PROJECT NAME	A Novel Method For Handwritten Digit Recognition System
TEAM ID	PNT2022TMID18129
COLLEGE	SNS College of Technology
TEAM LEADER	Vinuppriya.B
TEAM MEMBERS	Sindhuja.K Suguna.S Sumithasri.A
FACULTY MENTOR NAME	Janani.S.R
INDUSTRY MENTOR NAME	Pradeepthi

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

1. INTRODUCTION

The handwritten digit recognition is the ability of computers 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 flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

1.1 Project Overview:

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc. Here comes the use of Deep Learning. In the past decade, deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machine learning tools have been developed like scikit-learn, scipy-image etc. and pybrains, Keras, Theano, Tensorflow by Google, TFLearn etc. for Deep Learning. These tools make the applications robust and therefore more accurate. The Artificial Neural Networks can almost mimic the human brain and are akey ingredient in image processing field. For example, Convolutional Neural Networks with Back Propagation for Image Processing,

Deep Mind by Google for creating Art by learning from existing artist styles etc..Handwriting Recognition has an active community of academics studying it. The biggest conferences for handwriting recognition are the International Conference on Frontiers in Handwriting Recognition (ICFHR), held in even-numbered years, and the International Conference on Document Analysis and Recognition (ICDAR), held in odd-numbered years. Both of these conferences are endorsed by the IEEE. Active areas of research include: Online Recognition, OfflineRecognition,Signature Verification, Postal-Address Interpretation,Bank-Check Processing, Writer Recognition.Classification of images and patterns has been one of the major implementation of Machine Learning and Artificial Intelligence. People are continuously trying to make computers intelligent so that they can do almost all the work done by humans Handwriting recognition system is the most basic and an important step towards this huge and interesting area of Computer Vision.

1.2 Purpose:

The project aims to implement the concept of Convolution Neural Network which is one of the important architecture of deep learning. Understanding CNN and applying it to the handwritten recognition system, is the major target of the proposed system. There is a reason behind using CNN for handwritten digit recognition. Let us consider a multi-layer feedforward neural network to be applied on MNIST dataset which contains images of size 28×28 pixels (roughly 784 pixels). So if a hidden layer has about 100 units, then the first layer weights comes up to about 78k parameters, which is large but manageable. However, in the natural world the size of the image is much larger. If we consider the size of the typical image which is around 256×256 pixels (roughly about 56,000 pixels), then the first layer weights will have about 560k parameters! So that becomes too many parameters and hence make it unscalable for real images. Hence, it will be so large that it will become very difficult to generalize the new data fed into the network. Convolution Neural Network extracts the feature maps from the 2D images by applying filters and hence making the task of feature extraction from the images easier. Basically, convolution neural network considers the mapping of image pixels with the neighbourhood space rather than having a fully connected layer of neurons. Convolution Neural Networks has been proved to be a very important and powerful tool in signal and image processing. Even in the fields of computer vision such has handwriting recognition, natural object classification and segmentation, CNN has been a much better tool compared to allother previously implemented tools .The broader aim in mind was to develop a M.L. model that could recognize people's handwriting. However, as we

began developing the model we realized that the topic in hand was too tough and would require tremendous data to learn. Thus we settled on classifying a given handwritten digit image as the required digit and consequently testing its accuracy.

2.LITERATURE SURVEY

2.1 Existing Problem:

Poor quality or illegible handwriting is perhaps the most obvious issue when processing handwritten forms during the data capture process. We've all heard the stereotype about doctors' handwriting, so attempting to capture and validate accurate data on this type of form-filling may result in little meaningful data being extracted.

According to a 2012 study by the Educational Summit, "25-35% of students at a secondary school level have still not gained competency in the skill of handwriting," implying that forms filled out by hand may continue to pose a challenge to data collection processes.

2.2 References:

Paper 1: Handwritten Digit Recognition using Machine Learning Algorithms

Year: March 2018
Authors: S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana and Abdullah Al Jobair.

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning technique. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Several machines learning algorithm namely, Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA.

Paper 2: Hand-written digits recognition using miscellaneous machine learning and deep learning algorithms

Year: May 2022
Authors: Sarfraz Nawaz, Muhammad Arfan Jaffar, Syed Pervez, Hamza Afzal, Zaheer Udeen Babar

Identification of Hand-written digits is a rational key point in pattern identification applications. There are many uses of hand-written digits identification like mail sorting in postal, cheques processing in the banks, data entry through forms, etc. The key to the issue lies in the expertise to grow a well-organized algorithm that can accept hand-written numbers and which are submitted by end-users by the scanners, tablets, and other digital devices. This paper gives a viewpoint to handwritten numbers recognition constructed on machine learning models, and deep learning models and shows the outcomes in the shape of accuracy. The primary objective of this paper is to guarantee powerful and dependable methodologies for the acknowledgment of handwritten numbers using machine learning and deep learning algorithms. Several machine learning algorithms such as Decision Tree (DT), Naïve Bayesian (NB) classifier, Multilayer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), and deep learning algorithms such as Convolutional Neural Network (CNN), AlexNet, and Multilayer Perceptron (MLP) have been used for recognition of hand-written digits in Jupyter Notebook and Matlab. Through some features extraction, and different experiments and analysis of Machine Learning Algorithms (MLA) and Deep Learning Algorithms (DLA), the accuracy of deep learning algorithms is better than the machine learning algorithms. Keywords: Hand-written, Digits Recognition, MNIST Dataset, Machine Learning Models, Deep Learning Models, Algorithms, AlexNet, Features Extraction, fc6, fc7, and fc8. , Classification, Pattern Recognition, Supervised, Unsupervised Learning.

Paper 3: Digit Recognition Using MNIST Dataset

Year: Sep 2022
Authors: Anirudh Mhaske, Atharv Joshi, Dattaram Kajrekar, Ruturaj Jugdar, Prof. Ajita Mahapadi

In this paper, we have performed handwritten digit recognition using MNIST datasets using Support Vector Machines (SVM), Multi-Layer Perceptron (MLP) and Convolution Neural Network (CNN) models. Our main goal is to compare the accuracy of the above models along with their execution time to obtain the best possible model for digit recognition. Reliability of humans over machines has never been so high that from classifying objects in photographs to adding sound to silent movies can all be done using deep learning and machine learning algorithms. Similarly, handwriting recognition is one of the important areas research and development with a range of possibilities that could be achieved. Handwriting recognition (HWR), also known as handwritten text recognition (HTR), is a capability computers to receive and interpret comprehensible handwritten input from sources such as paper documents, photos, touch screen.

Paper 4: Digital Recognition of Handwritten Digits Using Convolutional Neural Networks

Year: August 2022
Authors: Anusha Nayak, Shrutha Jain, Tanya Shetty, K. Srikanth Bhat

Convolutional neural networks are widely used in vast scopes, and the number of sectors in which profound learning may be used is growing all the time in fields due to their different scope of uses like example pattern reputation, sentence classification, speech popularity, face reputation, textual content categorization, file evaluation, visual image evaluation, etc. In the current time of digitization, penmanship acknowledgment assumes a significant part in data handling by changing overwritten by hand characters into machine-decipherable organizations. The postal framework assumes a significant part in the improvement of mail transportation. The target of the task is to give an elective means to the customary arranging framework, which devours lesser time for handling and arranging the postcards in view of their individual regions. It likewise targets killing any conceivable human blunders which might happen during the manual arranging process.

Paper 5: Recognition of Handwritten Digit using Convolutional Neural Network (CNN)

Year: May 2019
Authors: Md. Anwar Hossain, Md. Mohon Ali

Humans can see and visually sense the world around them by using their eyes and brains. Computer vision works on enabling computers to see and process images in the same way that human vision does. Several algorithms developed in the area of computer vision to recognize images. The goal of our work will be to create a model that will be able to identify and determine the handwritten digit from its image with better accuracy. We aim to complete this by using the concepts of Convolutional Neural Network and MNIST dataset. We will also show how MatConvNet can be used to implement our model with CPU training as well as less training time. Though the goal is to create a model which can recognize the digits, we can extend it for letters and then a person's handwriting. Through this work, we aim to learn and practically apply the concepts of Convolutional Neural Networks.

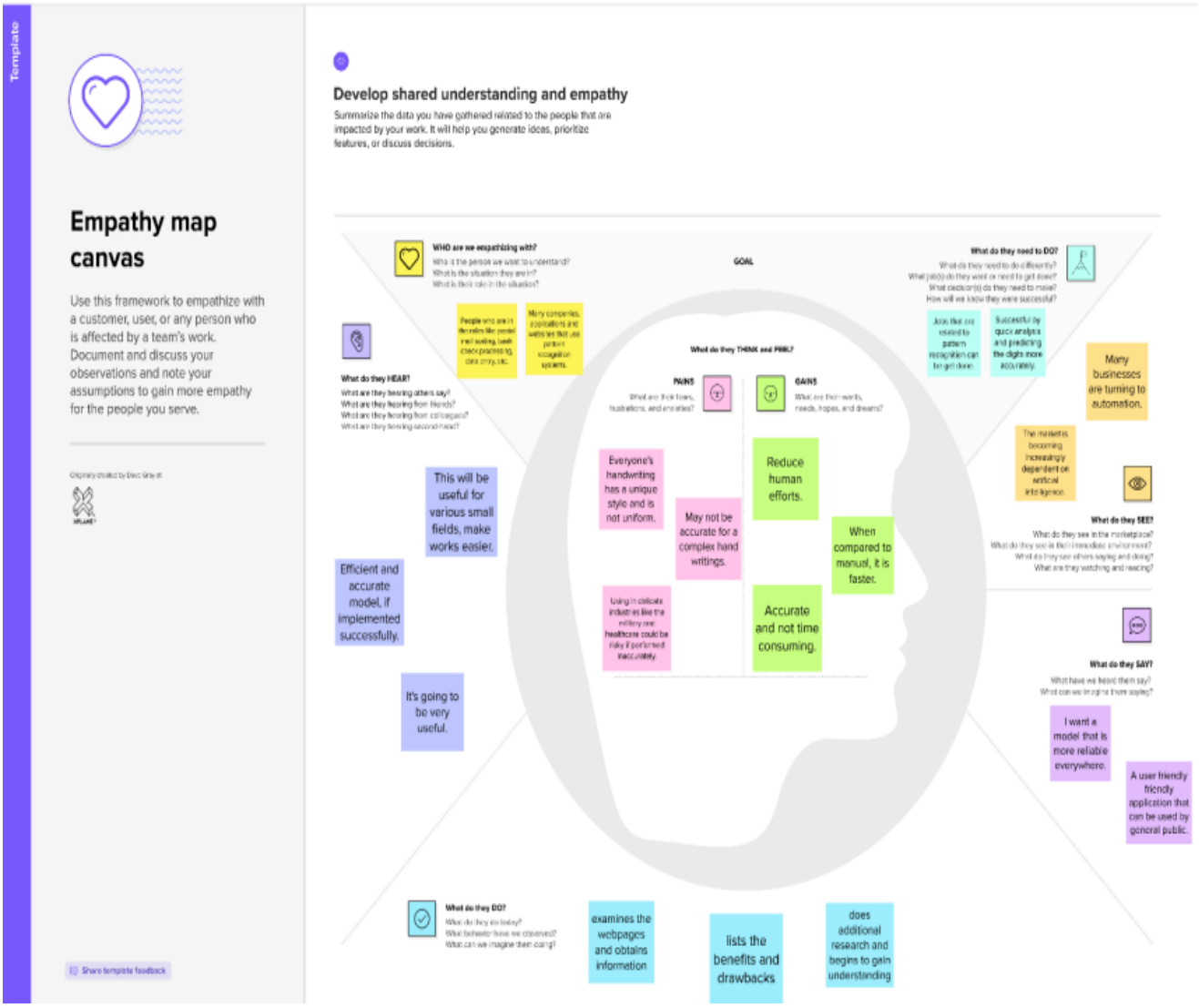
2.3 Problem Statement Definition:

Handwritten digit recognition is a challenging problem because it is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. And moreover false prediction may lead to drastic effects.

3.IDEATION & 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 helps 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.

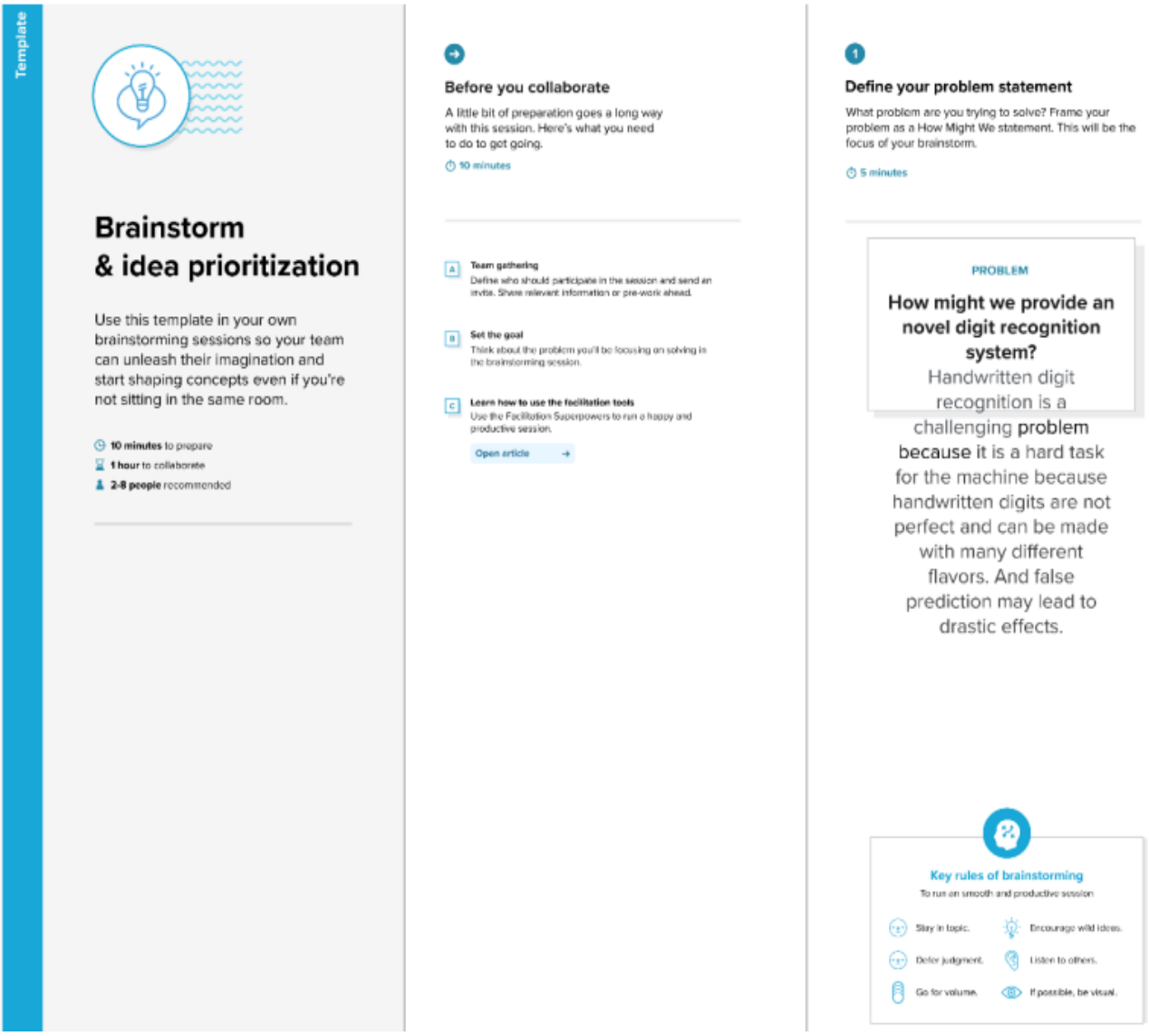


3.2 Ideation & Brainstorming:

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. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/empathy-map-canvas

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

Prioritize

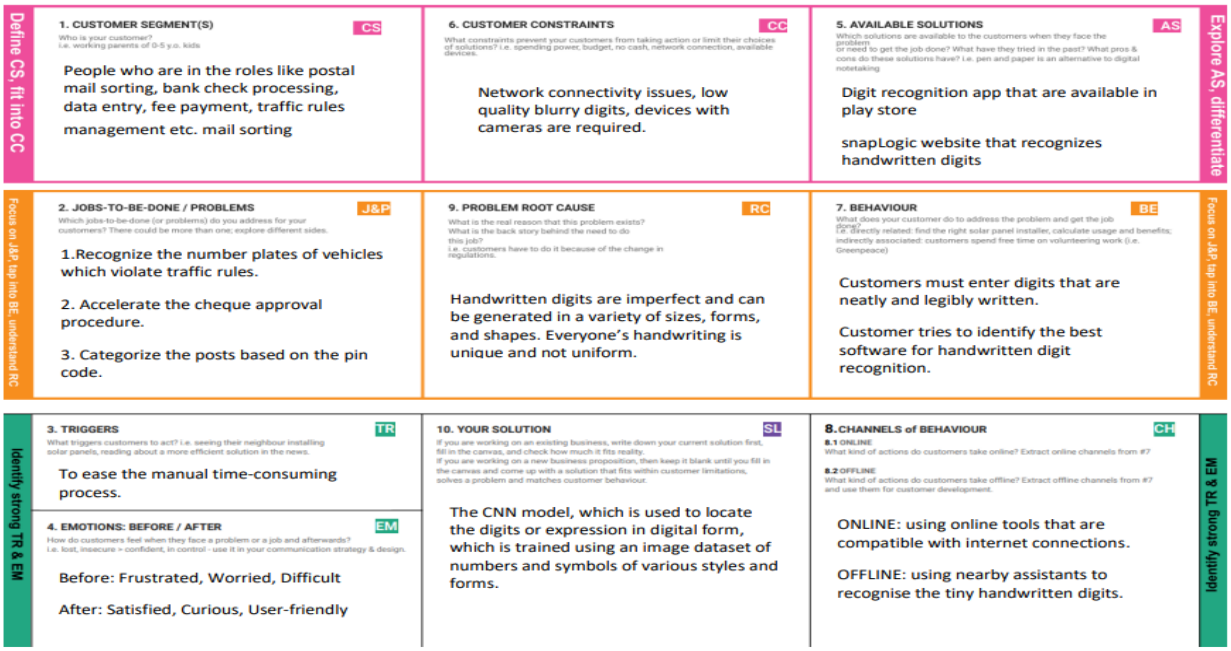
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Because handwritten digits are imperfect and can be generated in a variety of sizes, forms, and shapes, it is a challenging work for the machine.
2.	Idea / Solution description	The CNN model, which is used to locate the digits or expression in digital form, is trained using an image dataset of numbers and symbols of various styles and forms.
3.	Novelty / Uniqueness	Recognises symbols, can differentiate between known and unknown characters, and can work out mathematical formulas.
4.	Social Impact / Customer Satisfaction	This model used in real-time works fast, accurate and reduces human efforts, thus will have a good social impact and customer satisfaction.
5.	Business Model (Revenue Model)	The project can be integrated and used with people who are in the roles like postal mail sorting, bank check processing, data entry, fee payment, traffic rules management etc.
6.	Scalability of the Solution	Using cloud-native techniques is one of the applications to make handwritten digit recognition scalable. IBM cloud is one of the cloud-based AI scalability options. An application can effectively handle a vast volume of data and a big number of concurrent users.

3.4 Problem Solution fit:



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login via Registered Username Login via Email & Password

FR-4	User Authentication	Authentication through Captcha Banking Sector: Authentication through IFSC code & Finger Print. Library and postal Sector: Authentication through Identification Card.
FR-5	User Input	Upload the input as Scanned image Upload the input from database. Get the input as real time image.
FR-6	System Configuration	RAM at least 4GB System with graphical User Interface. Camera with better resolution
FR-7	Business Rules	System shows an error message when the input is not in a required format.

4.2 Non-Functional Requirements:

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

FR No	**Non-Functional Requirement**	**Description**
NFR-1	Usability	Useful for reducing complexity in fields that work with massive databases.
NFR-2	Security	Easy to track user. Access only to authorized person
NFR-3	Reliability	Due to extensive training, this model's accuracy has greatly increased.

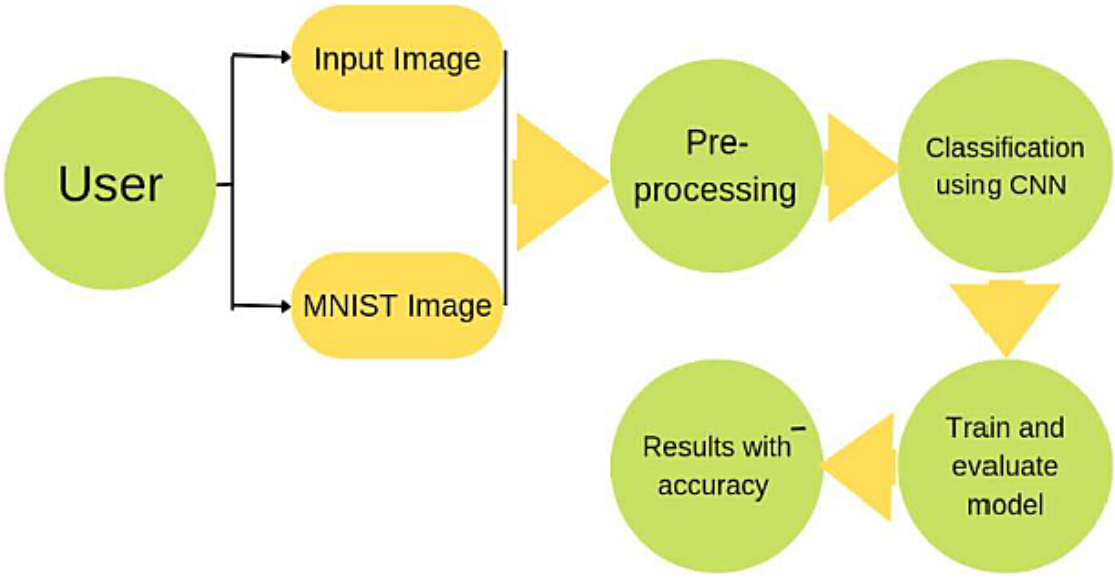
NFR-4	Performance	Decreases the need for human supervision and increases efficiency.
NFR-5	Availability	Accessible to all users, including banks, post offices, libraries, etc.
NFR-6	Scalability	Model is expected to be 95% accurate, and there is potential to expand the model to identify text.

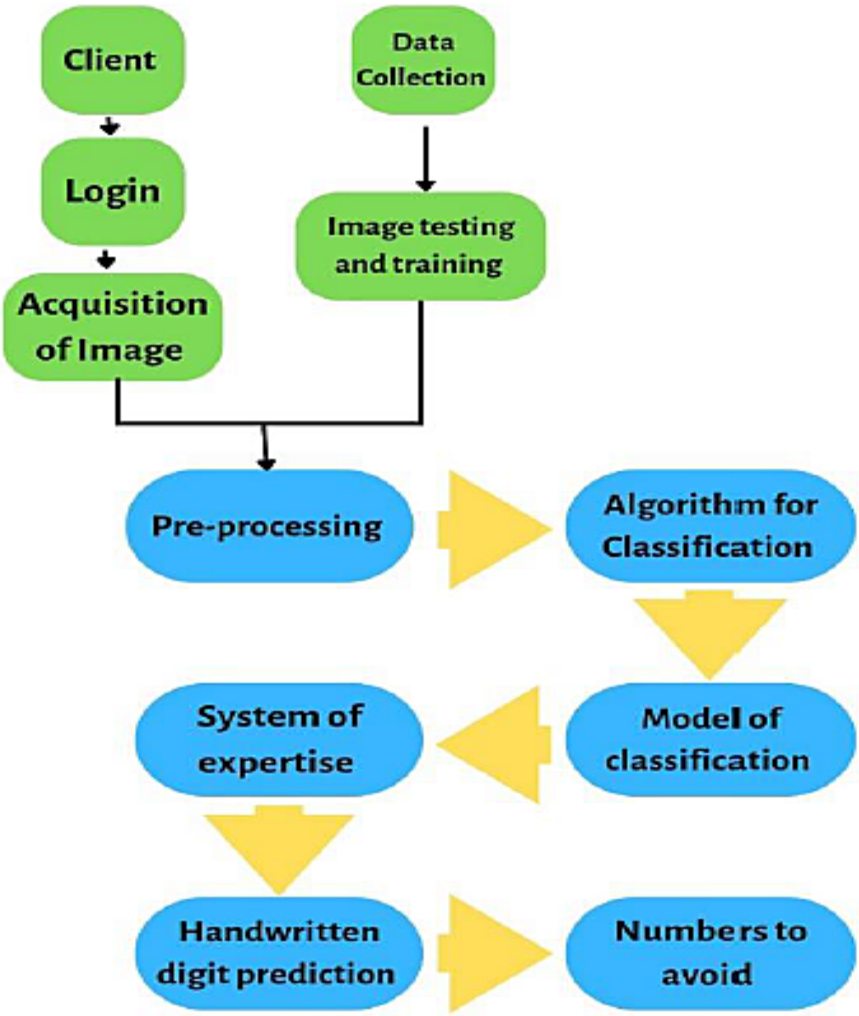
5. PROJECT DESIGN

5.1 Data Flow Diagram:

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

Example: (Simplified)





5.2 Solution and Technical Architecture:

Solution Architecture:

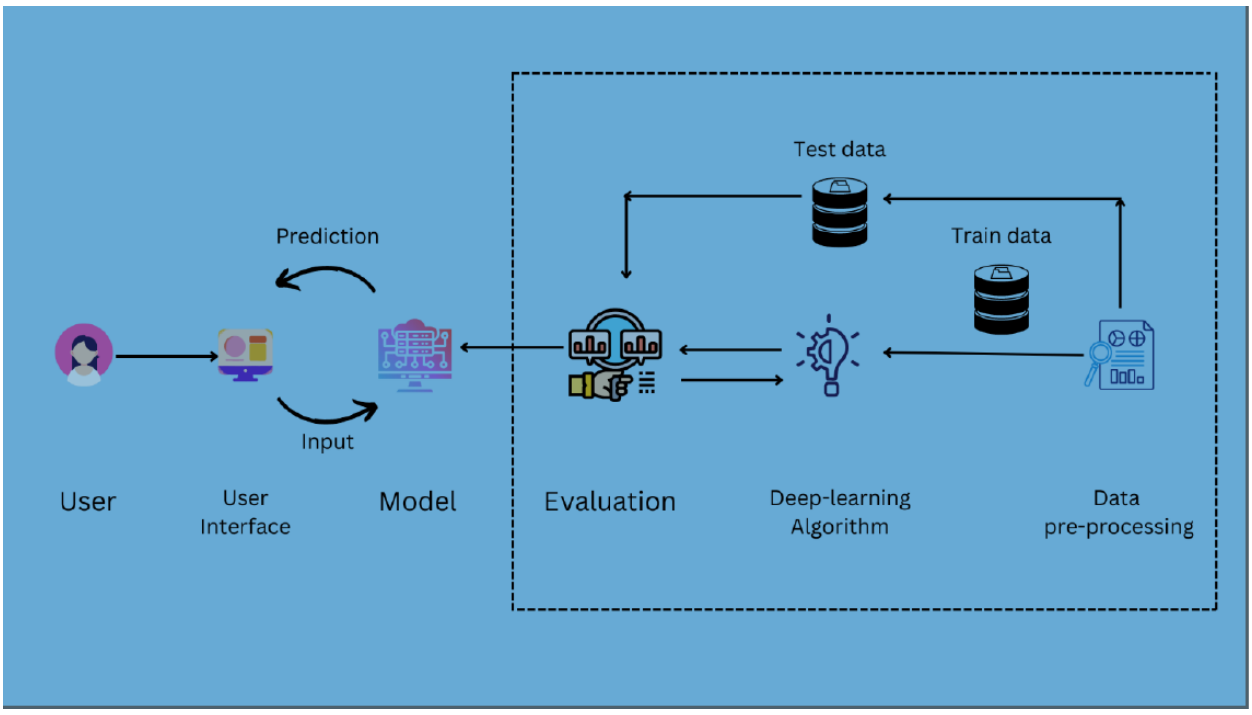
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the

software to project stakeholders.

- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:



5.3 User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my	I can access my account / dashboard	Medium	Sprint-1

			password.			
		USN-2	I can watch the instructional film as a user to use the interface.	I can see how to use this interface in a straight forward manner.	Low	Sprint-1
	Login	USN-3	I can access the application as a user by providing my email address and password.	I can sign in to the application.	Medium	Sprint-1
	Upload image	USN-4	I can add images as a user to the UI easily.	I can use it in a user friendly method.	Low	Sprint-2
		USN-5	I am allowed to enter photographs into the application for prediction as a user.	I can use the application to predict images.	Medium	Sprint-2
	Predict	USN-6	As a user, I can extract the recognised digits from photographs, digital documents, or photos.	As a user, I can retrieve the recognised digits from papers, digital documents, or photos.	High	Sprint-3
		USN-7	I can get the most output possible from the input as a user.	I can get the most out of the input	High	Sprint-3
Customer (Web user)	Accessibility	USN-8	As a web user, I can access the web application from	With internet connection,	Low	Sprint-1

			anywhere around the world.	I can use the application.		
		USN-9	I can access it for free because it is an opensource project.	I have free access to it because it is an open source.	High	Sprint-3
		USN-10	As a user, it requires no installation.	Users don't need to pay for installation.	Low	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Pre-processing	USN-1	As a user, I can load the dataset, handle missing data, scale it, and divide it into train and test sets.	2	Medium	Sindhuja.K Suguna.S
Sprint-1	Data Collection	USN-2	As a user, I can gather the dataset from a variety of sources with varying handwritings.	2	Low	Sumithasri.A Vinuppriya.B
Sprint-2	Train & test the model	USN-3	Let's use our image dataset to train our model as users	2	High	Suguna.S
Sprint-2	Save the model	USN-4	In order to forecast something for the user, the model is	1	Low	Sindhuja.K

			saved & connected with an android application or web application.			
Sprint-2	Model Building	USN-5	As a user, I will receive an application that uses an ML model to recognise handwritten digits with high accuracy	2	Medium	Vinuppriya.B
Sprint-2	Add CNN layers	USN-6	Creating the model and adding the input, hidden, and output layers to it.	1	Low	Sumithasri.A
Sprint-3		USN-7	I can view the anticipated and recognised digits in the programme as a user	2	Medium	Suguna.S Sumithasri.A
Sprint-3		USN-8	I can understand the specifics of the application's basic operation as a user.	1	Low	Sindhuja.K Vinuppriya.B
Sprint-3	Building UI Application	USN-9	I, as a user, will click an upload button to upload the image of the handwritten digit to the programme.	2	High	Sumithasri.A Vinuppriya.B Suguna.S
Sprint-4	Cloud Deployment	USN-10	I can utilise the online application as a user from anywhere and have access to it.	2	High	Sindhuja.K
Sprint-4	Train the model on IBM	USN-11	As a user, I integrate flask/Django with the scoring end point and train the model on IBM.	2	High	Vinuppriya.B

6.2 Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	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	20 Nov 2022

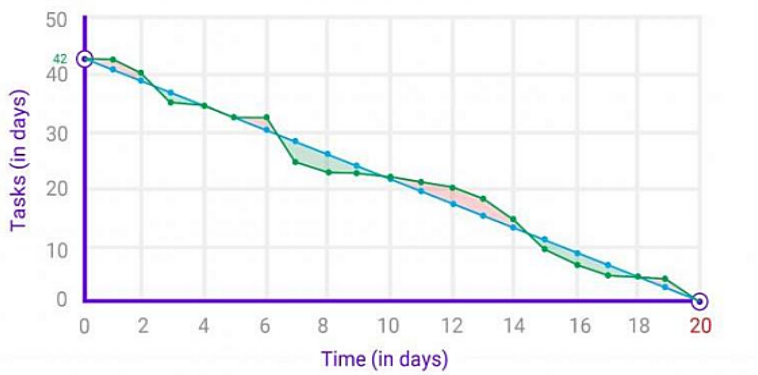
Velocity:

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).

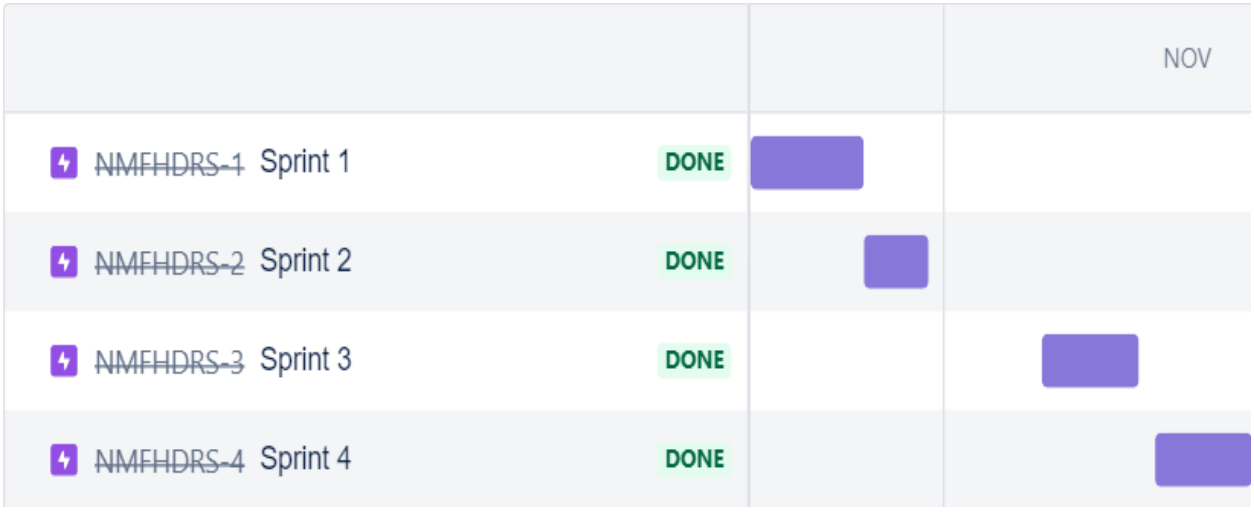
Average Velocity = Sprint Duration/ Velocity
= 20 / 6
= 3.33

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 overtime.



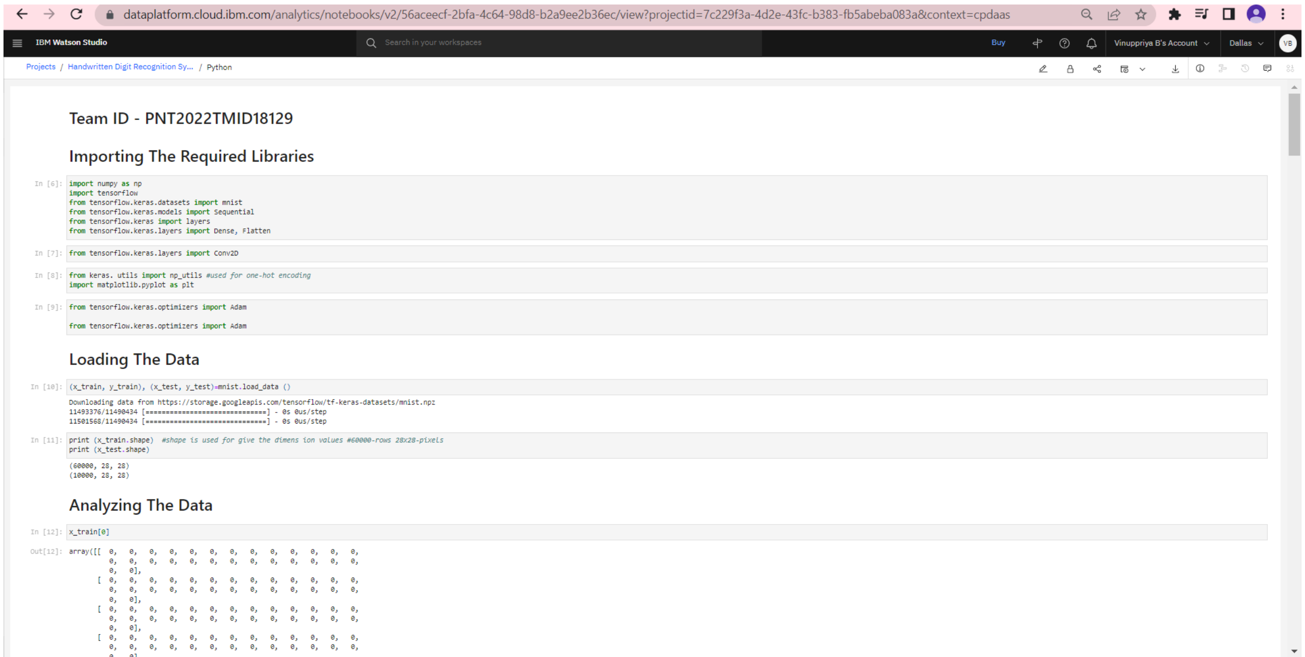
6.3 Reports From Jira:

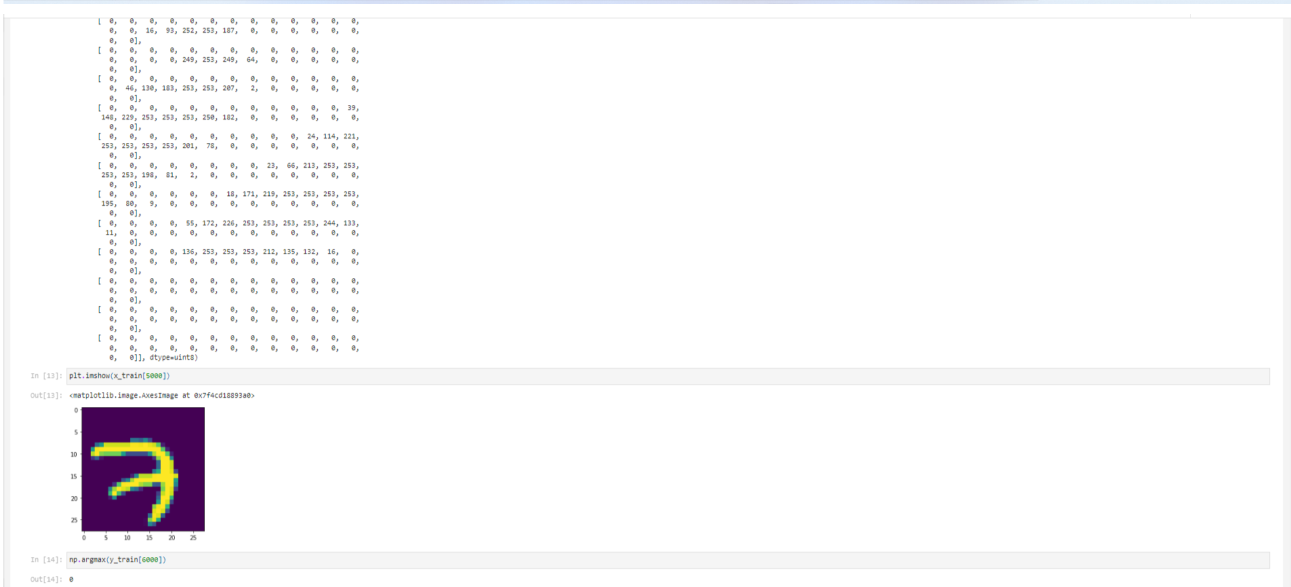


7. CODING & SOLUTIONING

7.1 Feature 1:

Training the model





Reshaping The Data

In [15]:
```
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

In [16]:
```
number_of_classes = 10
```

Applying One Hot Encoding

In [17]:
```
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

In [18]:
```
model=Sequential ()
```

In [19]:
```
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
```

In [20]:
```
model.add(Flatten())
```

In [21]:
```
model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling The Model

In [22]:
```
model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

In [23]:
```
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

Train The Model

In [24]:
```
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 163s 87ms/step - loss: 0.2704 - accuracy: 0.9478 - val_loss: 0.0974 - val_accuracy: 0.9730
Epoch 2/5
1875/1875 [==============================] - 160s 85ms/step - loss: 0.0721 - accuracy: 0.9784 - val_loss: 0.0726 - val_accuracy: 0.9776
Epoch 3/5
1875/1875 [==============================] - 162s 86ms/step - loss: 0.0505 - accuracy: 0.9839 - val_loss: 0.0939 - val_accuracy: 0.9759
Epoch 4/5
1875/1875 [==============================] - 161s 86ms/step - loss: 0.0379 - accuracy: 0.9883 - val_loss: 0.0935 - val_accuracy: 0.9759
Epoch 5/5
1875/1875 [==============================] - 162s 86ms/step - loss: 0.0285 - accuracy: 0.9907 - val_loss: 0.1100 - val_accuracy: 0.9784
```

Out[24]:
```
<keras.callbacks.History at 0x7f4cd1ec7880>
```

Observing The Metrics

In [25]:
```
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.11000255495303983, 0.9783999919891357]
```

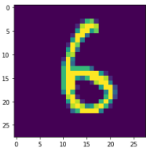
Test The Model

In [26]:
```
prediction=model.predict(x_test[3000:3001])
print(prediction)
```

```
[[1.5941167e-17 2.9809187e-17 2.2525793e-18 3.6060331e-16 2.6896594e-16
  2.9916810e-06 9.9999702e-01 3.2172763e-28 1.5404786e-12 3.6017933e-19]]
```

In [27]:
```
plt.imshow(x_test[3000])
```

Out[27]:
```
<matplotlib.image.AxesImage at 0x7f4cd05cef40>
```



In [28]:
```
print(np.argmax(prediction, axis=1))
```

```
[6]
```

In [29]:
```
np.argmax(y_test[3000:3001])
```

Out[29]:
```
6
```

Save The Model

In [30]:
```
model.save('models/mnistCNN.h5')
```

In [31]:
```
cd models
```

```
/home/wsuser/work/models
```

In [32]:
```
!tar -zcvf handwritten-digit-recognition-model_new.tgz mnistCNN.h5
```

```
mnistCNN.h5
```

7.2 Feature 2:

Testing the saved model

Test the Model

```
In [80]: from tensorflow.keras.models import load_model
         from keras.preprocessing import image
         from PIL import Image
         import numpy as np

In [81]: model = load_model("mnistCNN.h5")

In [82]: import os, types
         import pandas as pd
         from botocore.client import Config
         import ibm_boto3

         def __iter__(self): return 0

         # @hidden_cell
         # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
         # You might want to remove those credentials before you share the notebook.
         cos_client = ibm_boto3.client(service_name='s3',
             ibm_api_key_id='g0hL7_-weHOUF0vpK2Rd9t3iicWnzSKcF593KxxKx4Eg',
             ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
             config=Config(signature_version='oauth'),
             endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

         bucket = 'handwrittendigitrecognitionsystem-donotdelete-pr-bnrlkchvymusdw'
         object_key = 'mnist-dataset-3.png'

         streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

         # Your data file was loaded into a botocore.response.StreamingBody object.
         # Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
         # ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
         # pandas documentation: http://pandas.pydata.org/

In [83]: img = Image.open(streaming_body_1).convert("L")
         img = img.resize( (28,28) )

In [84]: img
Out[84]: 

In [85]: im2arr = np.array(img)
         im2arr = im2arr.reshape(1, 28, 28, 1)

In [86]: pred = model.predict(im2arr)
         print(pred)

         [[9.2903363e-08 3.4811689e-14 2.6743678e-11 9.9999976e-01 1.8006752e-14
           1.2555390e-07 1.1633670e-13 5.5902119e-12 2.7492250e-13 4.9024586e-11]]

In [87]: print(np.argmax(pred, axis=1))
         [3]
```

8. TESTING

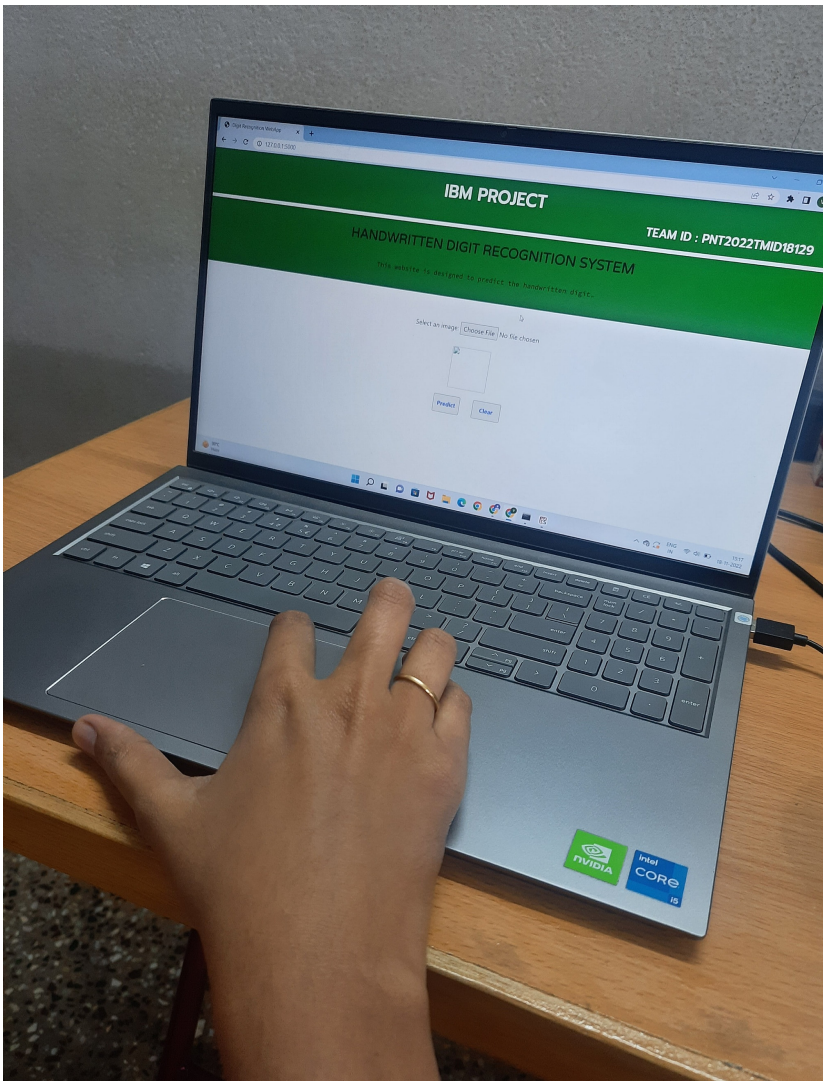
8.1 Test Cases:

Team ID - PNT2022TMID18129

```
In [18]: from tensorflow.keras.models import load_model
         from PIL import Image
         import numpy as np

In [19]: model = load_model(r'C:\Users\VINUPPRIYA\models\mnistCNN.h5')

In [24]: img = Image.open(r'C:\Users\VINUPPRIYA\OneDrive\Desktop\mnist-dataset-5.png').convert("L")
         img = img.resize( (28, 28) )
         im2arr = np.array(img)
         im2arr = im2arr.reshape(1, 28, 28, 1)

In [25]: img
Out[25]: 

In [26]: y_pred = model.predict(im2arr)
         print(y_pred)

         1/1 [==============================] - 0s 24ms/step
         [[5.0599657e-17 7.7438300e-18 1.2412449e-21 5.9922553e-12 1.5130368e-17
           1.0000000e+00 8.2508249e-23 1.9590003e-18 3.4609075e-16 1.6959033e-11]]

In [27]: import numpy as np
         print(np.argmax(y_pred, axis=1))

         [5]
```

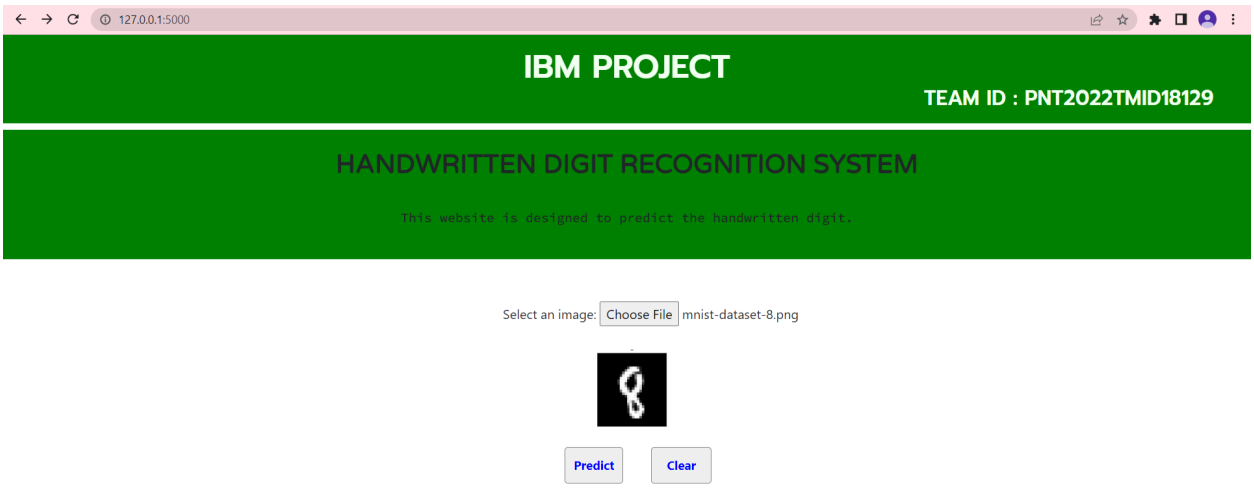
8.2 User Acceptance Testing:

Using the website



9. RESULTS

9.1 Performance Metrices:



10. ADVANTAGES AND DISADVANTAGES

Advantages: Handwritten digit recognition system is used to recognize the digits that are wriiten by hand. Handwritten didt recognition system is used to visualize artificial neural networks. It can be widely used in the automatic processing of bank

cheques, postal addresses, in mobile phones etc.

Disadvantages: It is not done in real time as a person writes and therefore not appropriate for immediate text input.

11. CONCLUSION

In this project, the Handwritten Digit Recognition using Deep learning methods has been implemented. The most widely used Machine learning algorithms CNN has been trained and tested on the MNIST dataset. With extensive testing using the MNIST data, the current function suggests the role of various hyperparameters. We also confirmed that a good adjustment of hyper parameters is important in improving the performance of Convolutional Neural Network. Utilizing this deep learning technique, a high amount of accuracy can be obtained. This model is able to achieve a recognition rate of 99.07% accuracy and is significantly identifying real world images as well. Digital recognition is an excellent model for learning about neural networks and provides a great way to develop the most advanced methods of deep learning.

12. FUTURE SCOPE

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems. In future, the application of these models lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people, we can use these models in hospitals application for detailed medical diagnosis, treatment and monitoring the patients, we can use it in surveillances system to keep tracks of the suspicious activity under the system, in fingerprint and retinal scanners, database filtering applications, Equipment checking for national forces and many more problems of both major and minor category. The advancement in this field can help us create an environment of safety, awareness and comfort by using these models in day-to-day application and high-level application (i.e., corporate level or Government level). Application-based on artificial intelligence and machine learning is the future of the technological world because of their absolute accuracy and advantages over many major problems.

13. APPENDIX

SOURCE CODE:

app.py

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory

UPLOAD_FOLDER = r'C:\Users\VINUPPRIYA\OneDrive\Desktop\Final Deliverables\Final Code\A
novel method for handwritten digit recognition system\flask_app\uploads'

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_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")
        img = img.resize((28, 28))

        im2arr = np.array(img)
```

```
        im2arr = im2arr.reshape(1, 28, 28, 1)

        pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1)
        return render_template('prediction.html', num=str(num[0]))


if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

index.html

```
<html>
<head>
  <title>Digit Recognition WebApp</title>

  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&displ
ay=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
```

```
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>

</head>

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

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

<body>

  <h1 class="welcome">IBM PROJECT
  <div id="team_id">TEAM ID : PNT2022TMID18129</div>
  </h1>
  <section id="title">
    <h4 class="heading">HANDWRITTEN DIGIT RECOGNITION SYSTEM</h4>
    <br><br>
      <p>
        This website is designed to predict the handwritten digit.
      </p>
  </section>

  <section id="content">
       <div class="leftside">
       <form action="/predict" method="POST" enctype="multipart/form-data">
```

```
      <label>Select an image:</label>
      <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
        <img id="frame" src="" width="100px" height="100px"/>
        <div class="buttons_div">
          <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
          <button type="button" class="btn btn-dark" id="clear_button">  Clear
 </button>
        </div>
      </form>
      </div>
 </section>
</body>
</html>
```

prediction.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Prediction</title>
</head>

<style>
   body{
    background-image: url('static/images/index6.jpg');
    background-repeat: no-repeat;
    background-size: cover;
   }

   #rectangle{
    width:400px;
    height:150px;
    background-color: #5796a5;
    border-radius: 25px;
```

```
    position:absolute;
    top:25%;
     left:50%;
    transform:translate(-50%,-50%);
   }

  #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 15%;
  color: white;
   }
</style>
<body>
   <div id="rectangle">
      <h1 id="ans">Predicted Number : {{num}}</h1>
   </div>
</body>
</html>
```

style.css

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: blue;
}

#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#content{
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
```

```
}

.welcome{
   text-align: center;
   position: relative;
   color: honeydew;
   background-color: green;
   padding-top: 1%;
   padding-bottom: 1%;
   font-weight: bold;
   font-family: 'Prompt', sans-serif;
}

#team_id{
   text-align: right;
   font-size: 25px;
   padding-right: 3%;
   background-color: green;
}

#predict_button{
  margin-right: 15px;
  color: blue;
  font-weight: bold;
}

#prediction_heading{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#result{
  font-size: 5rem;
}

#title{
  padding: 1.5% 15%;
  margin: 0 auto;
  text-align: center;
  background-color: green;
}
```

```
.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: none;
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}

.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;
  /* padding-left: 10%; */
}

#frame{
  margin-right: 10%;
}

.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;
  /* padding-left: 10%; */
}
```

```
p{
  font-family: 'Source Code Pro', monospace,sans-serif;
  margin-top: 1%;   }

@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}
```

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-44105-1660722167

PROJECT DEMO LINK: https://youtu.be/ZLoObvddmSs