CHAPTER 1
PREPARATION PHASE

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and Artificial Intelligence. With the use of Deep Learning and Machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

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

1.2 PURPOSE

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

1.3 PREREQUISITES:

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

Creation of github account and linked to the project repository **" A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM"**

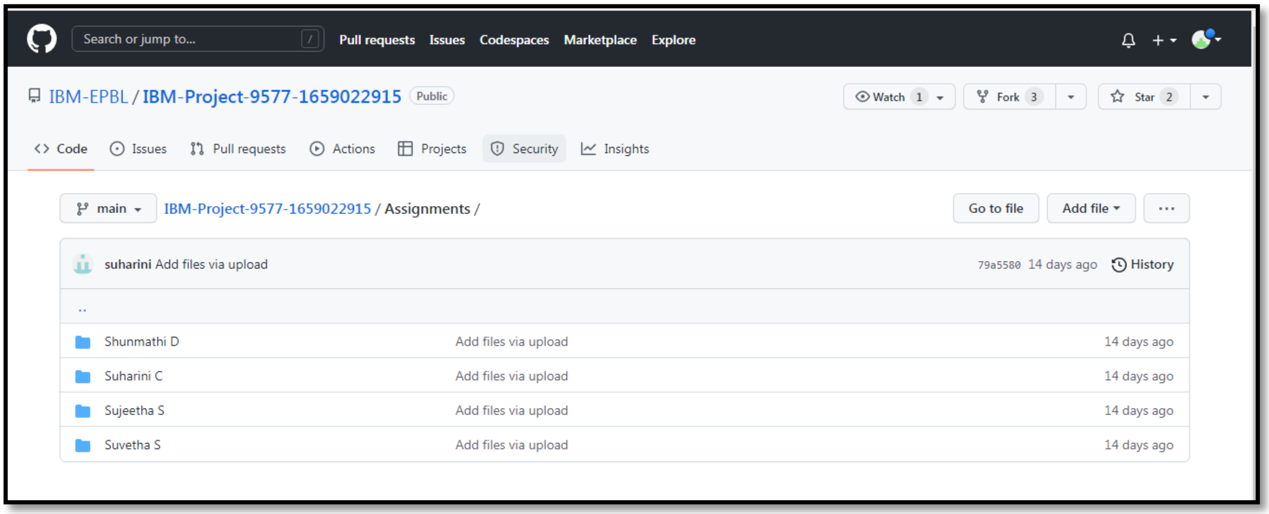


Fig 1.1 GitHub Account

CHAPTER 2

IDEATION PHASE

2.1 EXISTING PROBLEM

- The different architectures of CNN, hybrid CNN,CNN - RNN and CNNHMM models, and domain - specific recognition system, are not thoroughly inquired and evolutionary algorithms are not clearly explored for optimizing CNN learning parameters ,the number of layers, learning rate and kernel sizes of convolutional filters.
- The fluctuation of accuracies for handwritten digits was observed for 15 epochs by varying the hidden layers. There is no clear explanation given for observing variation in the overall classification accuracy by varying the number of hidden layers and batch size.

2.2 REFERENCES

[1] Rohan Sethi,ILa Kaushik(2020) .For, training supervised classification machine learning model the training dataset is fed to the classifier as input with labeled data and the model after successful training, becomes capable for the classification of handwritten digit. For classification, KNN supervised machine learning algorithm was used. And, for the computation of best fit supervisory signal to the input vector Euclidean distance formula was used. The advantages of the method is No Training Time for classification/regression. The disadvantages is With large data , the prediction stage might slow.

[2] Suthar, Anilkumar & Patel, Archanaben .This paper uses Blind De convolution technique and Iterative reweighted least square (IRLS) method and considers the

parameters like color and size. Image restoration techniques are being used to compare the original image's blurriness. For comparison two kinds of techniques such as blurring and non-blurring techniques are used. The advantages of the algorithm Does not require any complex filtering strategy to select salient edges .The disadvantages of This algorithm is not suitable for images having multiple objects .

[3] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, Hongdong Li. This paper proposes a new method which combines Two GAN models

1. Learning-to-Blur GAN (BGAN)

2. Learning-to-DeBlur GAN

There are two complementary processes. One module tries to mimic the real world blurry images by generating realistic photometric blurry images and the other module learns to recover the sharp images from these blurry images. A relativistic blur loss is introduced to measure the relative difference between real blur and synthesized blur. The developed relativistic blur loss estimates the probability that the given real-world blurry images are more realistic than the generated blurry images. This method produced more sharper results and shows higher PSNR value and good SSIM value which is almost equal to one.

[4] Jinshan Pan, Deqing Sun, Hanspeter Pfister, Ming-Hsuan Yang.This paper uses Dark Channel Prior for Image Deblurring and makes use of Kohler Dataset. Levin et al. showed that de blurring methods based on this prior tend to favour blurry images over original clear images, especially for algorithms formulated within the Maximum a Posteriori (MAP) framework. However, Natural image models do not generalize well to specific images, such as face, text, and low illumination images. But the proposed algorithm achieves state-of-the-art performance on widely-used

Natural Image De blurring benchmark and can also work on non-uniform deblurring. The advantages of the system is The proposed algorithm is easily extended to handle non-uniform blur and achieves state-of-the- art results on deblurring natural images.

[5] Pan, Ze & Lv, Qunbo & Tan, Zheng.A two-stage processing deblurring strategy combined with multi scale framework. A prior network is trained using relativistic GAN loss to generate sharper edges and fewer artifacts. This paper concentrates on PSNR, SSIM values .The advantages is Fewer artifacts. The disadvantages of the architecture is Processing time is a little higher than previous techniques.

2.3 PROBLEM STATEMENT

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 analyzed by the model and the detected result is returned on to UI(User Interface).

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

Fig 3.1 Empathy Map

3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

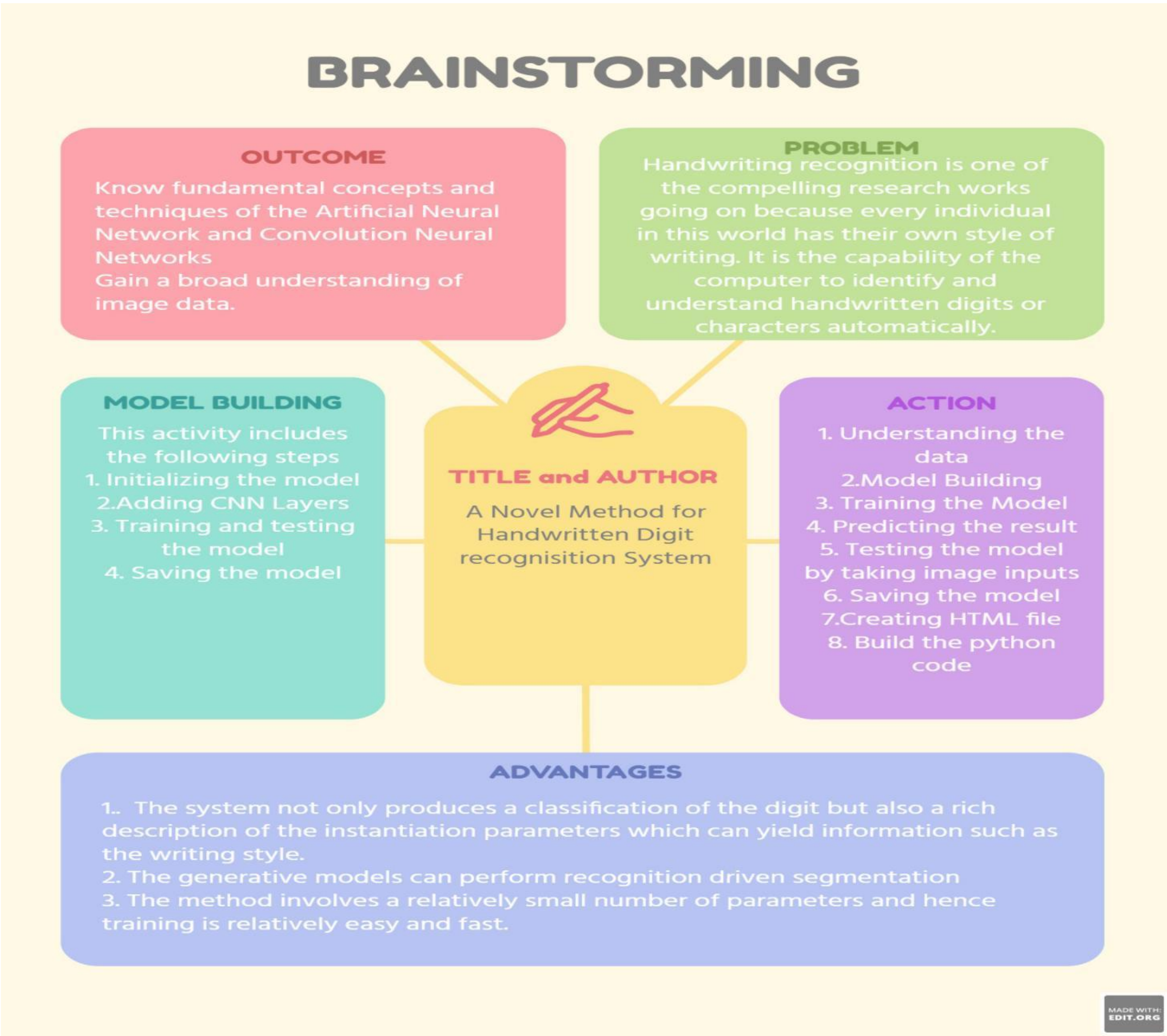


Fig 3.2 Brainstorming

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.

The main idea of the project is to,

- Initialise the model
- Adding CNN layers
- Training and testing the model
- Saving the model

3.3 PROPOSED SOLUTION

To create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which canrecognize the digits, it can be extended to letters and an individual’s handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	With the progress in the field of science and technology, everything is being digitized to reduce human effort. Hence there comes the need for handwritten digit recognition system in real time applications
2.	Idea / Solution description	To create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual's handwriting. The major goal of the proposed system is understanding CNN and applying it.
3.	Novelty / Uniqueness	Uses advanced digital techniques Compared to conventional techniques accuracy is high
4.	Social Impact / Customer Satisfaction	Has impact on physically impaired people and helps them in terms of safety
5.	Business Model (Revenue Model)	Cyber security applications
6.	Scalability of the Solution	Uses CNN techniques for improved scalability

TABLE 3.1 PROPOSED SOLUTION

3.4 PROBLEM SOLUTION FIT

This discusses the difficulties or challenges faced by people in recognizing handwritten digits. for example, while writing cheque and in order to overcome these isssues by customer this model is designed .

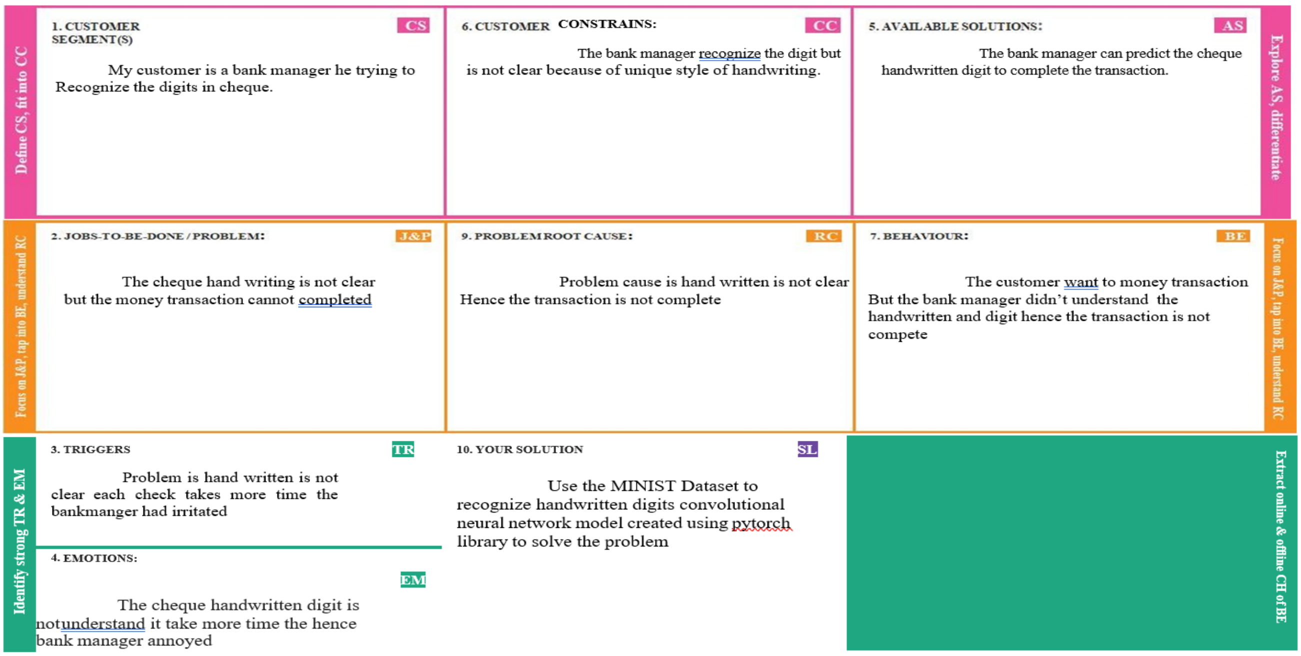


FIG 3.3 SOLUTION FIT

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of Artificial Intelligence, 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 typeof 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 and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described asa virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST 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.

4.2 NON FUCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	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	**Security**	1) The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. 2) The generative models are capable ofsegmentation driven by recognition. 3) The procedure uses a relatively.
NFR-3	**Reliability**	The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantityof training instances. Numerous techniques and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers.
NFR-4	**Accuracy**	With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

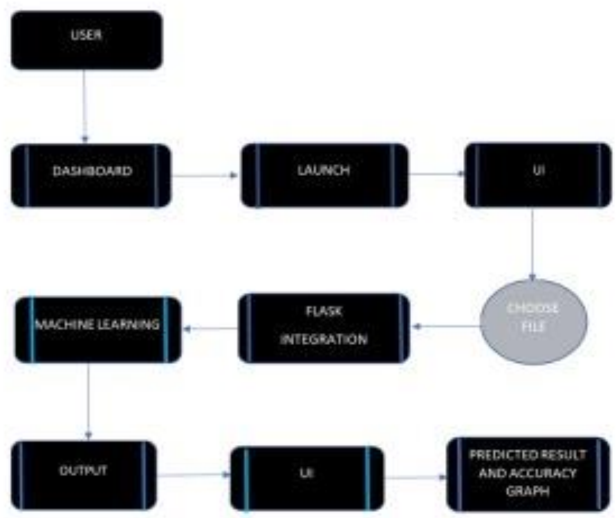


Fig 5.1 Data Flow Diagram

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

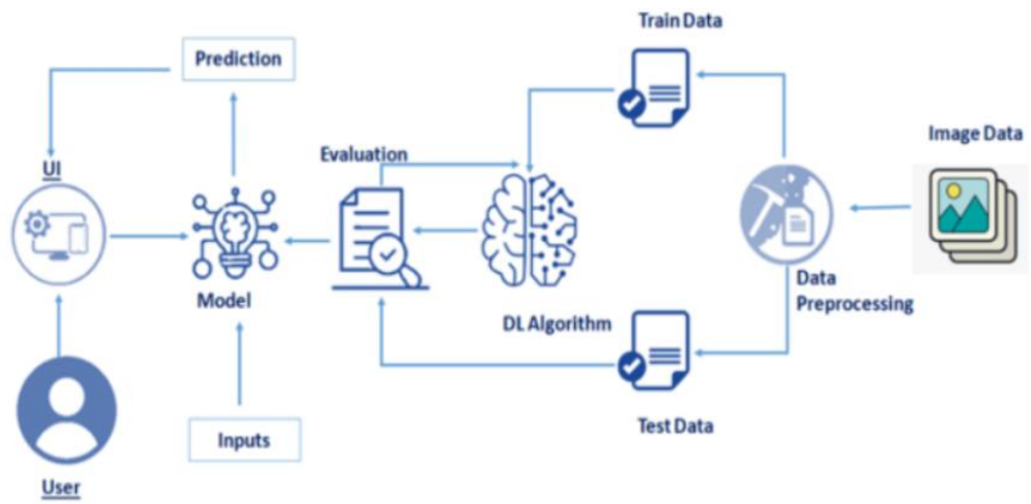


Fig 5.2 Block Diagram

The above Figure 5.2 illustrates the architecture diagram of the proposed system. The proposed model contains the four stages in order to classify and detect the digits:

A. Pre-processing

B. Segmentation

C. Feature Extraction

D. Classification and Recognition

5.2.2 PRE-PROCESSING:

The role of the pre-processing step is it performs various tasks on the input image. It basically upgrades the image by making it reasonable for segmentation. The fundamental motivation behind pre-processing is to take off a fascinating example from the background. For the most part, noise filtering, smoothing and standardization are to be done in this stage.

5.2.3 SEGMENTATION:

Once the pre-processing of the input images is completed, sub-images of individual digits are formed from the sequence of images. Pre-processed digit images are segmented into a sub-image of individual digits, which are assigned a number to each digit. Each individual digit is resized into pixels. In this step an edge detection technique is being used for segmentation of dataset images.

5.2.4 FEATURE EXTRACTION:

After the completion of pre-processing stage and segmentation stage, the pre-processed images are represented in the form of a matrix which contains pixels of the images that are of very large size. In this way it will be valuable to represent the digits in the images which contain the necessary information. This activity is called feature extraction. In the feature extraction stage redundancy from the data is removed.

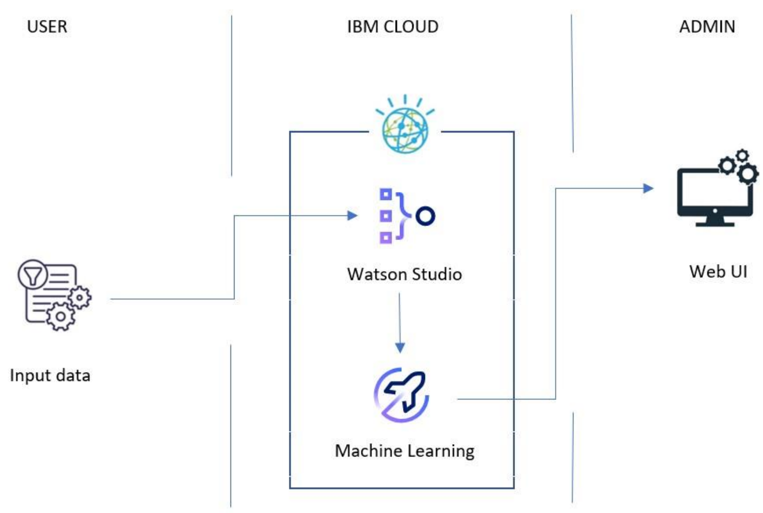
5.2.5 CLASSIFICATION AND RECOGNITION:

In the classification and recognition step the extracted feature vectors are taken as an individual input to each of the following classifiers. In order to showcase the working system model extracted features are combined and defined using following three classifiers:

- K-Nearest Neighbor
- Random Forest Classifier

The CNN model works in the following sequence. User uploads a particular image of any digit which he wants to recognize. The image will be processed by the system. On running the system code the output is generated that shows which is the digit uploaded by the user and also displays the accuracy rate predicted by the model. On uploading image with different resolutions other than the one mentioned in the code, the output generated shows error, and displays an error message to the user.

5.2 TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story / Task	Priority	Release
Customer (Mobile user)	Registration	As a user, I can register for the application by entering my email, password, and confirming my password.	High	Sprint-1
		As a user, I will receive confirmation email onceI have registered for the application	High	Sprint-1
		As a user, I can register for the application through Facebook	Low	Sprint-2
		As a user, I can register for the application through Gmail	Medium	Sprint-2
	Login	As a user, I can log into the application by entering email & password	High	Sprint-1
	Home	As a user, I can view the application’s home page where I can read the instructions to usethis application	Low	Sprint-1
	Upload Image	As a user, I can able to input the images ofdigital documents to the application	High	Sprint-3
	Predict	As a user I can able to get the recognised digitas output from the images of digital documents or images	High	Sprint-3
		As a user, I will train and test the input to get the maximum accuracy of output.	Medium	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

This is nothing but an organized plan to complete the activity and check ourselves whether the project reached its goal or not.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
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	Suharini
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Shunmathi, Sujeetha
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	Shunmathi ,Sujeetha
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	Suharini, Suvetha
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Suharini, Suvetha
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Suvetha
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	Suharini, Suvetha, shunamthi sujeetha
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	Suharini, Suvetha, shunamthi sujeetha

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	Shunmathi, Sujeetha
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	Suharini, Suvetha
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	Suharini, Suvetha, shunamthi sujeetha
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Suharini, Suvetha, shunamthi sujeetha

TABLE 6.1 SPRINT PLANNER

6.2 SPRINT DELIVERY SCHEDULE

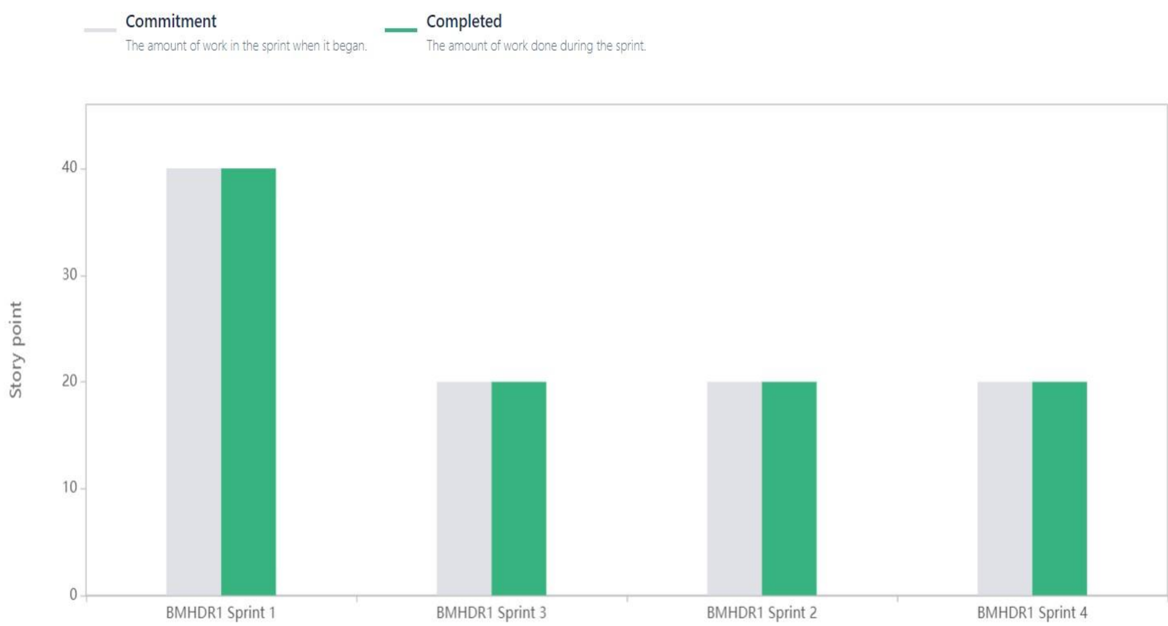
The main agenda of Sprint planning is to define the scope of delivery and how to accomplish the work. It sets up a common goal for the team, and everyone's focus is to achieve that goal during the Sprint.

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

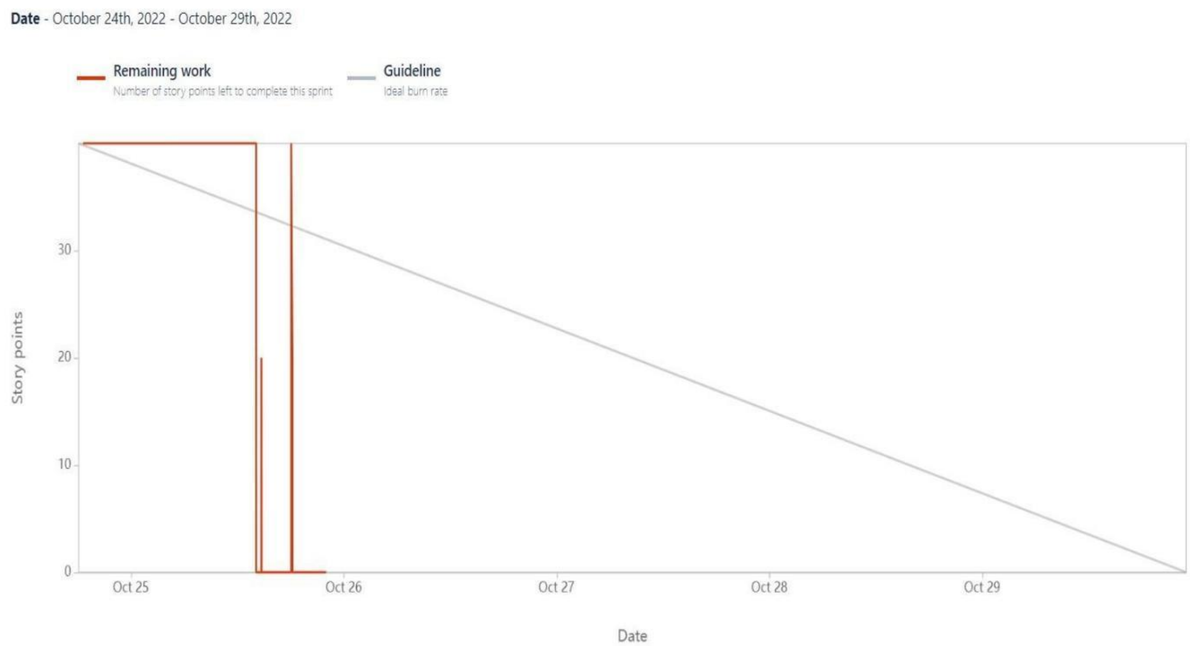
TABLE 6.2

6.3 REPORTS FROM JIRA

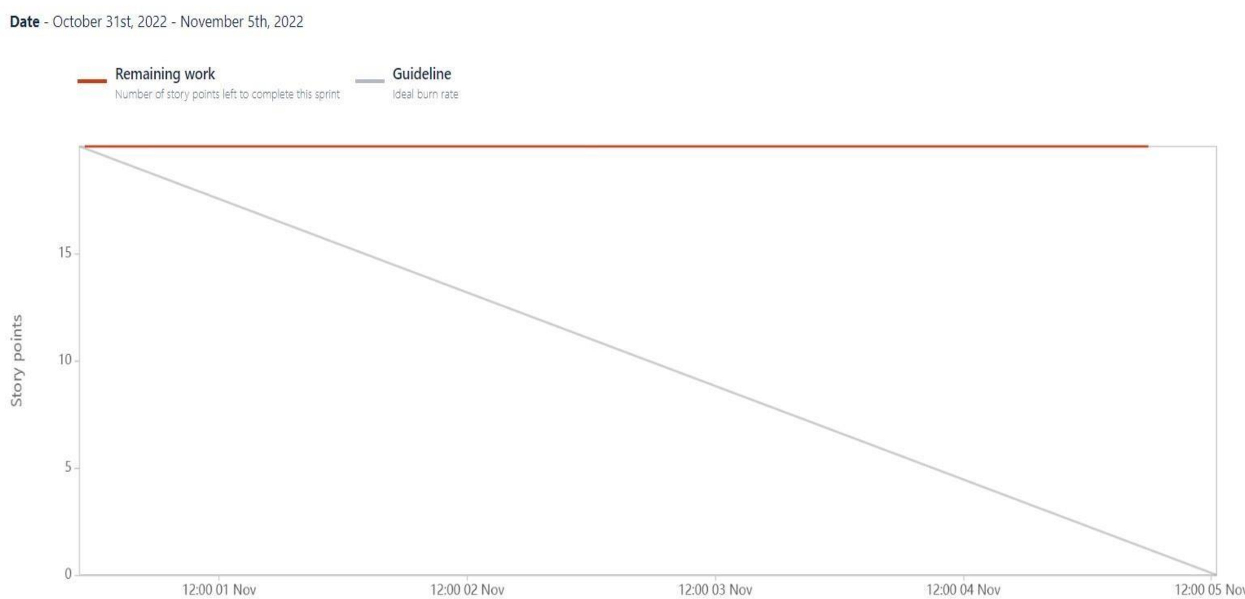
6.3.1 Velocity Report



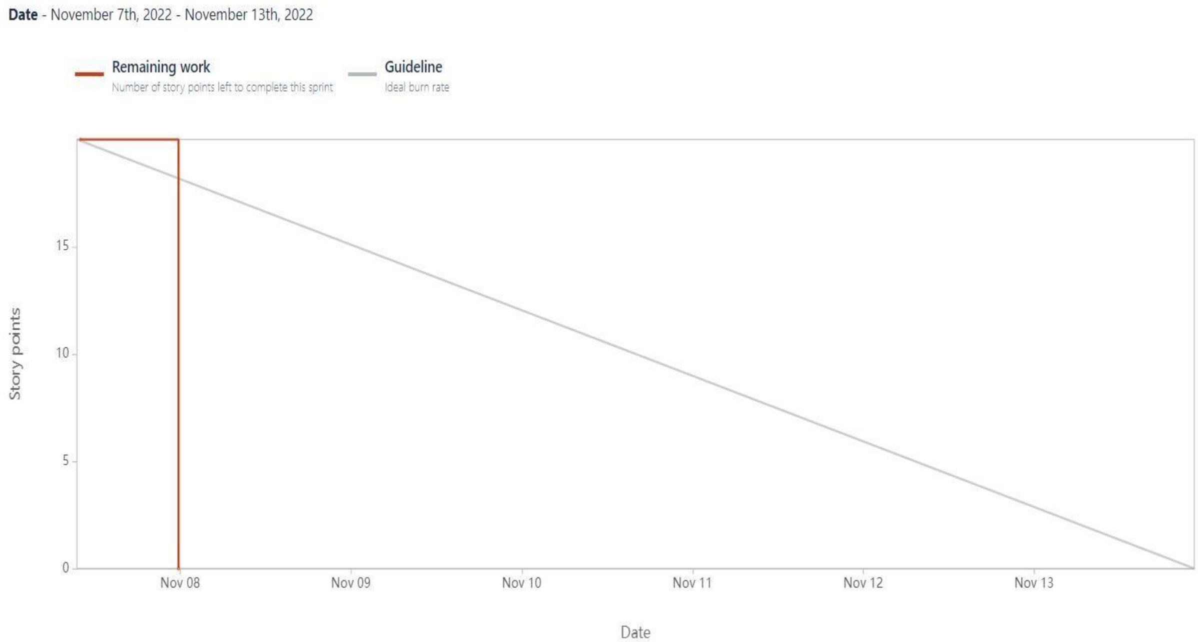
6.3.2 Sprint 1



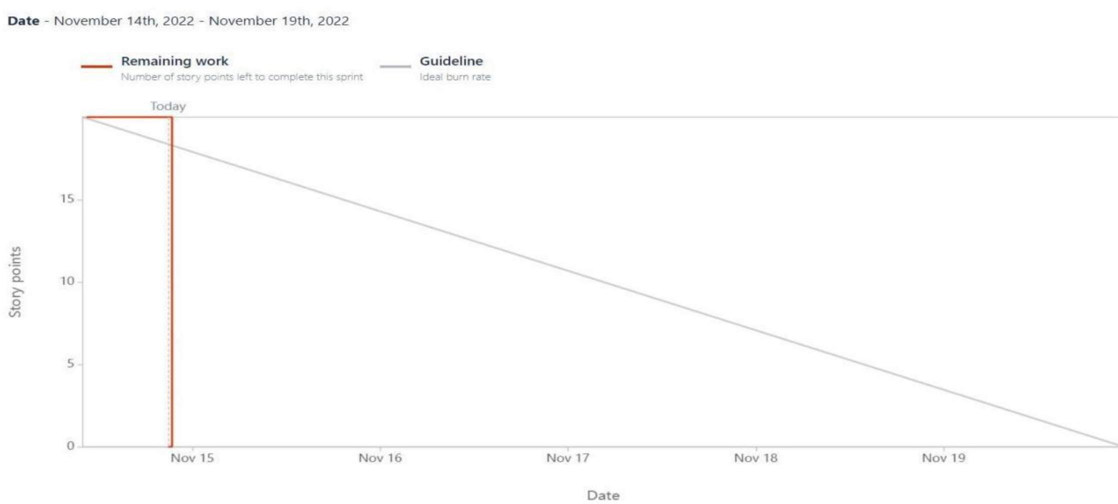
6.3.3 Sprint 2



6.3.4 Sprint 3

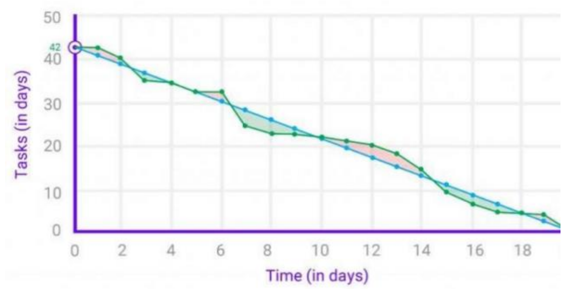


6.3.5 Sprint 4



6.3.4 BURNDOWN CHART

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



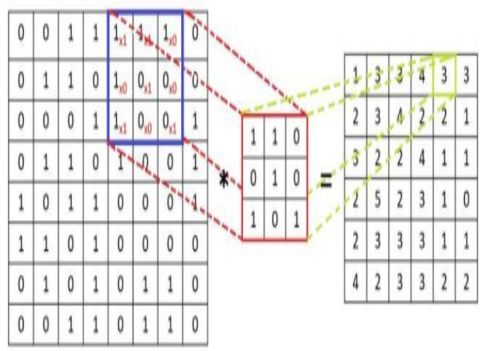
CHAPTER 7

CODING AND SOLUTIONING

7.1 FEATURE 1(CONVOLUTION LAYER)

Convolution Layers

The kernels/filters ($K(i,j)$) convolve with the input image ($I(i,j)$) to produce the feature map ($F(i,j)$)



CODE:

```
#adding modeL Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
```

7.2 FEATURE 2(The Fully Connected Layer)

The Fully Connected (FC) layer comprises the weights and biases together with the neurons and is used to connect the neurons between two separate layers. Fully Connected Layer (also known as Hidden Layer) is the last layer in the convolutional neural network. This layer is a combination of Affine function and Non-Linear function.

Affine Function **y = Wx + b,** Non-Linear Function **Sigmoid, TanH and ReLu**

CODE:

```
[ ] #flatten the dimension of the image
    model.add(Flatten())
```

7.3 FEATURE 3(Pooling Layer)

The Pooling layer is responsible for the reduction of the size(spatial) of the Convolved Feature. This decrease in the computing power is being required to process the data by a significant reduction in the dimensions.

1.AveragePooling

2 Max pooling.

A Pooling Layer is usually applied after a Convolutional Layer. This layer's major goal is to lower the size of the convolved feature map to reduce computational expenses. This is accomplished by reducing the connections between layers and operating independently on each feature map. There are numerous sorts of Pooling operations, depending on the mechanism utilised. The largest element is obtained from the feature map in Max Pooling. The average of the elements in a predefined sized Image segment is calculated using Average Pooling. Sum Pooling calculates the total sum of the components in the predefined section. The Pooling Layer is typically used to connect the Convolutional Layer and the FC Layer.

CODE:

```
[ ] #output layer with 10 neurons
    model.add(Dense(number_of_classes,activation = 'softmax'))
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Component	Test Scenario	Expected Result	Actual Result	Status
Homepage_TC_OO1	Home Page	Verify user is able to see the Homepage when clicked on the link	Home Page should be displayed.	Working as expected	Pass
Homepage_TC_OO2	Home Page	Verify the UI elements in Homepage	Application should show below UI elements: a.choose file button b.predict button c.clear button	Working as expected	Pass
Homepage_TC_OO4	Home page	Verify user able to select invalid file format	Application won't allow to attach formats other than ".png, .jiff, .pjp, .jpeg, .jpg, .pjpeg"	Working as expected	Pass
Predict_TC_OO5	Predict page	Verify user is able to navigate to the predict to and view the predicted result	User must be navigated to the predict page and must view the predicted result	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	0	0	0	0
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	0	0	0	0
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	0	0	0	0	0

8.2.2 **Test Case Analysis**

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5
Security	5	0	0	5
Final Report Output	5	0	0	5
Performance	5	0	0	5

CHAPTER 9

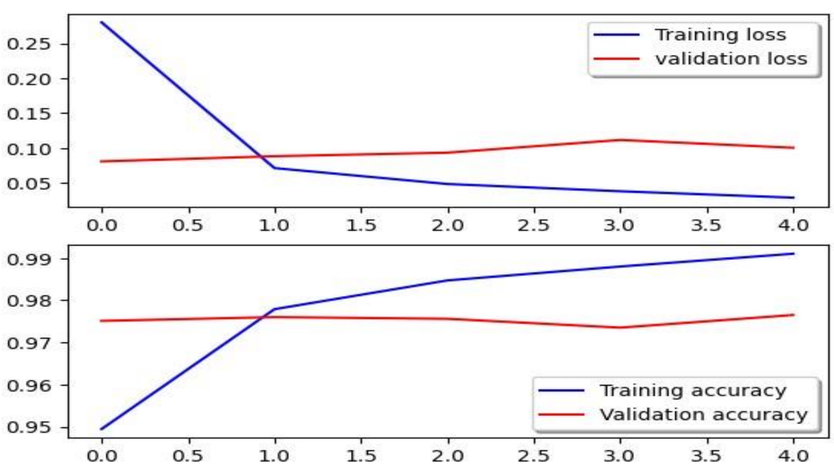
RESULTS

9.1 Performance Metrics

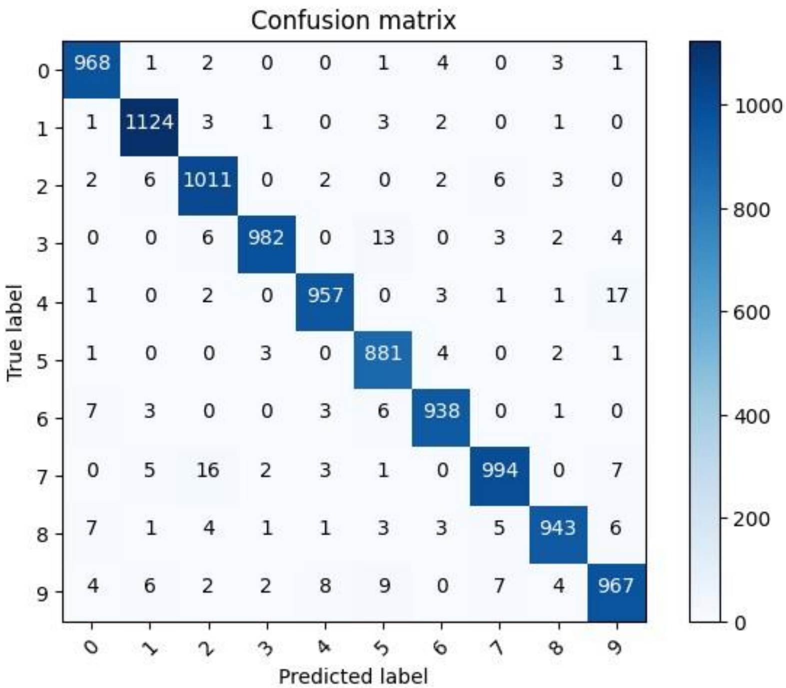
Model 9.1.1 Summary:

```
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
_________________________________________________________________
None
```

9.1.2 Accuracy:



9.1.3 Confusion Matrix:



9.1.4 Classification Report:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.98	0.99	0.99	1135
2	0.97	0.98	0.97	1032
3	0.99	0.97	0.98	1010
4	0.98	0.97	0.98	982
5	0.96	0.99	0.97	892
6	0.98	0.98	0.98	958
7	0.98	0.97	0.97	1028
8	0.98	0.97	0.98	974
9	0.96	0.96	0.96	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 Performance Metrics Result:

Locust Test Report

During: 11/15/2022, 10:52:19 PM - 11/15/2022, 10:56:36 PM

Target Host: http://127.0.0.1:5000/

Script: locustfile.py

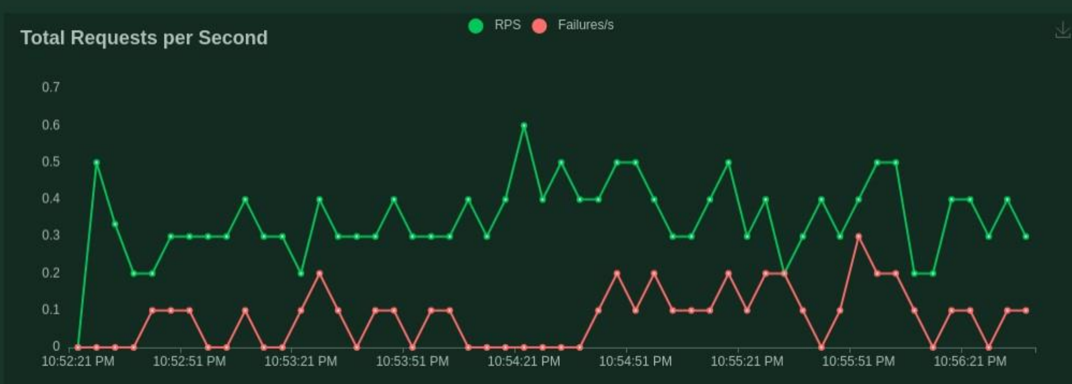
Request Statistics

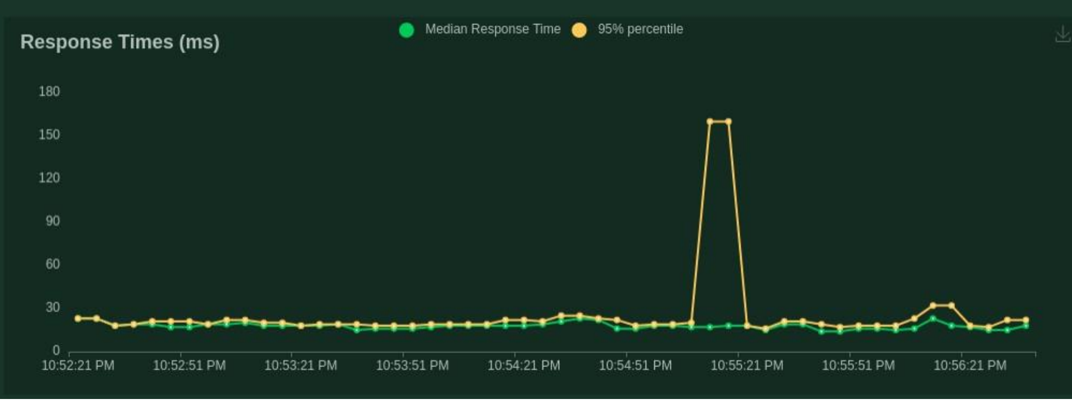
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	67	0	17	12	24	5875	0.3	0.0
GET	//predict	23	23	21	11	163	265	0.1	0.1
	Aggregated	**90**	**23**	**18**	**11**	**163**	**4441**	**0.4**	**0.1**

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	//	18	18	19	19	22	23	25	25
GET	//predict	15	15	16	16	17	32	160	160
	Aggregated	**17**	**18**	**18**	**19**	**22**	**23**	**160**	**160**

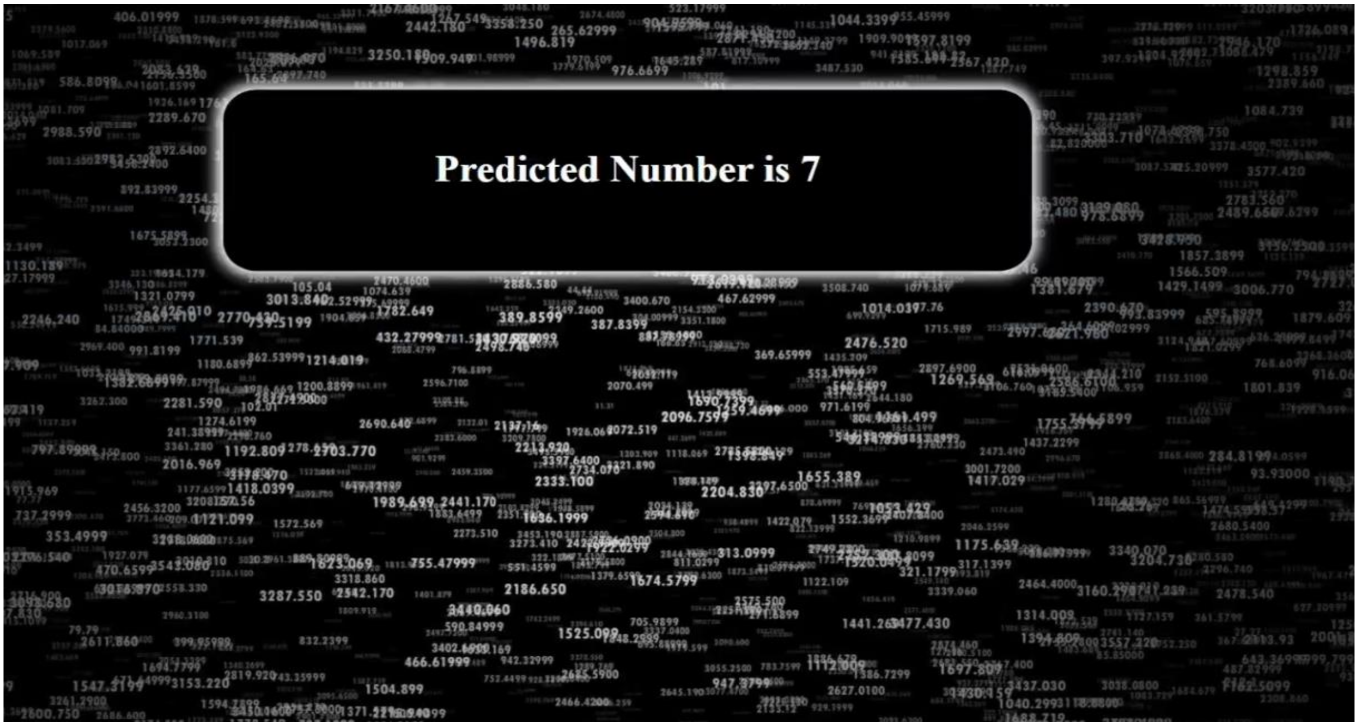
Charts





9.2 OUTPUT


HandWritten Digit Recognition System
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.
Choose File
Predict
Clear


Predicted Number is 7

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

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 styleThe task of handwritten digit recognition, using a classifier, has great importance and use such as – online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example - tax forms) and so on. Handwritten digit recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry.

The disadvantage of handwriting recognition technologies is that not everyone's handwriting is the same, everyone writes differently. This starts the problem in the handwriting recognition technology when it need to translate a person's handwriting into type and because of this problem many companies failed to perform well because many couldn't effectively use the program well enough

CHAPTER 11

CONCLUSION

This project explores the Machine Learning algorithms of convolutional neural network. MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers. The variations of accuraciesfor handwritten digit were observed for 5 epochs by varying the hidden layers. A successful model is developed which can generate output that is a recognises the handwritten input images based on probabilities generated in the decoder functions.Many classifiers like KNN, SVM, CNN are used to identify the digit from the handwritten image. as per the review, CNN is providing better performance than others. Hand written digit recognition system can be extended to a recognition systemthat can also able to recognize handwritten character and handwritten symbols. Future studies might consider on hardware implementation of recognition system.

CHAPTER 12

FUTURE SCOPE

This project explores the Machine Learning algorithms of convolutional neural network. MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers. The variations of accuracies for handwritten digit were observed for 5 epochs by varying the hidden layers. A successful model is developed which can generate output that recognizes the handwritten input images based on probabilities generated in the decoder functions. Many classifiers like KNN, SVM, CNN are used to identify the digit from the handwritten image. as per the review, CNN is providing better performance than others. Handwritten digit recognition system can be extended to a recognition system that can also able to recognize handwritten character and handwritten symbols. Future studies might consider on hardware implementation of recognition system.

CHAPTER 13

APPENDIX 1

SOURCE CODE

```
import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out
computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular
deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt   #used for data visualization

(x_train, y_train), (x_test, y_test)=mnist.load_data ()
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  #storing the no of classes in a variable
y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in
binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

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'))
#flatten the dimension of the image
model.add(Flatten())
#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling the model

```
#Compile model
model.compile(loss= 'categorical_crossentropy', optimizer="Adam",
metrics=['accuracy'])
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

Train the model

```
#fit the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

Observing the metrics

```
# Final evaluation of the model
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
prediction=model.predict(x_test[6000:6001])
print(prediction)
plt.imshow(x_test[5100])
import numpy as np
```

print(np.argmax(prediction, axis=1)) *#printing our Labels from first 4 images*

np.argmax(y_test[5100:5101]) *#printing the actual labels*

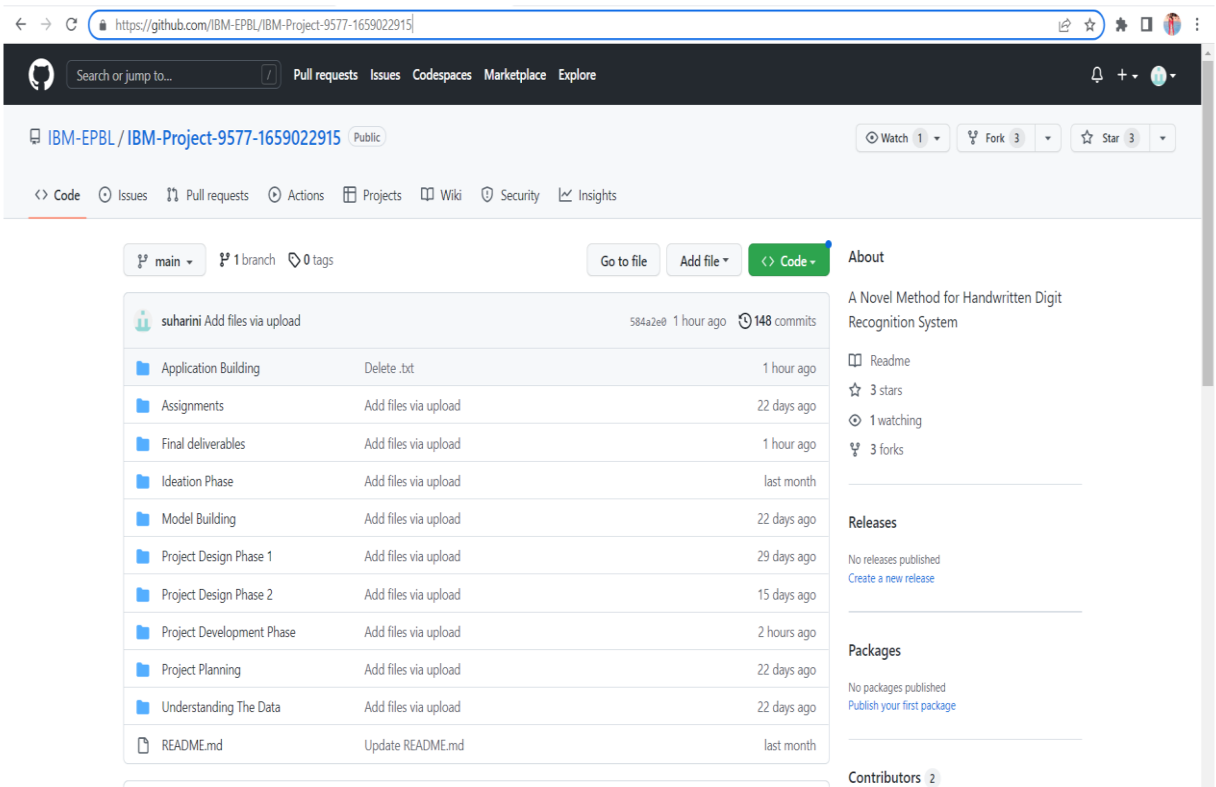
FLASK CODE:

```
mport 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 = 'D:/ibm/data'


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
 model =
load_model("./DigitRecog_IBM_model/mnistCNN.h5")

@app.route('/
') def
index():
    return
render_template('index.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
```

```
return render_template('predict.html', num=str(num[0]))
if __name__ == '__main__':      app.run(debug=True, threaded=False)
```

APPENDIX II

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-9577-1659022915



DEMO VIDEO LINK: https://drive.google.com/file/d/16yysPTUB4rILNhh41_hbaCOcUaayojd/view?usp=sharing