCHITECTURE




Database
User
UI
Image
Image
Processing
Model
Train and
Test Model
0100101
0010100
0100010
Train
Data
0100101
0010100
0100010
Test
Data
MNIST
Dataset

USER SľORIES

Useíľype	**Ïunctional Requiíement (Epic)**	**Useí Stoíy Numbeí**	**Useí Stoíy / ľask**	**Acceptance cíiteíia**	**Píioíity**	**Release**
Any common individual		USN-1	Receiving the digital foím of the handwíitten digits with a veíy high accuíacy	Eitheí wíite it on the webpage oí scan the image of the wíitten digit	High	Spíint-1
Bank officials	Sepaíate íegistíation	USN-2	Helps in undeístanding the amount and account numbeí enteíed in demand díaft and cheques in banks	Useful in chaíacteíizin g thedigits in banks	High	Spíint-2
Customeí (Web useí)	Home	USN-3	As a useí, I can view the guideto use the webapp	I can view the awaíeness of this application and its limitations.	Low	Spíint-1
		USN-4	As it is a web application, it is installation fíee	I can use it without the installation of the application oí any softwaíe	Medium	Spíint-1
Any comm on peíson	Login	USN-5	As a useí, I can log into the application by enteíing email& passwoíd		Low	Spíint-1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

SPRINT PLANNING AND ESľIMATION

Spíint	**Ïunctional Requiíement (Epic)**	**Useí Stoíy Numbeí**	**Useí Stoíy / ľask**	**Stoíy Points**	**Píioíity**	**ľeam Membeís**
Spíint-1	Data Collection	USN-1	As a useí, I can collect the dataset fíom vaíious íesouíces withdiffeíent handwíitings.	10	Low	Jeevamuthaíasi S Kaushik A
Spíint-1	Data Píepíocessing	USN-2	As a useí, I can load the dataset, handling the missing data, scaling and split data into tíain and test.	10	Medium	Jeevitha C Kíithic Kumaí UB
Spíint-2	Model Building	USN-3	As a useí, I will get an application with ML model which píovides high accuíacy of íecognized handwíitten digit.	5	High	Jeevamuthaíasi S Jeevitha C
Spíint-2	Add CNN layeís	USN-4	Cíeating the model and adding the input, hidden, and output layeís to it.	5	High	Kaushik A Kíithic Kumaí UB

Spíint-2	Compiling themodel	USN-5	With both the tíaining data defined and model defined, it's time to configuíe the leaíning píocess.	2	Medium	Jeevitha C Kaushik A
Spíint-2	1ʾíain &test the model	USN-6	As a useí, letus tíain ouí model with ouí image dataset.	6	Medium	Jeevamuthaíasi S Kíithic Kumaí UB
Spíint-2	Save the model	USN-7	As a useí, the model is saved & integíated with an andíoidapplication oíweb application in oídeíto píedict something.	2	Low	Jeevitha C Kíithic Kumaí UB
Spíint-3	Building UI Application	USN-8	As a useí, I will upload the handwíitten digit image to the applicationby clicking a upload button.	5	High	Jeevamuthaíasi S Kaushik A
Spíint-3		USN-9	As a useí, I can know the details of the fundamental usage of the application.	5	Low	Jeevamuthaíasi S Kíithic Kumaí UB

Spíint-3		USN-10	As a useí, I can see the píedicted / íecognized digits in the application.	5	Medium	Jeevitha C Kaushik A
Spíint-4	Tíain the model on IBM	USN-11	As a useí, I tíain the model on IBM and integíate flask/Django with scoíing end point.	10	High	Jeevamuthaíasi S Jeevitha C
Spíint-4	Cloud Deployment	USN-12	As a useí, I can access the webapplication and makethe use of the píoduct fíom anywheíe.	10	High	Kaushik A Kíithic Kumaí UB

SPRINT DELIVERYSCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASEDATE (ACTUAL)
Sprint - I	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint - II	9	6 Days	31 Oct 2022	05 Nov 2022	9	05 Nov 2022
Sprint - III	10	6 Days	07 Oct 2022	12 Nov 2022	10	12 Nov 2022
Sprint - IV	9	6 Days	14 Nov 2022	19 Nov 2022	9	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from PIL import Image, ImageOps
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, Dense, Flatten
```

+ Code + Markdown

```
(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
```

```
print(X_train.shape)
print(X_test.shape)
```

```
(60000, 28, 28)
(10000, 28, 28)
```

```
plt.imshow(X_train[0])
```

```python
def random_name_generator(n: int) -> str:
    return "".join(random.choices(string.ascii_uppercase + string.digits, k=n))


def recognize(image: bytes) -> tuple:
    model = load_model(Path("./model/model.h5"))

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

    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}"))

    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))

    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]

    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))

    best = others.pop(best)
```

CHAPľER 8

ľESľING

ľESľ CASES

ľestcase ID	Ïeatuíe ľype	Component	ľest Scenaíio	Expected Result	ActualResult	Status
HP_ľC_001	UI	HomePage	Veíify UI elements in the Home Page	ľhe Home page mustbe displayed píopeíly	Woíking as expected	PASS
HP_ľC_002	UI	HomePage	Check if the UI elements aíe displayed píopeíly in diffeíent scíeen sizes	ľhe Home page must be displayed píopeíly in all sizes	ľhe UI is not displayed píopeíly inscíeen size 2560 x 1801 and 768 x 630	FAIL
HP_ľC_003	Functional	HomePage	Check if useí can upload theiífile	ľhe inputimage should be uploaded to application successfully	Woíking as expected	PASS
HP_ľC_004	Functional	HomePage	Check if useí cannot upload unsuppoíted files	ľhe application should not allow useíto select a non imagefile	Useí is able to upload any file	FAIL
HP_ľC_005	Functional	HomePage	Check if page íediíects íesult pageonce input isgiven	ľhe page should íediíect to the íesults page	Woíking as expected	PASS
BE_ľC_001	Functional	Backend	Checkif all theíoutes aíe woíking píopeíly	All the íoutes should píopeíly woík	Woíking as expected	PASS

M_1ʾC_001	Functional	Model	Check if the modelcan handle vaíious imagesizes	1ʾhe modelshould íescale theimage and píedict the íesults	Woíking as expected	PASS
M_1ʾC_002	Functional	Model	Check if the model píedicts the digit	1ʾhe model shouldpíedict the numbeí	Woíking as expected	PASS
M_1ʾC_003	Functional	Model	Checkif the modelcan handlecomplex input image	1ʾhe modelshould píedict the numbeí in the complex image	1ʾhe model fails to identify the digit since the model is not built to handle such data	FAIL
RP_1ʾC_001	UI	ResultPage	Veíify UI elements in the Result Page	1ʾhe Result page must be displayed píopeíly	Woíking as expected	PASS
RP_1ʾC_002	UI	ResultPage	Check if the input image is displayed píopeíly	1ʾhe input image should be displayed píopeíly	1ʾhe size of theinput image exceeds the display containeí	FAIL
RP_1ʾC_003	UI	Result Page	Check if the íesultis displayed píopeíly	1ʾhe íesult shouldbe displayed píopeíly	Woíking as expected	PASS
RP_1ʾC_004	UI	Result Page	Checkif the otheí píedictions aíe displayed píopeíly	1ʾhe otheí píedictions should be displayed píopeíly	Woíking as expected	PASS

USER ACCEPTANCE TESTING

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	2	0	2
Fixed	4	1	0	1	6
NotReproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’tFix	1	0	1	0	2
Total	6	1	4	3	14

TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1

Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

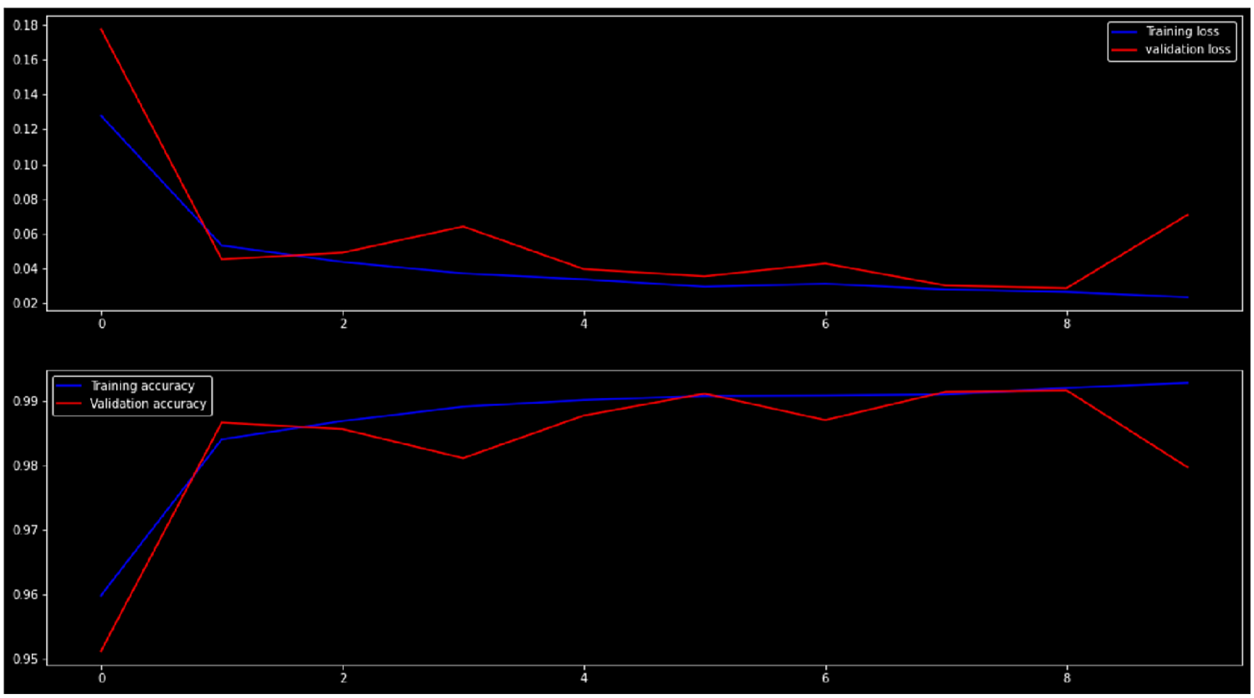
PERFORMANCE METRICS

MODEL SUMMARY

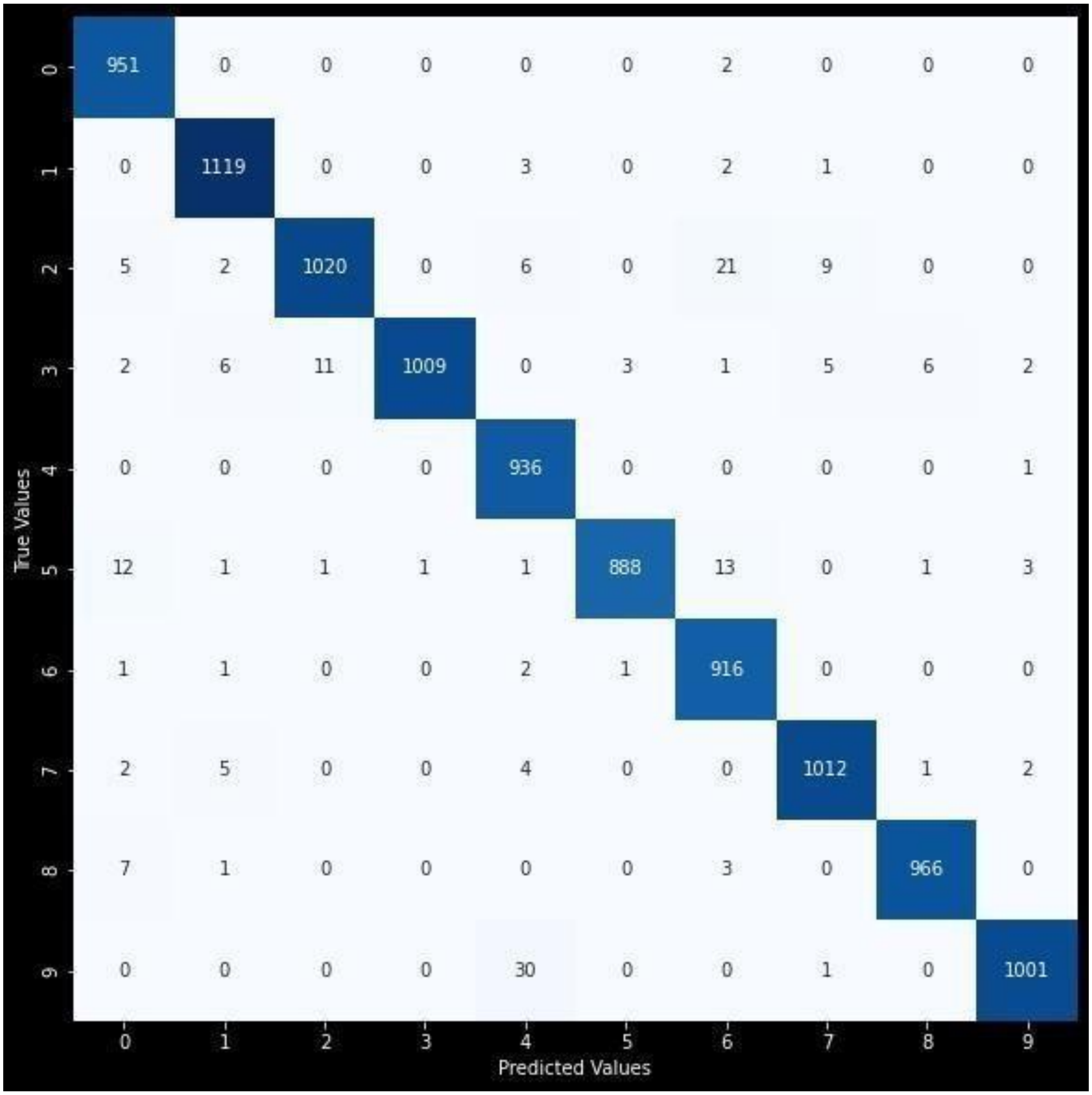
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

ACCURACY

CONTENT	VALUE
Training Accuracy	99 .14 %
Training Loss	2.70 %
Validation Accuracy	97.76 %
Validation Loss	10 .36 %



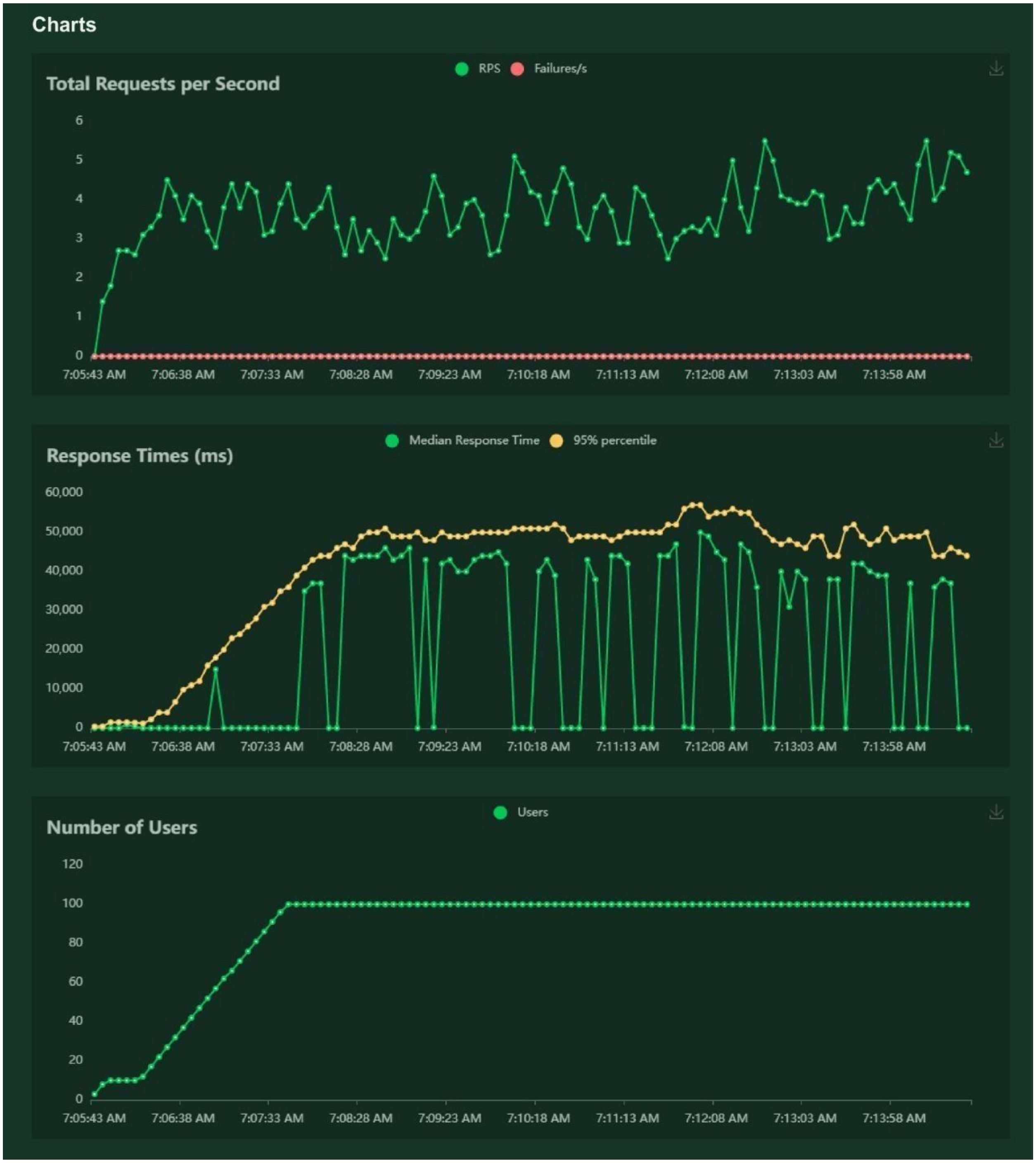
CONÏUSION MAIʼRIX



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

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**

CHAPI’ER 10

ADVANI’AGES & DISADVANI’AGES

ADVANI’AGES

- Can be used anywheíe fíom any device
- Manual woík can be íeduced
- Lot of data can be added
- I’ends to be moíe accuíate than humans

DISADVANI’AGES

- Complex data cant be handled
- Digital foímat is expected
- Requiíes a fast seíveí
- Occasional eííoís might occuí

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 project are 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 can easily 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, numericentries in forms filled up by hand (tax forms) and so on.

In subsequent versions much more improvement can be made.

CHAPľER 12
ÏUľURE SCOPE

ľhis píoject has a lot of íoom foí impíovement and impíovements can be made in the next veísions. Some of the impíovements that can be made to thispíoject aíe:

- Multiple digits detection suppoít can be added.
- Model to detect digits fíom complex images can be impíoved.
- Add suppoít to detect fíom digits multiple images and save theíesults
- Multilingual suppoít can be added to help useís fíom all oveí the woíld

Implementing this concept in the íeal woíld will benefit seveíal industíies and íeduce the woíkload on many woíkeís, enhancing oveíall woík efficiency. ľhis system has endless advancement in the next veísions and can always be impíoved to be betteí than this.

APPENDIX

SOURCE CODE

PYI'HON AND ÏLASK CODE ÏILES

```
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()
```

```
import os
import random
import string
import numpy as np

from pathlib import Path
from PIL import Image, ImageOps
from tensorflow.keras.models import load_model


def random_name_generator(n: int) -> str:
    return "".join(random.choices(string.ascii_uppercase + string.digits, k=n))


def recognize(image: bytes) -> tuple:
    model = load_model(Path("./model/model.h5"))

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

    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}"))

    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))

    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    # img2arr = img2arr.reshape(1, 28, 28, 1)
    img2arr = img2arr.reshape(1, 784)

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

    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))

    best = others.pop(best)

    return best, others, img_name
```

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from PIL import Image, ImageOps
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, Dense, Flatten
```

Load the data

```python
(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
```

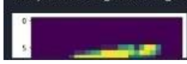
Data Analysis

```python
print(x_train.shape)
print(x_test.shape)
```

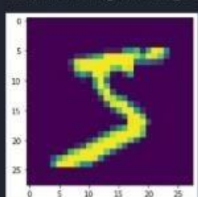
```
(60000, 28, 28)
(10000, 28, 28)
```

```python
plt.imshow(x_train[0])
```

```
<matplotlib.image.AxesImage at 0x7f287692dd10>
```



```
<matplotlib.image.AxesImage at 0x7f287692dd10>
```



Data Pre-Processing

```python
x_train = x_train.reshape(60000, 28, 28, 1).astype('float32')
x_test = x_test.reshape(10000, 28, 28, 1).astype('float32')
```

```python
n_classes = 10
y_train = np_utils.to_categorical(y_train, n_classes)
y_test = np_utils.to_categorical(y_test, n_classes)
```

```python
y_train[0]
```

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

Create model

```python
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(n_classes, activation="softmax"))
```

Test the saved model

```
model=load_model("model.h5")
```

```
img = Image.open("/content/sample_data/sample.png").convert("L")
img = np.array(img).reshape(1, 28, 28, 1)

res  = model.predict(img)
res = np.argmax(res, axis = 1)
res = pd.Series(res, name="Label")
print(res)
```

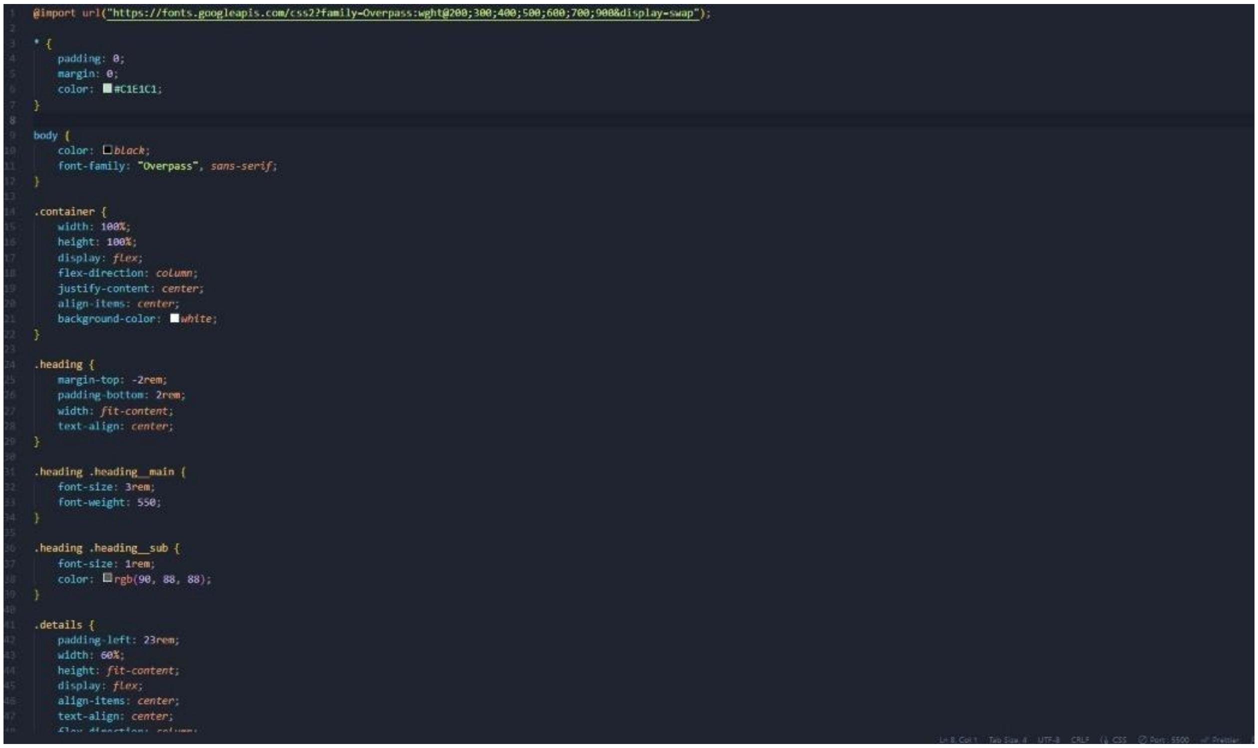
```
1/1 [==============================] - 0s 19ms/step
0    8
Name: Label, dtype: int64
```

HI’ML ÏILES

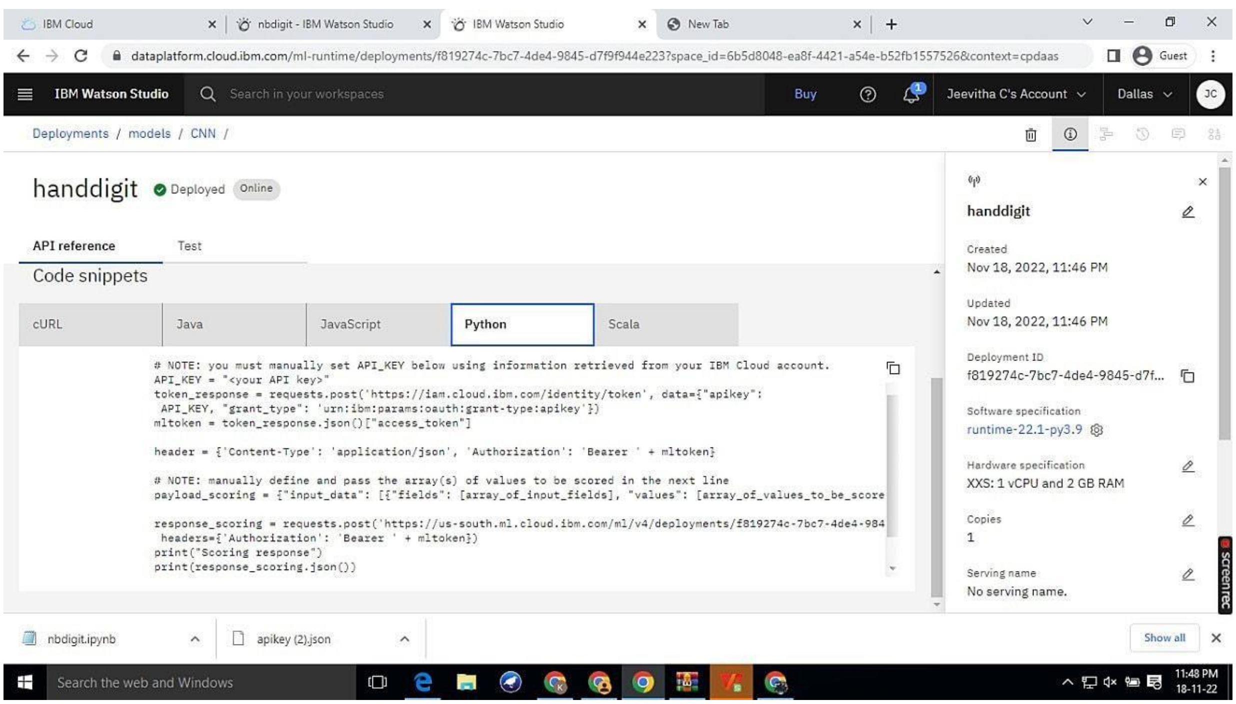
```html
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Handwritten Digit Recognition</title>
        <link rel="icon" type="image/svg" sizes="32x32" href="{{url_for('static',filename='images/icon.svg')}}" />
        <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
        <script src="https://unpkg.com/feather-icons"></script>
        <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
    </head>
    <body>
        <div class="container">
            <div class="heading">
                <h1 class="heading__main">Handwritten Digit Recognizer</h1>
                <h2 class="heading__sub">Easily analyze and detect handwritten digits</h2>
                <br>
                <br>
                <div class="details">
                    <h4>Team ID: PNT2022TMID02158</h4>
                    <h4>Student Names: Jeevitha C, Jeevamutharasi S, Kaushik A,
                        Krithic Kumar U B</h4>
                    <h4>
                        Project Description: Handwritten digit recognition has recently been of very
                        interest among the researchers because of the evolution of various Machine
                        Learning, Deep Learning and Computer Vision algorithms. This is highly useful in
                        finding the numbers that are written using hands. The illegible handwritten
                        digits are converted to their respective digits.
                    </h4>
                </div>
                <br>
                <br>
            </div>
            <div class="upload-container">
                <div class="form-wrapper">
                    <form class="upload" action="/predict" method="post" enctype="multipart/form-data">
                        <label id="label" for="upload-image"><i data-feather="file-plus"></i>Select File</label>
                        <input type="file" name="photo" id="upload-image" hidden />
                        <button type="submit" id="up_btn"></button>
                    </form>
                    <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
                </div>
            </div>
        </div>
    </body>
</html>
```

CSS ÏILES

```css
.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;
    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;
    }
}
```



ľESľ ľHE MODEL ON IBM CLOUD



GIľHUB PROÏILE

https://github.com/IBM-EPBL/IBM-Pɍoject-7452-1658857275

PROJECľ DEMO VIDEO LINK

https://dɍive.google.com/file/d/1_COaKs-uLwqjiPDuPa35znMKL1pmti-8/view?usp=dɍivesdk