A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

TEAM ID	PNT2022TMID46690
PROJECT NAME	A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM
TEAM MEMBERS	J. FASILA AFREEN R. AARTHI T. RETHINAPRIYA A. SURYA

A PROJECT REPORT

CONTENTS

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits. We have taken this a step further where our handwritten digit recognition system not only detects scanned images of handwritten digits but also allows writing digits on the screen with the help of an integrated GUI for recognition.

1.2 PURPOSE

- The issue of transcribed digit acknowledgment has for some time been an open issue in the field of example order. A few examined have demonstrated that Neural Network has an incredible execution in information arrangement. The fundamental target of this paper is to give effective and solid procedures to acknowledgment of transcribed numerical by looking at different existing arrangement models. This paper thinks about the exhibition of Convolutional Neural Network (CCN). Results demonstrate that CNN classifier beat over Neural Network with critical improved computational effectiveness without relinquishing execution. Handwritten digit recognition can be performed using the Convolutional neural network from Machine Learning.

- Using the MNIST (Modified National Institute of Standards and Technologies) database and compiling with the CNN gives the basic structure of my project development. So, basically to perform the model we need some libraries such as, 'Pandas', Tensor Flow. These are the main structure on which my main project stands. MNIST data contains about 70,000 images of handwritten digits from 0-9. So, it is a class 10 classification model. This dataset is divided into 2 parts i.e. Training and Test dataset. Image representation as 28*28 matrix where each cell contains grayscale pixel value.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. Handwritten Digit Recognition System involves reception and interpretation of handwritten digits by a machine. Due to variation in shape and orientation of handwritten digits, it is difficult for a machine to interpret handwritten digits. Handwritten digit Recognition has a wide area of research due to its vast applications like automatic bank processing, billing and automatic postal service. In this thesis, an Offline Handwritten Digit Recognition, first part is feature extraction from handwritten images and the second one is classification of feature vector into digits.

   We propose descriptors for handwritten digit recognition based on Histogram of Oriented Gradient (HOG) feature. It is one of the widely used feature vector for object detection in computer vision.

For classification of features, linear Proximal Support Vector Machine Classifier is proposed. This is a binary class classifier which is further converted to a 10-class classifier by means of One against all algorithm. Due to small training time, PSVM classifier is preferable over standard Support Vector Machine (SVM) Classifier. The handwritten images both for training and testing are taken from MNIST database. The performance of the system is measured in terms of Sensitivity, Accuracy, Positive Predictively and Specificity.

2. The handwritten digit recognition problem becomes one of the most notorious problems in machine" "literacy and computer vision operations. numerous machine literacy ways have been employed to break the handwritten number recognition problem. This paper focuses on Neural Network (NN) approaches. The three most" "popular NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural "network (CNN). In this paper, the three NN approaches are compared and estimated in terms of numerous factors" "similar as delicacy and performance. Recognition delicacy rate and performance, still, isn't the only criterion in the evaluation process, but there are intriguing criteria similar as prosecution time. Random and

   standard dataset of handwritten number have been used for conducting the trials. The results show that among the three NN approaches, DNN is the most accurate algorithm; it has98.08 delicacy rate. still, the prosecution time of DNN is similar with the "other two algorithms.

3. Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. This is a system widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed. Bayesian Decision, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition uses Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

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

5. 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. In this report, we compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolution neural networks and with

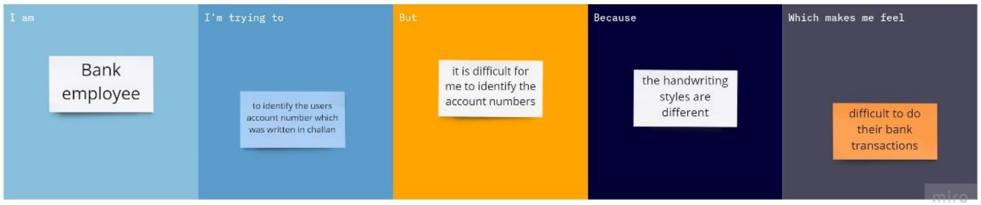
Deep Learning algorithm like multilayer CNN using Keras with Theano and TensorFlow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consists of 60,000 training images and 10,000 test images. The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field. For example, Convolution Neural networks with back propagation for image processing. The applications where these handwritten digits recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.

2.2 REFERENCES

- ➢ https://www.ijnrd.org/papers/IJNRD1704024.pdf
- ➢ https://www.irjet.net/archives/V9/i6/IRJET-V9I6208.pdf
- ➢ https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.228.158&rep=rep1&type=pdf
- ➢ http://ijcsit.com/docs/Volume%207/vol7issue1/ijcsit2016070101.pdf
- ➢ http://troindia.in/journal/ijcesr/vol6iss6part2/32-36.pdf

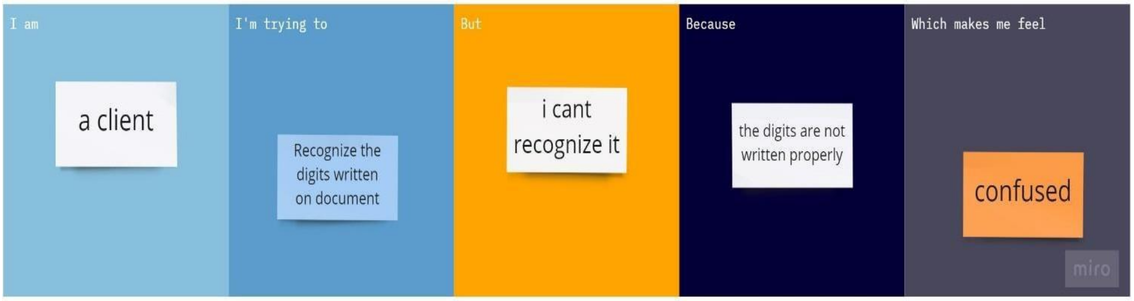
2.3 PROBLEM STATEMENT DEFINITION

Problem Statement 1



I am	I'm trying to	But	Because	Which makes me feel
Bank employee	to identify the users account number which was written in challan	it is difficult for me to identify the account numbers	the handwriting styles are different	difficult to do their bank transactions

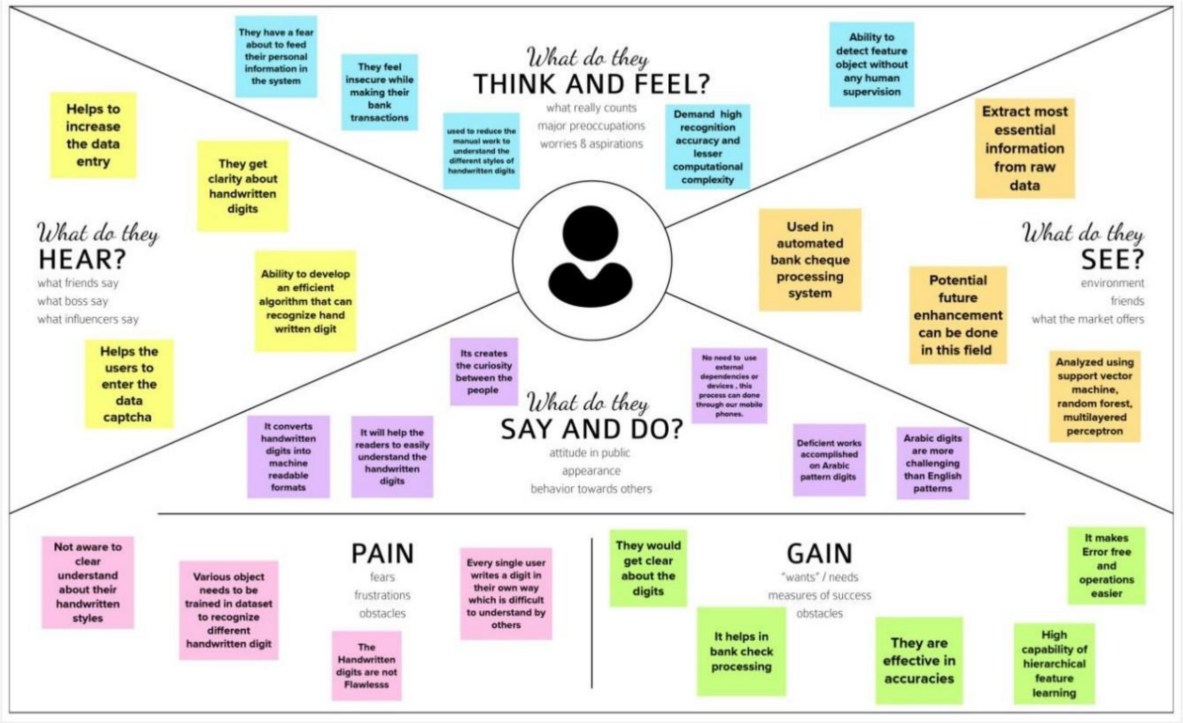
Problem Statement 2



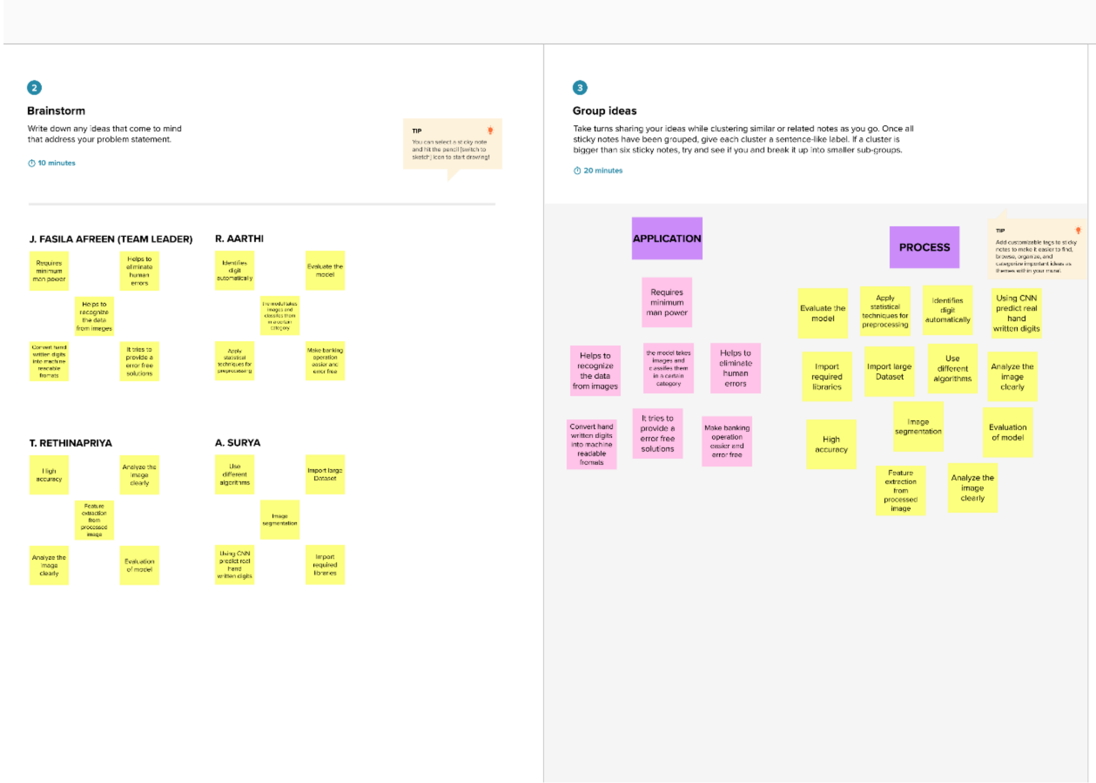
I am	I'm trying to	But	Because	Which makes me feel
a client	Recognize the digits written on document	i cant recognize it	the digits are not written properly	confused

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATY MAP CANVAS


What do they THINK AND FEEL?
what really counts
major preoccupations
worries 8 aspirations
They have a fear about to feed their personal information in the system
They feel insecure while making their bank transactions
used to reduce the manual work to understand the different styles of handwritten digits
Demand high recognition accuracy and lesser computational complexity
Ability to detect feature object without any human supervision
What do they HEAR?
what friends say
what boss say
what influencers say
Helps to increase the data entry
They get clarity about handwritten digits
Ability to develop an efficient algorithm that can recognize hand written digit
Helps the users to enter the data captcha
What do they SEE?
environment
friends
what the market offers
Extract most essential information from raw data
Used in automated bank cheque processing system
Potential future enhancement can be done in this field
Analyzed using support vector machine, random forest, multilayered perceptron
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Its creates the curiosity between the people
No need to use external dependencies or devices , this process can done through our mobile phones.
It converts handwritten digits into machine readable formats
It will help the readers to easily understand the handwritten digits
Deficient works accomplished on Arabic pattern digits
Arabic digits are more challenging than English patterns
PAIN
fears
frustrations
obstacles
Not aware to clear understand about their handwritten styles
Various object needs to be trained in dataset to recognize different handwritten digit
The Handwritten digits are not Flawlesss
Every single user writes a digit in their own way which is difficult to understand by others
GAIN
"wants" / needs
measures of success
obstacles
They would get clear about the digits
It helps in bank check processing
They are effective in accuracies
High capability of hierarchical feature learning
It makes Error free and operations easier

3.2 IDEATION & BRAINSTROMING



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

J. FASILA AFREEN (TEAM LEADER)
Requires minimum man power
Helps to eliminate human errors
Helps to recognize the data from images
Convert hand written digits into machine readable fromats
It tries to provide a error free solutions
R. AARTHI
Identifies digit automatically
Evaluate the model
The model takes images and classifies them in a certain category
Apply statistical techniques for preprocessing
Make banking operation easier and error free
T. RETHINAPRIYA
High accuracy
Analyze the image clearly
Feature extraction from processed image
Analyze the image clearly
Evaluation of model
A. SURYA
Use different algorithms
Import large Dataset
Image segmentation
Using CNN predict real hand written digits
Import required libraries

3

Group ideas

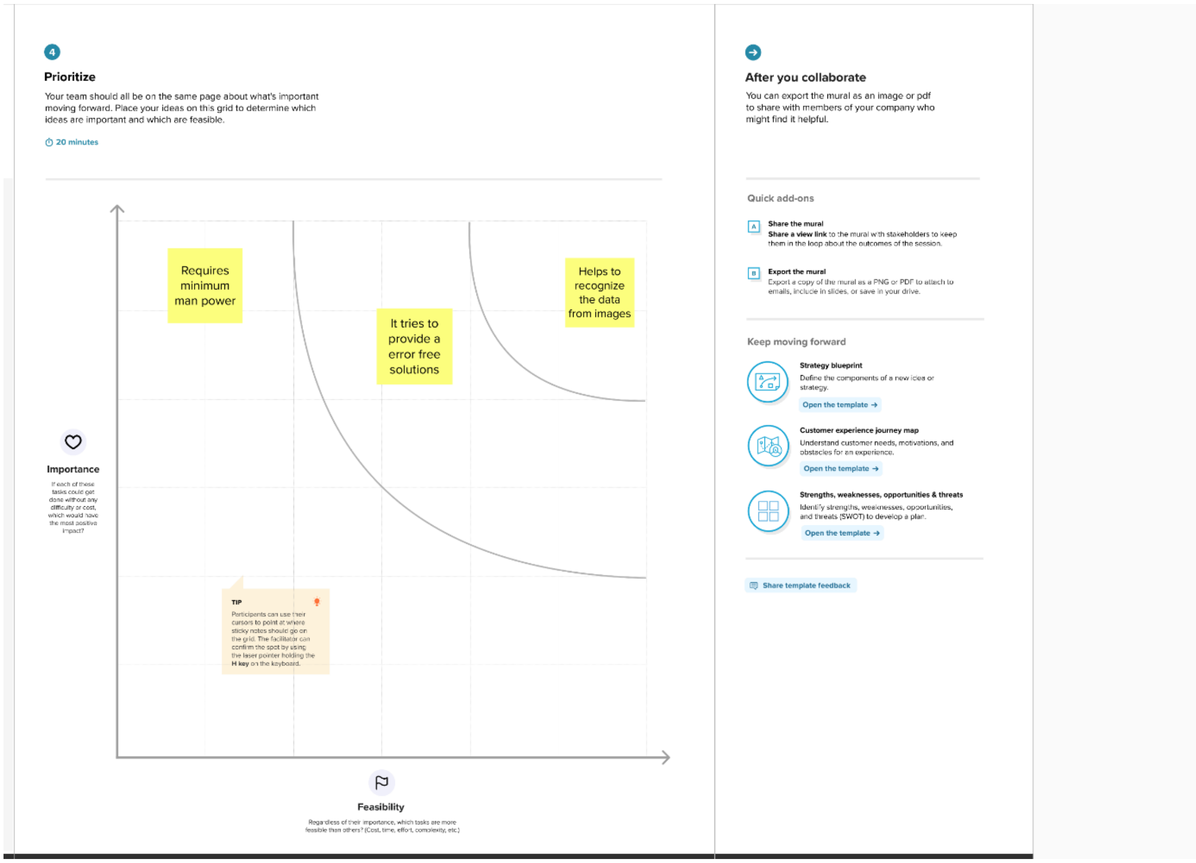
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

APPLICATION
Requires minimum man power
Helps to recognize the data from images
the model takes images and classifies them in a certain category
Helps to eliminate human errors
Convert hand written digits into machine readable fromats
It tries to provide a error free solutions
Make banking operation easier and error free
PROCESS
Evaluate the model
Apply statistical techniques for preprocessing
Identifies digit automatically
Using CNN predict real hand written digits
Import required libraries
Import large Dataset
Use different algorithms
Analyze the image clearly
High accuracy
Image segmentation
Evaluation of model
Feature extraction from processed image
Analyze the image clearly



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Requires minimum man power
It tries to provide a error free solutions
Helps to recognize the data from images

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

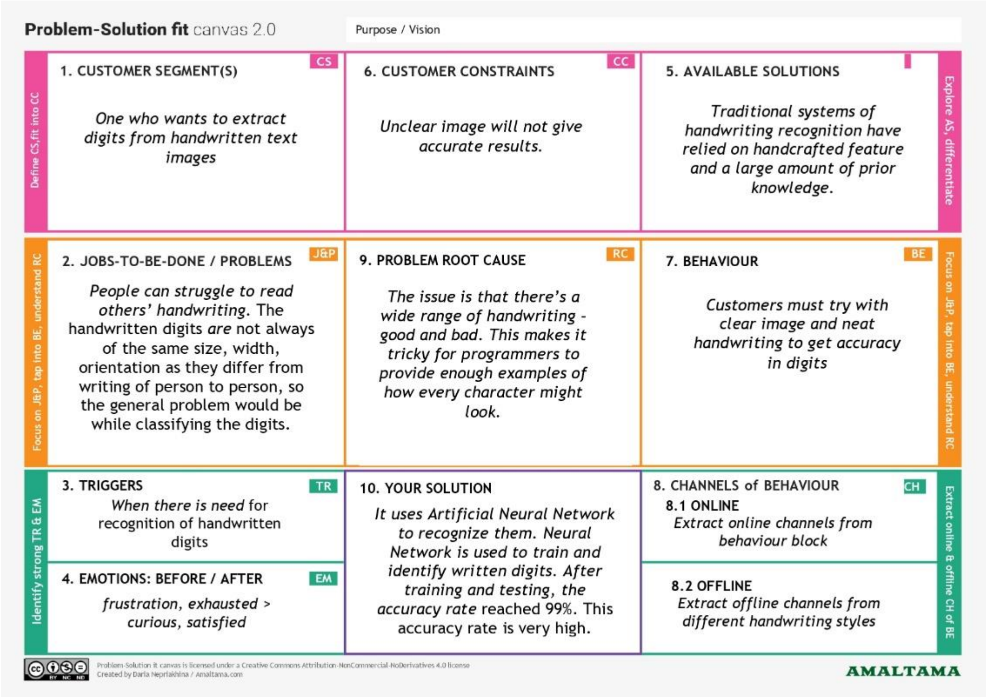
Share template feedback

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• The handwritten digit recognition is the capability of computer applications to recognize human handwritten digits. • It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. • The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. • In this competition, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn what works well and how techniques compare.
2.	Idea / Solution description	The proposed solution is to classify the digits which are in handwritten format by using CNN based model and this model can be trained by using the MNIST database which contains 60,000 training samples and 10,000 test samples.
3.	Novelty / Uniqueness	To classify the image datasets by using CNN, which provides an efficient solution compared to other methods. Here ANN algorithm is used for voice recognition which helps blind people.
4.	Social Impact / Customer Satisfaction	Users no need to use external dependencies or devices to recognize the digits, this process can be done through our mobile phones.
5.	Business Model (Revenue Model)	• The applications where these handwritten digit recognition can be used are the Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind people by using sound output. • Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc
6.	Scalability of the Solution	One of the approaches to make the handwritten digit recognition system scalable is to make use of cloud-native methods. For example, one of the cloud solutions for making AI scalable is IBM Cloud. IBM Cloud Build helps run and manage AI models and optimize decisions at scale across any cloud. The advantage of using the cloud to make solutions scalable is that we can deploy our AI application on the specific cloud environment that best supports our business needs. We can take advantage of built-in security capabilities and AI model monitoring. We can Automate AI lifecycles with ModelOps pipelines, deploy and run models through one-click integration and also prepare and build models visually and

		programmatically. Looking at these advantages, we can drive better business outcomes by optimizing our decisions and also make our solution scalable using cloud

3.4 PROBLEM SOLUTION FIT



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

One who wants to extract digits from handwritten text images

6. CUSTOMER CONSTRAINTS CC

Unclear image will not give accurate results.

5. AVAILABLE SOLUTIONS

Traditional systems of handwriting recognition have relied on handcrafted feature and a large amount of prior knowledge.

Explore AS, differentiate

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

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

People can struggle to read others' handwriting. The handwritten digits *are* not always of the same size, width, orientation as they differ from writing of person to person, so the general problem would be while classifying the digits.

9. PROBLEM ROOT CAUSE RC

The issue is that there's a wide range of handwriting - good and bad. This makes it tricky for programmers to provide enough examples of how every character might look.

7. BEHAVIOUR BE

Customers must try with clear image and neat handwriting to get accuracy in digits

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

Identify strong TR & EM

3. TRIGGERS TR

When there is need for recognition of handwritten digits

4. EMOTIONS: BEFORE / AFTER EM

frustration, exhausted > curious, satisfied

10. YOUR SOLUTION

It uses Artificial Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Extract online channels from behaviour block

8.2 OFFLINE

Extract offline channels from different handwriting styles

Extract online & offline CH of BE

Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license
Created by Daria Nepriakhina / Amaltama.com

AMALTAMA

4. REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	The product essentially converts handwritten digits to digital form.	The user is first asked to draw a number on the canvas, and the model that is built is then utilized to compare the data and provide an output in digitalized form.
FR-2	Recognizing the handwritten digit and displaying	Recognizing the handwritten digit and displaying.
FR-3	Import the dataset file directly to the program from a command that will download the dataset from its website. Save the dataset file in the same directory as the program	Installing packages and applications
FR-4	Build a Neural Network with a number of nodes in the input layer equal to the number of pixels in the arrays	Nil
FR-5	Activating the Neural Network	Packages – TensorFlow

4.2 NON FUNCTIONAL REQUIREMENTS

Non-functional Requirements:

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

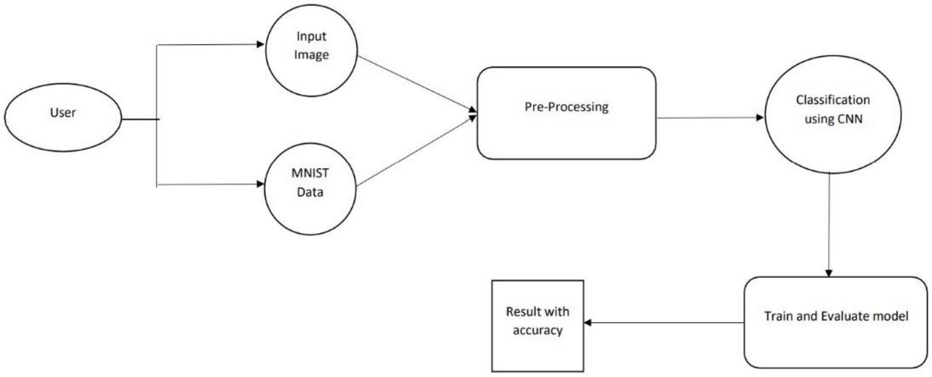
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	System design should be easily understood and user-friendly to users. Furthermore, users of all skill levels should be able to navigate it without problems.
NFR-2	**Security**	The system should automatically be able to authenticate all users with their unique usernames and password
NFR-3	**Reliability**	Should consistently perform according to its specifications.

NFR-4	**Performance**	Should reduce the delay in information when hundreds of requests are given.
NFR-5	**Availability**	Information is restricted to each user with limited access
NFR-6	**Scalability**	The system should be able to handle 10000 users accessing the site at the same time

5. PROJECT DESIGN

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat an amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the i stored.

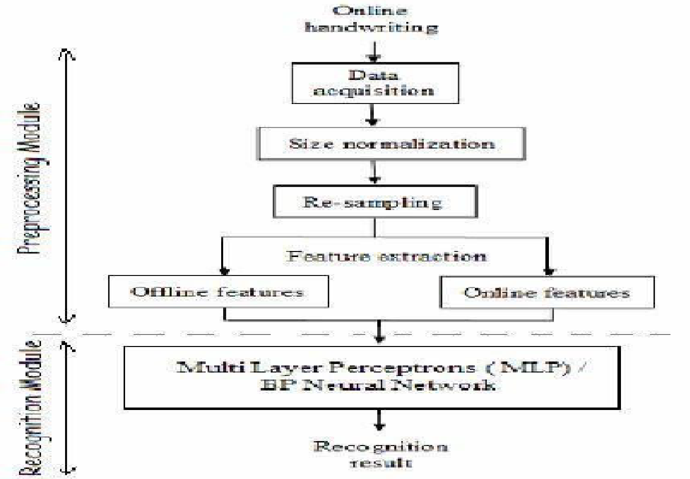


5.2 SOLUTION & TECHNICAL ARCHITECTURE

Project Description :

Handwritten digit recognition is one of the important problems in computer vision these days. There is a great interest in this field because of many potential applications, most importantly where a large number of documents must be dealt such as post mail sorting, bank cheque analysis, handwritten form processing etc. So a system should be designed in such a way that it is capable of reading handwritten digits and providing appropriate results. We propose a solution on neural network approaches to recognize handwritten digits.

Technical Architecture :



Working :

Dataset used :

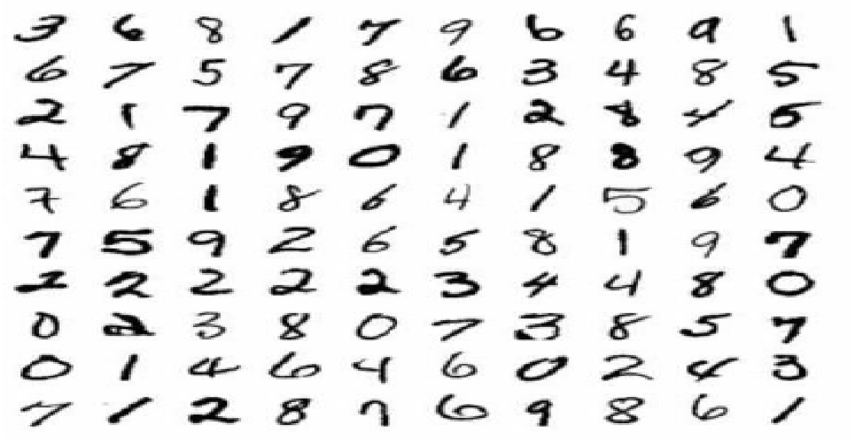
MNIST ("Modified National Institute of Standards and Technology") is the “Hello World” dataset of computer vision. Since its release in 1999, this classic dataset of handwritten images has served as the basis for benchmarking classification algorithms. As new machine learning techniques emerge, MNIST remains a reliable resource for researchers and learners alike.

In this competition, we aim to correctly identify digits from a dataset of tens of thousands of handwritten images. Kaggle has curated a set of tutorial-style kernels which cover everything from regression to neural networks. They hope to encourage us to experiment with different algorithms to learn first-hand what works well and how techniques compare.

Dataset description :

For this competition, we will be using Keras (with TensorFlow as our backend) as the main package to create a simple neural network to predict, as accurately as we can, digits from handwritten images. In particular, we will be calling the Functional Model API of Keras, and creating a 4-layered and 5-layered neural network.

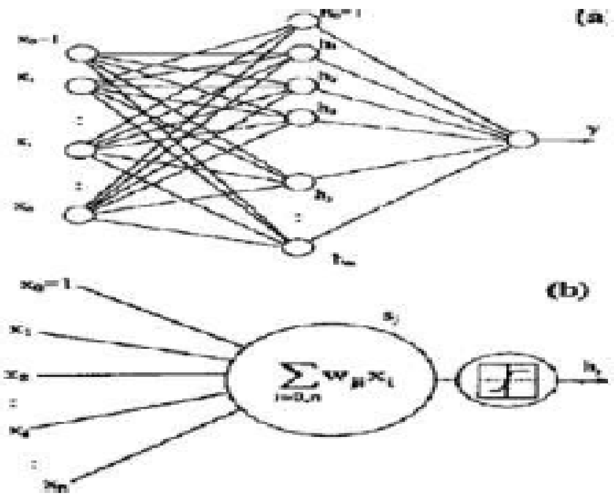
The MNIST Handwritten Digit Recognition Dataset contains 60,000 training and 10,000 testing labeled handwritten digit pictures. Each picture is 28 pixels in height and 28 pixels wide, for a total of 784 (28×28)pixels. Each pixel has a single pixel value associated with it. It indicates how bright or dark that pixel is (larger numbers indicate darker pixel). This pixel value is an integer ranging from 0 to 255.



Procedure :

- Install the latest TensorFlow library.
- Prepare the dataset for the model.
- Develop Single Layer Perceptron model for classifying the handwritten digits. Plot the change in accuracy per epochs.
- Evaluate the model on the testing data.
- Analyze the model summary.
- Add a hidden layer to the model to make it a Multi-Layer Perceptron.
- Add Dropout to prevent overfitting and check its effect on accuracy.
- Increasing the number of Hidden Layer neurons and checking its effect on accuracy.
- Use different optimizers and check its effect on accuracy.
- Increase the hidden layers and check its effect on accuracy.
- Manipulate the batch size and epochs and check its effect on accuracy.

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of



Working :

Neural Networks receive an input and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer. Neurons in a single layer function completely independently. The last fully connected layer is called the "output layer“.

Tensor flow :

TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. See the sections below to get started. By scanning the numerical digit and converting it into png format using the python3 command in the terminal we can get text output and sound output.

Feature extraction :

All neurons in a feature share the same weights .In this way all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.

Classification :

Convolutional neural network that is very popular for computer vision tasks like image classification, object detection, image segmentation and a lot more. Image classification is one of the most needed techniques in today's era, it is used in various domains like healthcare, business, and a lot more.

Result :

We do not consider our results to be flawless, as with any study or project undertaken in the field of machine learning and image processing. There is always opportunity for improvement in your methods because machine learning is a topic that is continually developing; there will always be a fresh new idea that solves a given problem more effectively. Three models were used to test the application: Multi-Layer Perceptron (MLP), Convolution NeuralNetwork, and (CNN). We obtain a different classifier accuracy with each model, indicating which is superior.

Technical Architecture:

The architectural diagram of the model is as below and the Technology used is shown in table1 & table 2

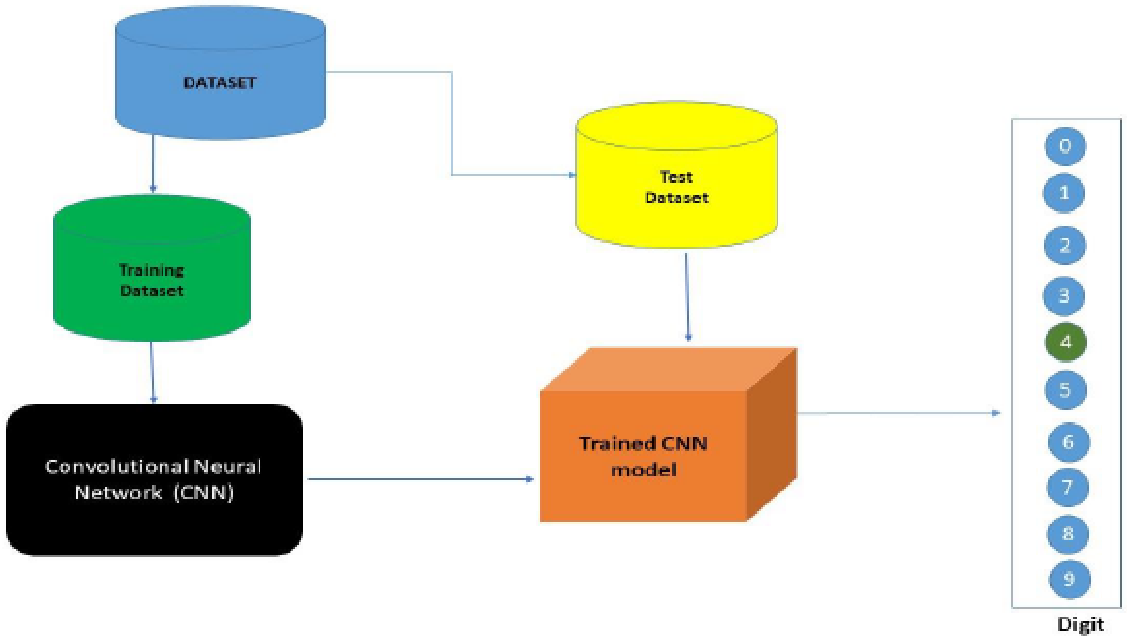


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with the application e.g. e.g., Mobile Application	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	IBM AI Platform
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / AI Local Server Configuration: AI Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Deep learning frameworks can help you upload data and train a deep learning model that would lead to accurate and intuitive predictive analysis.	Tensorflow, PyTorch
2.	Security Implementations	The system should automatically be able to authenticate all users with their unique username and password.	N/A
3.	Scalable Architecture	The system should be able to handle 10000 users accessing the site at the same time	N/A
4.	Availability	Information is restricted to each users and limited access	N/A
5.	Performance	Should reduce the delay in information when hundreds of requests are given	Google Co-Lab Pro/ Require high-end system.

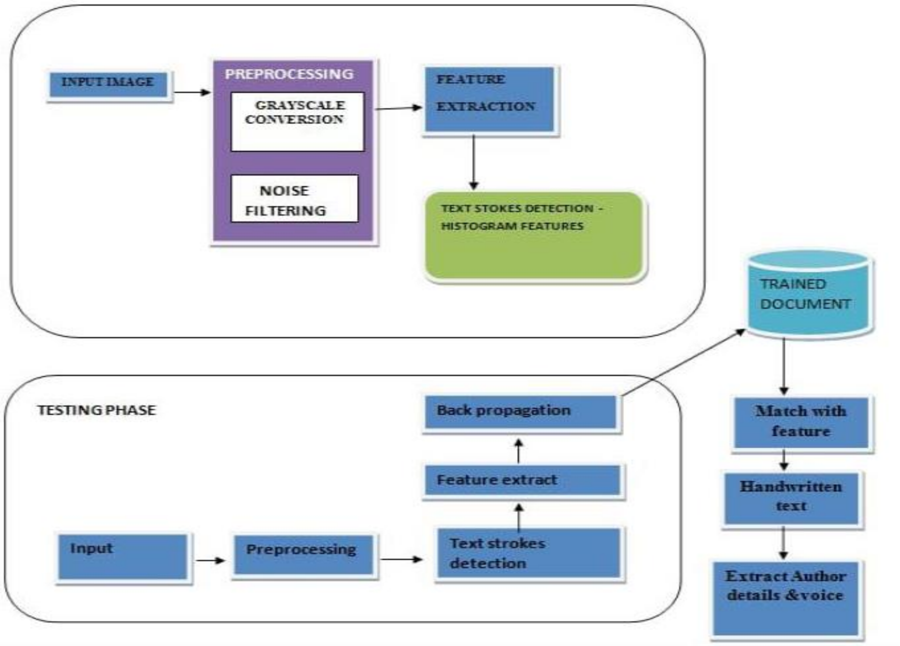
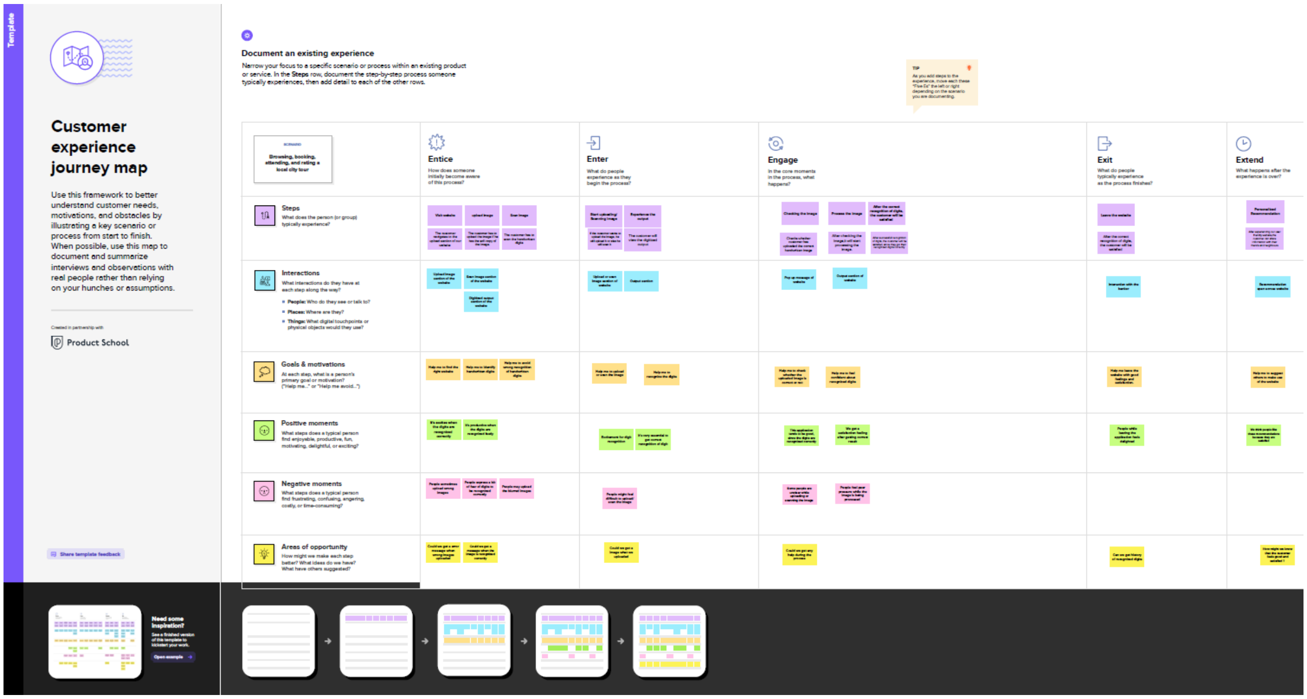


FIG. 1. BLOCK DIAGRAM

5.3 USER STORIES



Template

Customer experience journey map

Use this framework to better understand customer needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. When possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.

Created in partnership with

Product School

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Document an existing experience

Narrow your focus to a specific scenario or process within an existing product or service. In the **Steps** row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.

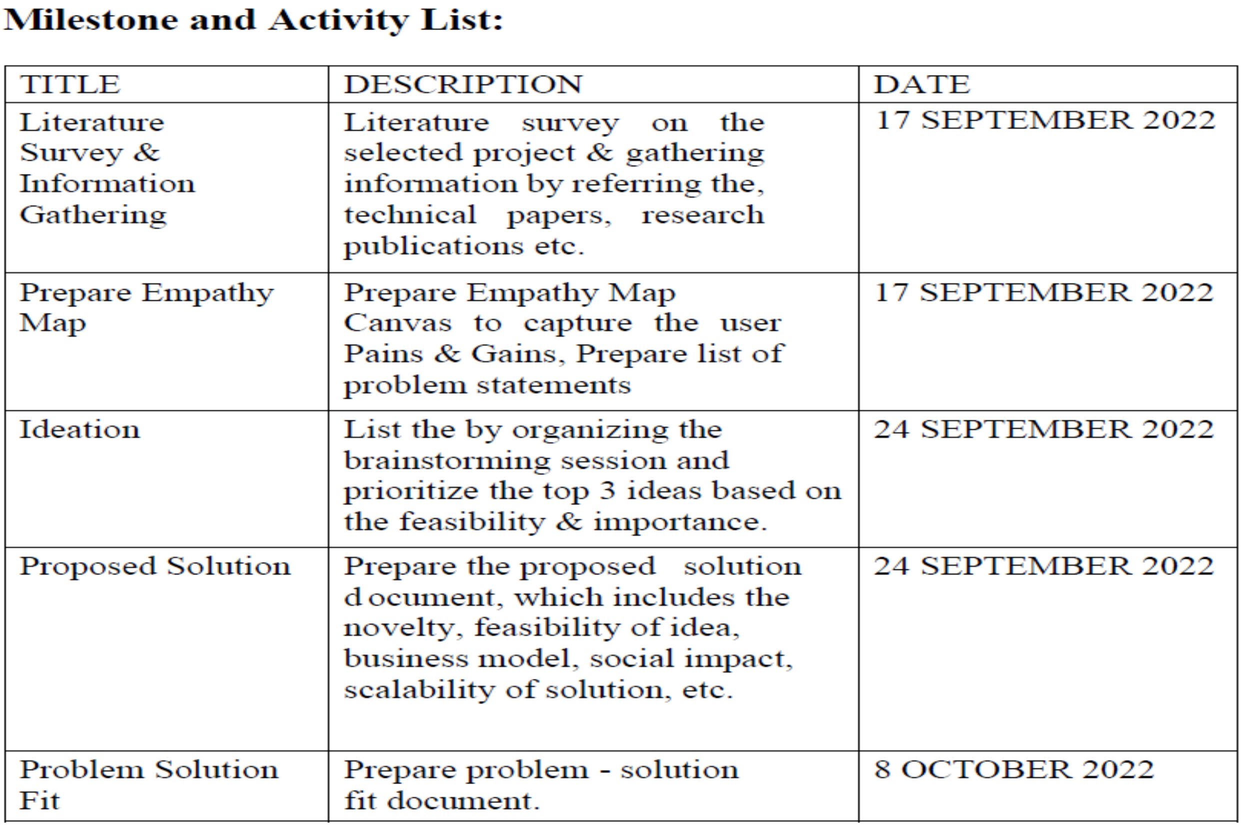
TIP

As you add steps to the experience, move each these "Five Es" the left or right depending on the scenario you are documenting.

SCENARIO Browsing, booking, attending, and rating a local city tour	**Entice** How does someone initially become aware of this process?	**Enter** What do people experience as they begin the process?	**Engage** In the core moments in the process, what happens?	**Exit** What do people typically experience as the process finishes?	**Extend** What happens after the experience is over?
Steps What does the person (or group) typically experience?	Visit website upload image Scan image The customer navigates in the upload section of our website The customer has to upload the image if he has the soft copy of the image The customer has to scan the handwritten digits	Start uploading/ Scanning image Experience the output If the customer wants to upload the image, he will upload it or else he will scan it The customer will view the digitized output	Checking the image Process the image After the correct recognition of digits, the customer will be satisfied Checks whether customer has uploaded the correct handwritten image After checking the image,it will start processing the image. [illegible]	Leave the website After the correct recognition of digits, the customer will be satisfied	Personalized Recommendation After experiencing our user friendly website,the customer can share information with their friends and neighbours
Interactions What interactions do they have at each step along the way? ▪ **People:** Who do they see or talk to? ▪ **Places:** Where are they? ▪ **Things:** What digital touchpoints or physical objects would they use?	Upload image section of the website Scan image section of the website Digitized output section of the website	Upload or scan image section of website Output section	Pop up message of website Output section of website	Interaction with the banker	Recommendation upon a nice website
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")	Help me to find the right website Help me to identify handwritten digits Help me to avoid wrong recognition of handwritten digits	Help me to upload or scan the image Help me to recognize the digits	Help me to check whether the uploaded image is correct or not Help me to feel confident about recognized digits	Help me leave the website with good feelings and satisfaction.	Help me to suggest others to make use of the website
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	It's soothes when the digits are recognized correctly It's productive when the digits are recognized easily	Excitement for digit recognition It's very essential to get correct recognition of digit	This application tends to be good, since the digits are recognized correctly We get a satisfaction feeling after getting correct result	People while leaving the application feels delighted	We think people like these recommendations because they are satisfied
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	People sometimes upload wrong images People express a lot of fear of digits to be recognized correctly People may upload the blurred images	People might feel difficult to upload/ scan the image	Some people are unclear while uploading or scanning the image People feel peer pressure while the image is being processed		
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested?	Could we get a error message when wrong images uploaded Could we get a message when the image is recognized correctly	Could we get a image what we uploaded	Could we get any help during the process	Can we get history of recognized digits	How might we know that the customer feels good and satisfied !

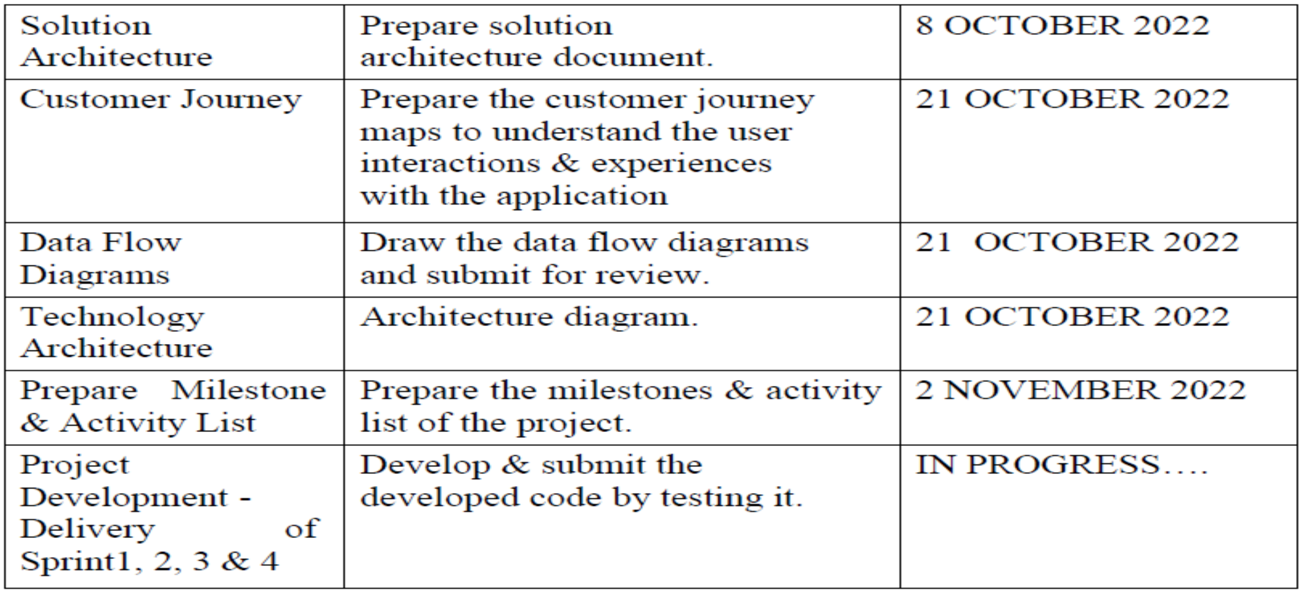
6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION



Milestone and Activity List:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	17 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	17 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	24 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution d ocument, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	8 OCTOBER 2022



Solution Architecture	Prepare solution architecture document.	8 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	21 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	21 OCTOBER 2022
Technology Architecture	Architecture diagram.	21 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	2 NOVEMBER 2022
Project Development - Delivery of Sprint1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS….

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points
Sprint-1	Data Collection and Pre-processing	USN-1	As a user, I can upload any kind of image with the pre-processing step is involved in it.	2
Sprint-1		USN-2	As a user, I can upload the image to any resolution.	1
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML the model which provides high accuracy of recognized handwritten digit.	2
Sprint-2		USN-4	As a user, I can pass the handwritten digit image for recognizing the digit and get the recognized digit.	1
Sprint-3	Building User Interface Application	USN-5	As a user, I will upload the handwritten digit image to the application by clicking an upload button and see the predicted/recognized digits in the application.	2
Sprint-4	Train and deployment of the model in the IBM Cloud	USN-6	As a user, I can access the web application and make use of the product from anywhere.	1

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	S (
Sprint-1	2	6 Days	24 Oct 2022	29 Oct 2022	2	2
Sprint-2	2	6 Days	31 Oct 2022	05 Nov 2022	2	0
Sprint-3	2	6 Days	07 Nov 2022	12 Nov 2022	2	1
Sprint-4	2	6 Days	14 Nov 2022	19 Nov 2022	2	1

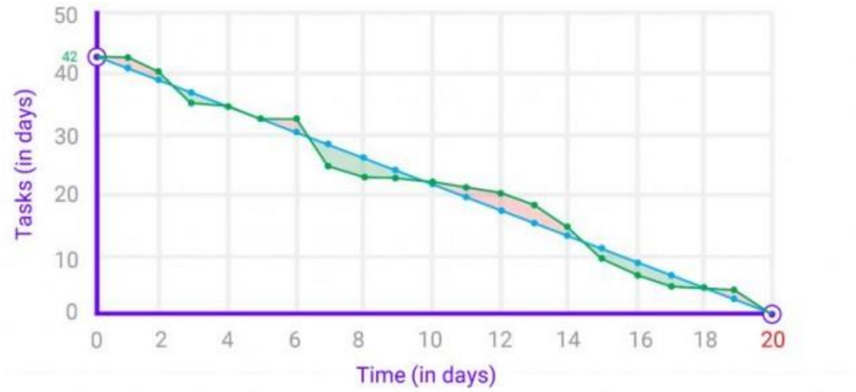
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

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.



7. CODING & SOLUTIONING

Understanding The Data

Importing the required libraries

:

```
import numpy
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.optimizers import Adam
from keras.utils import np_utils
```

Loading the data

In [48]:

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

In [49]:

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

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

Analyzing the data

In [50]:

```
X_train[0]
```

Out[50]:

```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
         90,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 139, 253,
        190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
        253,  70,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
        148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
        253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
        253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
        195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
         11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0]], dtype=uint8)
```

In [51]:

```
y_train[0]
```

Out[51]:

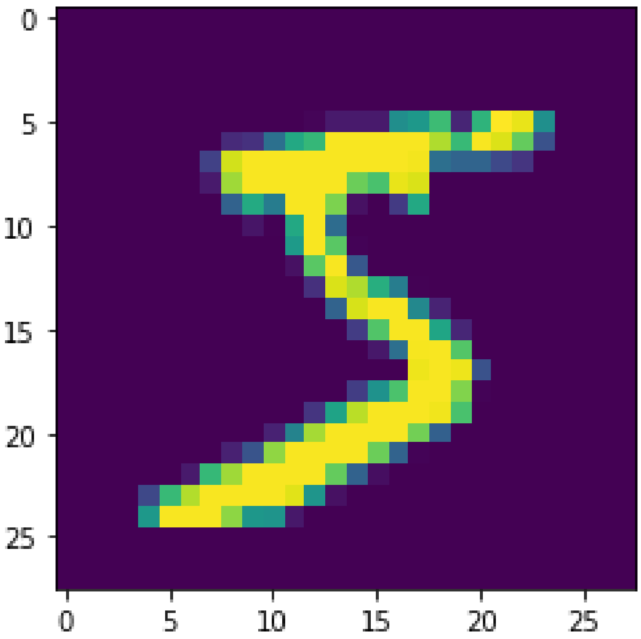
```
5
```

In [52]:

```
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
```

Out[52]:

```
<matplotlib.image.AxesImage at 0x17a91a21100>
```



Reshaping the data

In [53]:

```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

Applying One Hot Encoding

In [54]:

```
number_of_classes = 10
y_train = np_utils.to_categorical(y_train, number_of_classes)
y_test = np_utils.to_categorical(y_test, number_of_classes)
```

In [55]:

```
y_train[0]
```

Out[55]:

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

Model Building

In [56]:

```
model = Sequential()
```

Add CNN Layers

In [57]:

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

Compiling the Model

In [58]:

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

Train the Model

In [59]:

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=5,batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 124s 66ms/step - loss: 0.2609 - accuracy: 0.9500
- val_loss: 0.0936 - val_accuracy: 0.9703
Epoch 2/5
1875/1875 [==============================] - 127s 68ms/step - loss: 0.0723 - accuracy: 0.9775
- val_loss: 0.0916 - val_accuracy: 0.9722
Epoch 3/5
1875/1875 [==============================] - 125s 67ms/step - loss: 0.0495 - accuracy: 0.9840
- val_loss: 0.0861 - val_accuracy: 0.9763
Epoch 4/5
1875/1875 [==============================] - 127s 68ms/step - loss: 0.0376 - accuracy: 0.9876
- val_loss: 0.1267 - val_accuracy: 0.9714
Epoch 5/5
1875/1875 [==============================] - 123s 65ms/step - loss: 0.0311 - accuracy: 0.9902
- val_loss: 0.1045 - val_accuracy: 0.9781
```

Out[59]:

```
<keras.callbacks.History at 0x17a91969940>
```

Observing the Metrics

In [62]:

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics(Test loss & Test Accuracy): ")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy):
[0.10448186099529266, 0.9781000018119812]
```

Test the Model

In [63]:

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

```
[[4.64307141e-14 9.41230704e-18 4.35460451e-14 2.37232678e-10
  8.19107247e-18 1.01514965e-18 5.26431904e-26 1.00000000e+00
  2.99863284e-14 1.21474001e-12]
 [1.10410819e-10 2.67242872e-09 9.99998212e-01 1.56112367e-08
  1.33440385e-15 1.30546174e-17 1.79986603e-06 1.23274669e-17
  3.49710122e-11 2.44014946e-19]
 [4.22704433e-13 9.99994636e-01 1.11693601e-06 1.44851945e-13
  3.41596440e-09 1.15770497e-10 8.58048840e-11 1.36920466e-08
  4.30813316e-06 1.51135549e-10]
 [1.00000000e+00 4.40986330e-17 2.06547930e-12 1.14893435e-17
  5.04777020e-14 1.04432565e-11 4.01654855e-11 6.38017905e-13
  2.91141694e-11 8.11927470e-10]]
```

In [64]:

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

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

Observing the Metrics

In [65]:

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics(Test loss & Test Accuracy)): ")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy)):
[0.10448186099529266, 0.9781000018119812]
```

Test the Model

In [66]:

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

```
[[4.64307141e-14 9.41230704e-18 4.35460451e-14 2.37232678e-10
  8.19107247e-18 1.01514965e-18 5.26431904e-26 1.00000000e+00
  2.99863284e-14 1.21474001e-12]
 [1.10410819e-10 2.67242872e-09 9.99998212e-01 1.56112367e-08
  1.33440385e-15 1.30546174e-17 1.79986603e-06 1.23274669e-17
  3.49710122e-11 2.44014946e-19]
 [4.22704433e-13 9.99994636e-01 1.11693601e-06 1.44851945e-13
  3.41596440e-09 1.15770497e-10 8.58048840e-11 1.36920466e-08
  4.30813316e-06 1.51135549e-10]
 [1.00000000e+00 4.40986330e-17 2.06547930e-12 1.14893435e-17
  5.04777020e-14 1.04432565e-11 4.01654855e-11 6.38017905e-13
  2.91141694e-11 8.11927470e-10]]
```

Saving the Model

In [67]:

```
model.save('models/mnistCNN.h5')
```

8. TESTING

8.1 TEST CASE

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	FAIL
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 the 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 non image file	User is able to upload any file	FAIL

HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the 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 properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_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 be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

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
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1

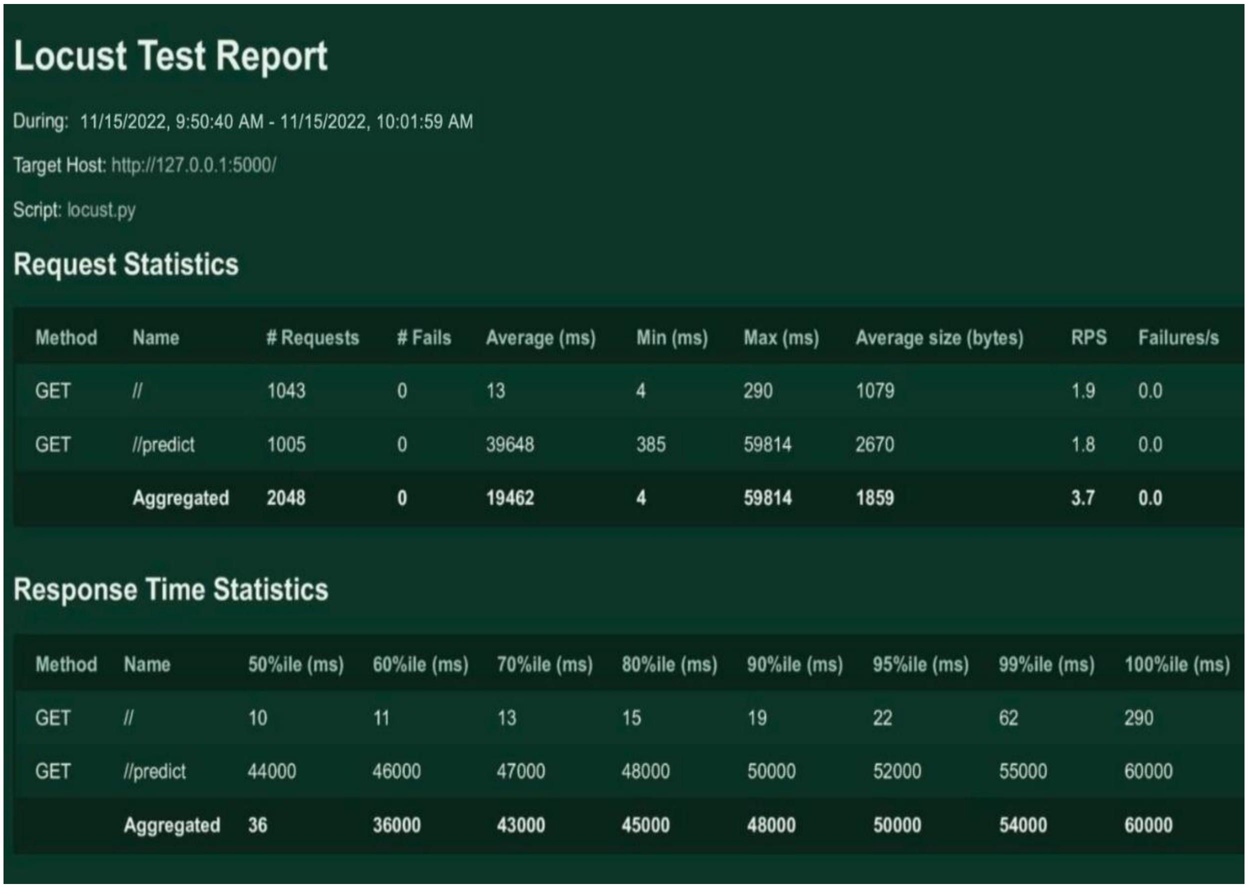
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

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

9. RESULTS

9.1 PERFORMANCE METRICS



Locust Test Report

During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 AM

Target Host: http://127.0.0.1:5000/

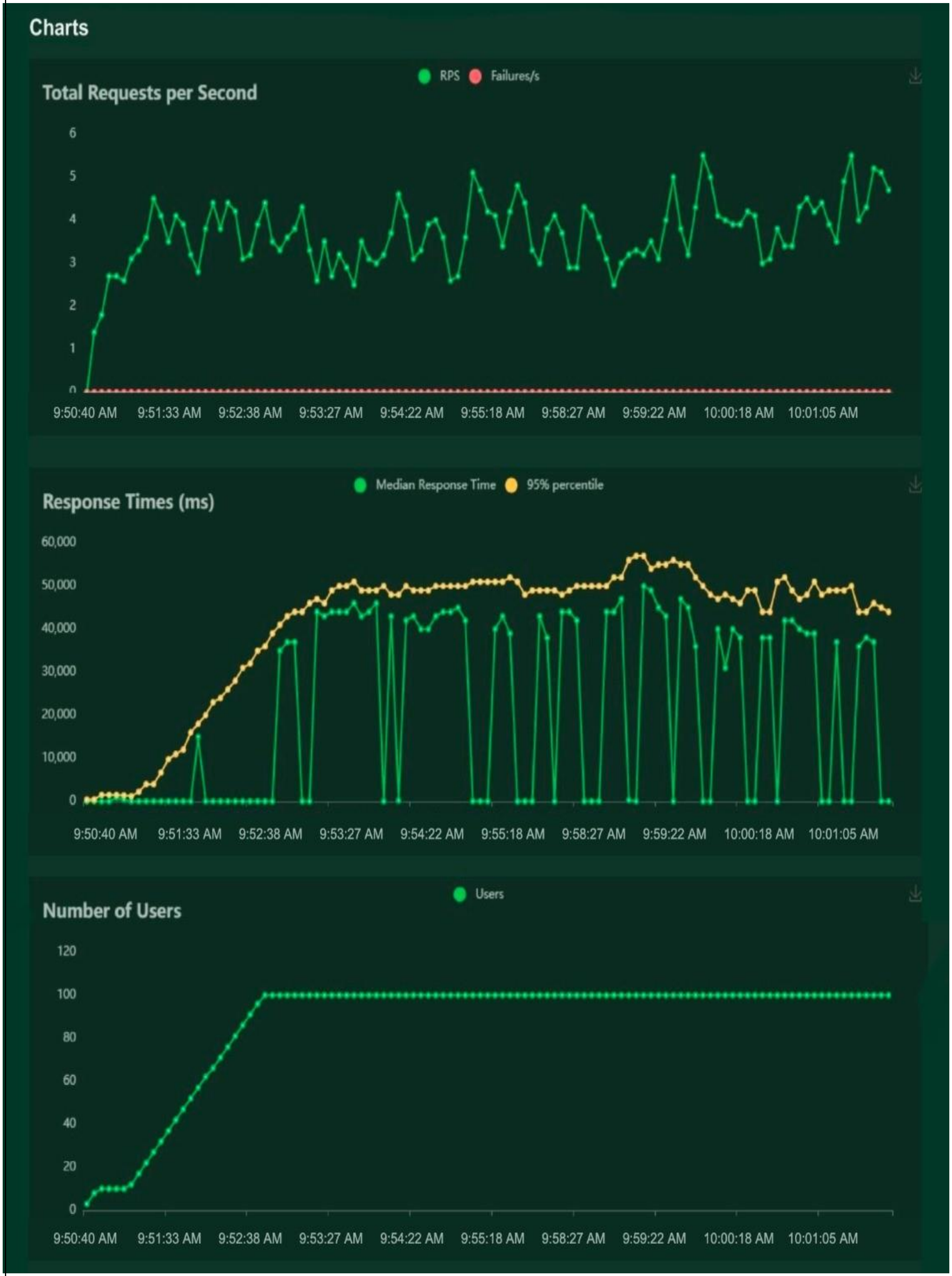
Script: locust.py

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**


Charts
Total Requests per Second
● RPS ● Failures/s
6
5
4
3
2
1
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Response Times (ms)
● Median Response Time ● 95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Number of Users
● Users
120
100
80
60
40
20
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM

10. ADVANTAGES & DISADVANTAGES

❖ ADVANTAGE

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

❖ DISADVANTAGE

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

11. CONCLUSION

- This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is 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. There is so much room for improvement, which can be implemented in subsequent versions.

12.FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

- The most required application today is Speech recognition. The recognized Printed or Handwritten character could be recorded and through a voice synthesizer speech output could be generated. This would help the blind to send and receive information.
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world
- This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall

work efficiency.

13. APPENDIX

SOURCE CODE

app.py

```
from flask import Flask, request, render_template, flash
from PIL import Image
import numpy as np
import os
from werkzeug.utils import secure_filename
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import tensorflow as tf


app = Flask(__name__, template_folder='template')
app.config['UPLOAD_FOLDER']= 'uploads/'
model = load_model("./models/mnistCNN.h5")
@app.route('/')
def batch():
    return render_template("index.html")
```

```
@app.route('/web')
def batch2():
    return render_template("web.html")


@app.route('/web',methods=['GET','POST'])
def web():
    imagefile = request.files['imagefile']
    image_path ="./uploads/"+imagefile.filename
    imagefile.save(image_path)
    img = image.load_img(image_path).convert("L")
    img = img.resize((28, 28))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1, 28, 28, 1)
    y_pred = model.predict(im2arr)
    pred = np.argmax(y_pred, axis=1)
    index = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
    output = str(index[pred[0]])
    return render_template('web.html', prediction=output)





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

Hand written recognition system.ipynb

Understanding The Data

Importing the required libraries

:

```
import numpy
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.optimizers import Adam
from keras.utils import np_utils
```

Loading the data

In [48]:

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

In [49]:

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

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

Analyzing the data

In [50]:

```
X_train[0]
```

Out[50]:

```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
  18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
 253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
 253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
 253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
 205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
  90,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 139, 253,
 190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
 253,  70,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  35,
 241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
 148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
 195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
  11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```

In [51]:

```
y_train[0]
```

Out[51]:

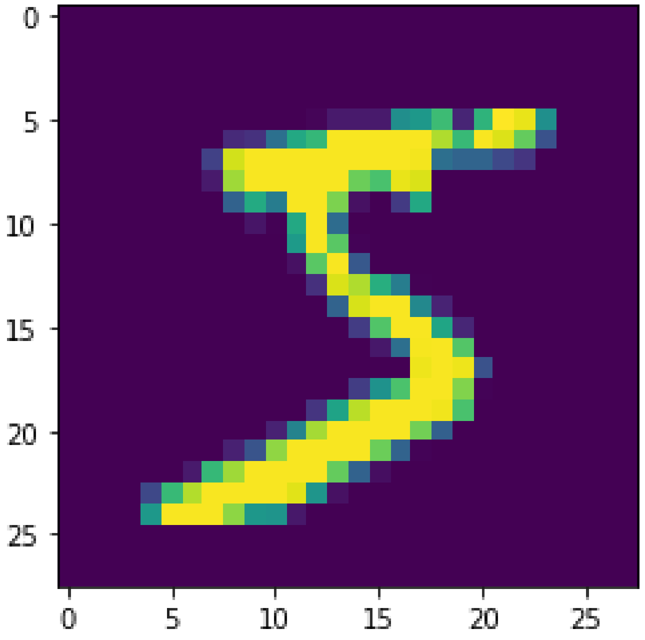
```
5
```

In [52]:

```
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
```

Out[52]:

```
<matplotlib.image.AxesImage at 0x17a91a21100>
```



Reshaping the data

In [53]:

```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

Applying One Hot Encoding

In [54]:

```
number_of_classes = 10
y_train = np_utils.to_categorical(y_train, number_of_classes)
y_test = np_utils.to_categorical(y_test, number_of_classes)
```

In [55]:

```
y_train[0]
```

Out[55]:

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

Model Building

In [56]:

```
model = Sequential()
```

Add CNN Layers

In [57]:

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

Compiling the Model

In [58]:

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

Train the Model

In [59]:

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=5,batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 124s 66ms/step - loss: 0.2609 - accuracy: 0.950
0 - val_loss: 0.0936 - val_accuracy: 0.9703
Epoch 2/5
1875/1875 [==============================] - 127s 68ms/step - loss: 0.0723 - accuracy: 0.977
5 - val_loss: 0.0916 - val_accuracy: 0.9722
Epoch 3/5
1875/1875 [==============================] - 125s 67ms/step - loss: 0.0495 - accuracy: 0.984
0 - val_loss: 0.0861 - val_accuracy: 0.9763
Epoch 4/5
1875/1875 [==============================] - 127s 68ms/step - loss: 0.0376 - accuracy: 0.987
6 - val_loss: 0.1267 - val_accuracy: 0.9714
Epoch 5/5
1875/1875 [==============================] - 123s 65ms/step - loss: 0.0311 - accuracy: 0.990
2 - val_loss: 0.1045 - val_accuracy: 0.9781
```

Out[59]:

```
<keras.callbacks.History at 0x17a91969940>
```

Observing the Metrics

In [62]:

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics(Test loss & Test Accuracy): ")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy):
[0.10448186099529266, 0.9781000018119812]
```

Test the Model

In [63]:

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

```
[[4.64307141e-14 9.41230704e-18 4.35460451e-14 2.37232678e-10
  8.19107247e-18 1.01514965e-18 5.26431904e-26 1.00000000e+00
  2.99863284e-14 1.21474001e-12]
 [1.10410819e-10 2.67242872e-09 9.99998212e-01 1.56112367e-08
  1.33440385e-15 1.30546174e-17 1.79986603e-06 1.23274669e-17
  3.49710122e-11 2.44014946e-19]
 [4.22704433e-13 9.99994636e-01 1.11693601e-06 1.44851945e-13
  3.41596440e-09 1.15770497e-10 8.58048840e-11 1.36920466e-08
  4.30813316e-06 1.51135549e-10]
 [1.00000000e+00 4.40986330e-17 2.06547930e-12 1.14893435e-17
  5.04777020e-14 1.04432565e-11 4.01654855e-11 6.38017905e-13
  2.91141694e-11 8.11927470e-10]]
```

In [64]:

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

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

Observing the Metrics

In [65]:

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics(Test loss & Test Accuracy)): ")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy)):
[0.10448186099529266, 0.9781000018119812]
```

Test the Model

In [66]:

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

```
[[4.64307141e-14 9.41230704e-18 4.35460451e-14 2.37232678e-10
  8.19107247e-18 1.01514965e-18 5.26431904e-26 1.00000000e+00
  2.99863284e-14 1.21474001e-12]
 [1.10410819e-10 2.67242872e-09 9.99998212e-01 1.56112367e-08
  1.33440385e-15 1.30546174e-17 1.79986603e-06 1.23274669e-17
  3.49710122e-11 2.44014946e-19]
 [4.22704433e-13 9.99994636e-01 1.11693601e-06 1.44851945e-13
  3.41596440e-09 1.15770497e-10 8.58048840e-11 1.36920466e-08
  4.30813316e-06 1.51135549e-10]
 [1.00000000e+00 4.40986330e-17 2.06547930e-12 1.14893435e-17
  5.04777020e-14 1.04432565e-11 4.01654855e-11 6.38017905e-13
  2.91141694e-11 8.11927470e-10]]
```

Saving the Model

In [67]:

```
model.save('models/mnistCNN.h5')
```

index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

                        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

           <title>Handwritten Digit Recognition System</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

                              <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

     <link rel="stylesheet" type="text/css" href="/static/css/style2.css">

<style>

                              body{

                                   background-image: url("../static/css/bg-img.jpg");

                                   background-size: cover;

                                }

</style>

</head>

<body>

<div id="home">

                     <div class = "landing-text">

                                   <a href="./" class="btn btn-default btn-lg">Home</a>

                                   <a href="./web" class="btn btn-default btn-lg">Recognize</a>

                                   <h1>A Novel Method For Handwritten Digit Recognition System</h1>
```

```
                                    </div>
</div>
</body>
</html>
```

web.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
                       <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
          <title>Handwritten Digit Recognition System</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
                             <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>


</head>
<style>
  body{
    background-image: url("../static/css/bg-img.jpg");
    background-size: cover;
  }
 .fit-image{
    width: 100%;
```

```
        object-fit: cover;

        height: 200px; /* only if you want fixed height */


     }

   .img{

       height: 200px;

       width: 200px;

   }

   img{

  max-width:180px;

}



</style>

<body>

<div class="head">


</div>

<div class="container">

    <div id="content" style="...">

        <div class="container">

            <div class="row">


                <div class="col">

                    <div class="text-center"></div>

                </div>
                <div class="col text-center">
                    <form action="/web" method="post" enctype="multipart/form-data">
                        <h3 class = "page-header text-primary" style="color: #FFFFFF">Upload Image</h3>
                        <div class="form-group">
                            <label style="color: #FFFFFF">Browse Image</label>
                            <input type="file" onchange="preview()" class="form-control" name="imagefile">
                            <img id="frame" width="100px" height="100px"/>

                        </div>
                        <div class="text-center">
                            <input class="btn btn-primary mt-3" type="submit" value="Recognize">
                        </div>
                    </form>
                        <div>
                            {% if prediction%}
                                <p style="font-size:160%; color:#FFFFFF;" class="text-center">Predicted Number is {{prediction}}</p>
                            {% endif %}
                        </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
</body>
<script>
   function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
        });
    </script>
</html>
```

GITHUB

- https://github.com/IBM-EPBL/IBM-Project-43040-1660712213

PROJECT DEMO LINK

- https://drive.google.com/file/d/1EXFeTagnaCDuhXWPj9CYG-lyU6wWP-Yu/view?usp=share_link