**PROJECT REPORT**

# **A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM**

**Team ID: PNT2022TMID37246**

**TEAM MEMBERS:**

**AISHWARYA.H– 311619104004**

**DHARANISHREE.A – 311619104016**

**HARITHA.S– 311619104023**

**KAVYA.R- 311619104031**

**MONICA -311619104302**

# Abstract

Hand Written Digit recognition is also remarkable an important issue. As handwritten digits are not a same size, thickness, position and direction, in this case by the way, various difficulties should be considered find the handwritten digital recognition problem. I unique and a variety of creative styles for different people moreover have an influence on the model as well the presence of digits. It is a strategy to see again edit written digits. It has a wide variety of applications, for example, scheduled bank checks, post offices and tax documents and so on. The purpose of this project is to use the classification algorithm to identify handwritten digits. Background results are probably the most widely used Machine Learning Algorithms such as SVM, KNN and RFC and in-depth reading calculations like CNN multilayer using Keras and Theano and Tensorflow. Using these, 98.70% accuracy was used by CNN (Keras + Theano) compared to 97.91% using SVM, 96.67% using KNN, 96.89% using RFC was obtained.

INTRODUCTION

1.1 Project Overview:

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. This image is analyzed by the model and the detected result is returned on to UI. The aim of a handwriting recognition system is to convert handwritten characters into machine readable formats. Handwritten digit recognition has not only professional and commercial applications, but also has practical application in our daily life and can be of great help to the visually impaired. It also helps us to solve complex problems easily thus making our lives easier. Handwritten digit recognition has gained so much popularity from the aspiring beginner of machine learning and deep learning to an expert who has been practicing for years.

1.2 Purpose:

Nowadays the whole world is a shift in the digital world. Images of handwritten digits as 10 digits (09). Handwritten digits from the MNIST database are already famous among the community for many recent decades now, as decreasing the error rate with different classifiers and parameters. Digit recognition system is the working of a machine to train itself or recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (say tax forms) and so on. The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. This problem is faced more when many people write a single digit with a variety of different handwritings.

2. LITERATURE SURVEY

2.1 Existing System:

In most of the existing systems recognition accuracy is heavily dependent on the quality of the input document. In handwritten text adjacent characters tend to be touched or overlapped. Therefore, it is essential to segment a given string correctly into its character components. In most of the existing segmentation algorithms, human writing is evaluated empirically to deduce rules. But there is no guarantee for the optimum results of these heuristic rules in all styles of writing. Moreover, handwriting varies from person to person and even for the same person it varies depending on mood, speed etc. This requires incorporating artificial neural networks, hidden Markov models and statistical classifiers to extract segmentation rules based on numerical data.

2.2 References:

1. H. Zeng,(2020)An Off-line Handwriting Recognition Employing Tensorflow. International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE).
2. A. Beltrán and S. Mendoza, “Efficient algorithm for real-time handwritten character recognition in mobile devices “,2011, 8th International Conference on Electrical Engineering, Computing Science and Automatic Control, 2011, pp. 1-6, doi: 10.1109/ICEEE.2011.6106583.
3. H. A. Shiddieqy, T. Adiono and I. Syafalni, \"Mobile Client-Server Approach for Handwriting Digit Recognition\'\',2019 International Symposium on Electronics and Smart Devices (ISESD), 2019, pp. 1-4, doi: 10.1109/ISESD.2019.8909448.
4. R. Vaidya, D. Trivedi, S. Satra and P. M. Pimpale, \"Handwritten Character Recognition Using DeepLearning\",2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 772-775, doi:

10.1109/ICICCT.2018.8473291.

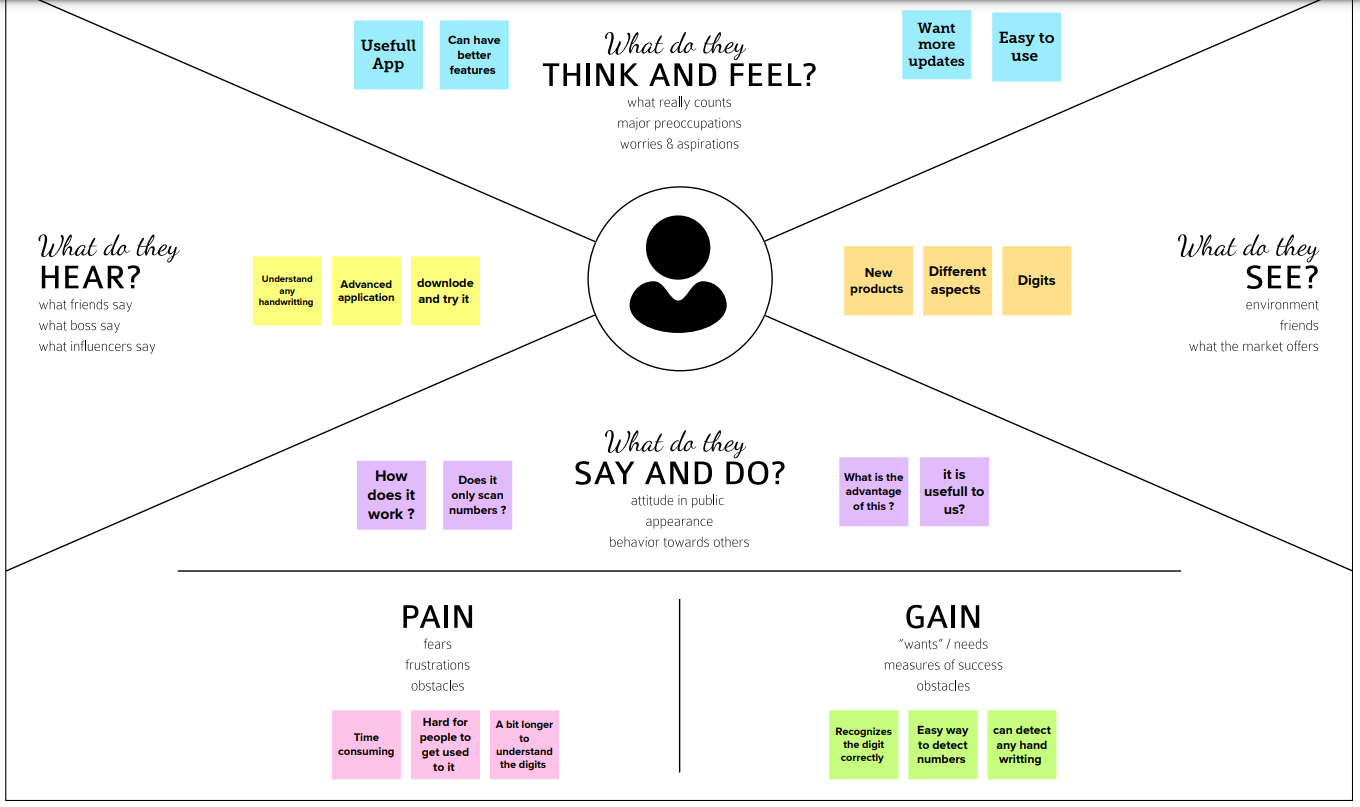
1. H. Du, P. Li, H. Zhou, W. Gong, G. Luo and P. Yang, "Word Recorder: Accurate Acousticbased Handwriting Recognition Using Deep Learning, "IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, 2018, pp. 1448-1456, doi: 10.1109/INFOCOM.2018.8486285.

Fig - 7: Image capturing Fig -8: Printed text output © 2021, IRJET | Impact Factor value: 7.529. 2.3 Problem Statement:

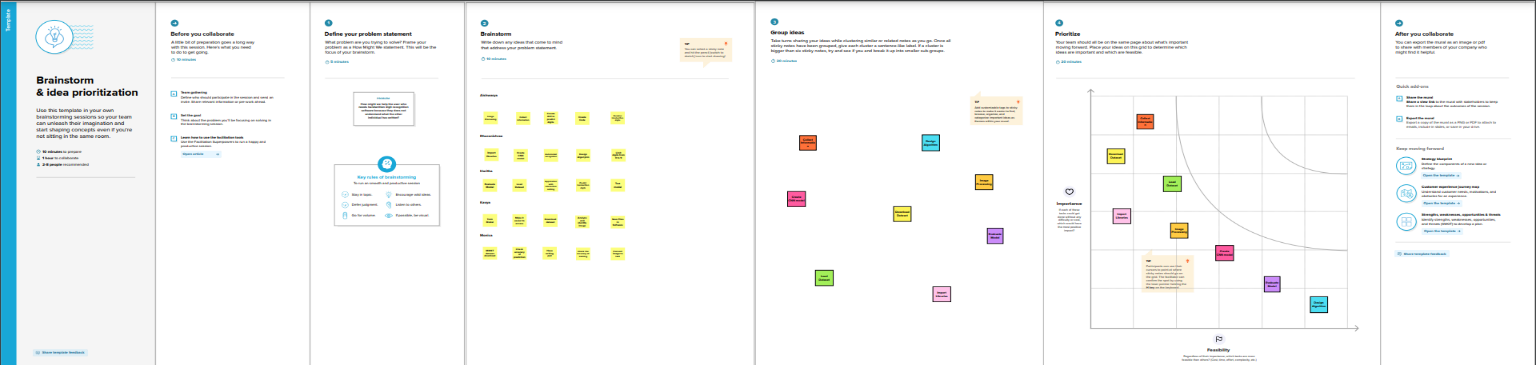
The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual’s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

### 3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:



Team Member 1:

* Focus on one feature at a time
* Clean and Simple UI
* No bulk commits in GitHub
* Analyze existing solutions
* Compare with requirements more often
* Clean customer navigations

Team Member 2:

* Stick together as a team - frequent discussions
* Break into pieces - discuss, code, test and deploy
* Each phase - working model to beta test
* Use project management tools
* Analyse the data source and explore other sources for future use
* Bring drag and drop options to upload images + preview

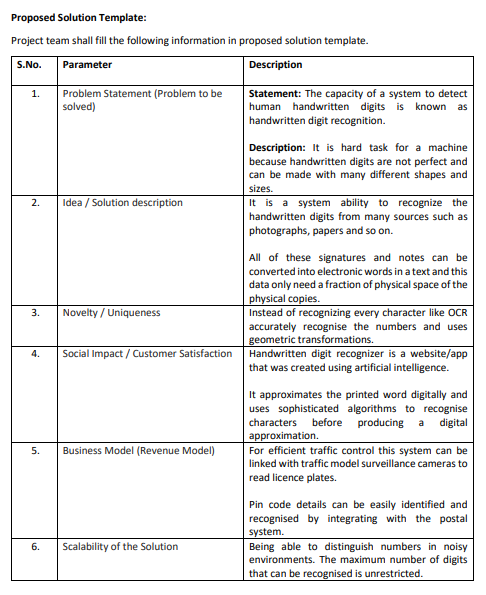
Team Member 3:

* Discussion before every phase and writing code
* Get feedback from other friends as a user
* Develop while learning approach
* Analyze user requirements and thoughts
* Use collaborative todo apps
* Proper project structure - develop layout and logic subsequently

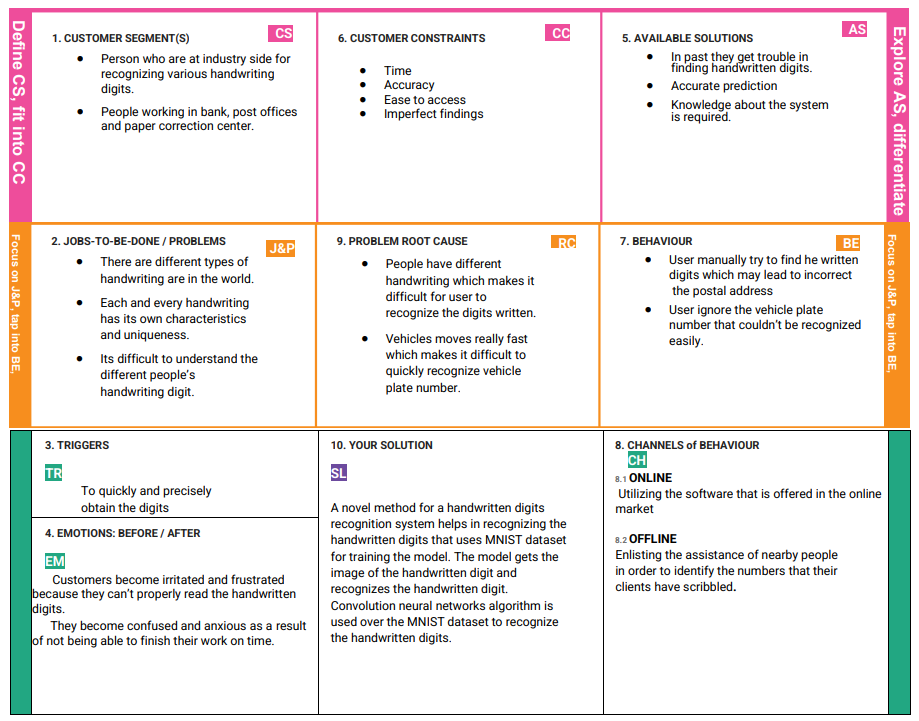
Team Member 4:

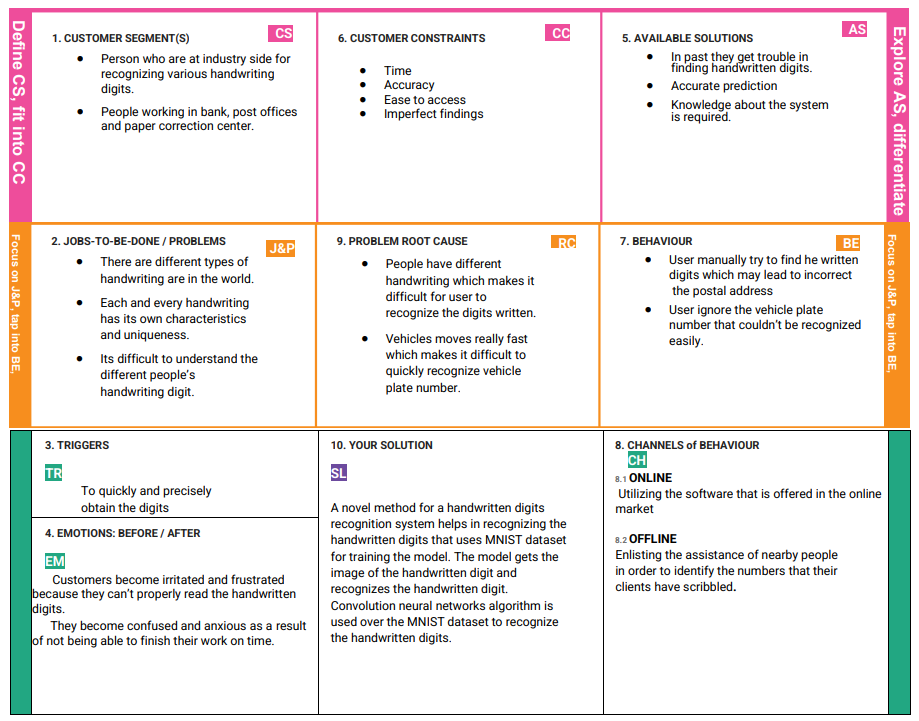
* Think like a user - then design
* No unwanted features - quality first
* clean and commented code - avoid inline comments
* Work in branches, and give pull request
* Consider scalability and continuous development right from the start
* Instead of developing all at once and deploying, do continuous development and deployment

3.3Proposed Solution:



3.4Problem Solution fit:





4. REQUIREMENT ANALYSIS

* 1. Functional requirement:

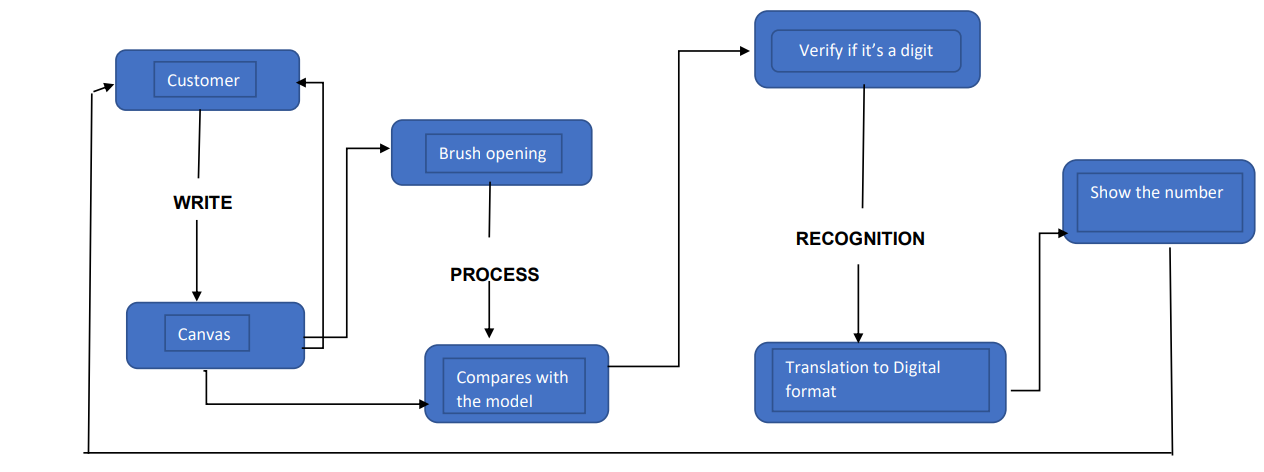
|  |  |
| --- | --- |
| IR No: | Functional Requirement and description: |
| FR-1 | Image Data: Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc., and classify them into 10 predefined classes (0-9). this has been a topic of boundless-research in the field of deep learning. |
| FR-2 | Website: Web hosting makes the files that comprise a website (code, images, etc.) available for viewing online. Every website you've ever visited is hosted on a server. The amount of space allocated on a server to a website depends on the type of hosting. the main types of hosting are shared, dedicated, VPS.. |
| FR-3 | Digit\_Classifier\_Model: Use the MNIS database of handwritten digits to train a convolutional network to predict the digit given an image. First obtain the training and validation data. |
| FR-4 | MNIS dataset: the MNIST dataset is an acronym that stands for the Modified  National  Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9. |
| FR-5 | Cloud: The cloud provides a number of IT services such as servers, |
|  | databases, software, virtual storage, and networking, among others. In layman's terms, Cloud Computing is defined as a virtual platform that allows you to store and access your data over the internet without any limitations. |

* 1. Non-Functional requirements:

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

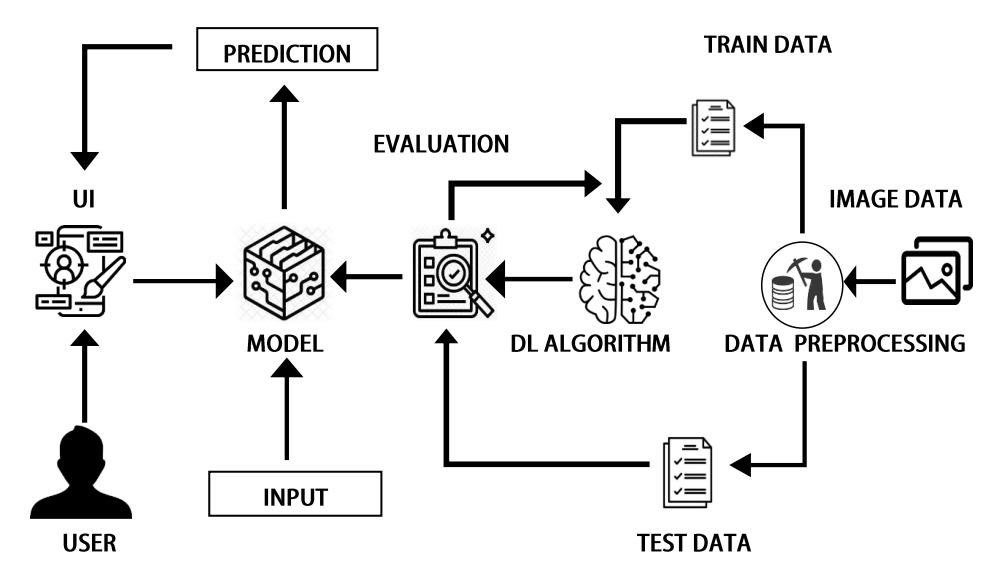
## 5. PROJECT DESIGN

5.1 Data Flow Diagrams:



5.2 Solution & Technical Architecture:

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

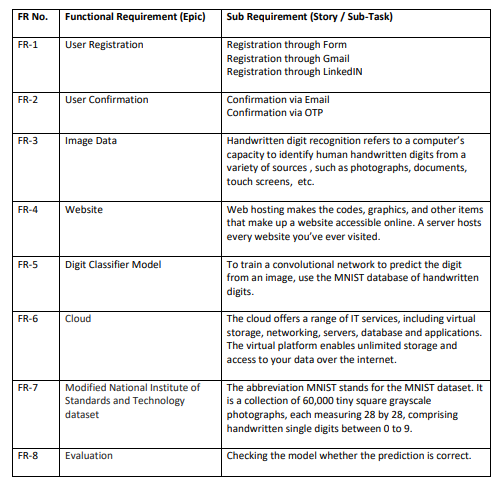


Components & Technologies:

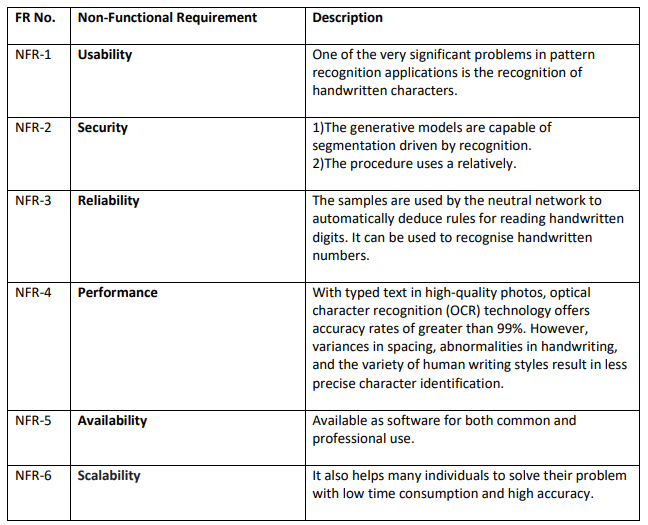
|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology |
| 1. | User Interface | How user interacts with application e.g., Mobile Application | HTML, CSS, JavaScript / AngularJS / Node Red. |
| 2. | Application Logic-1 | Logic for a process in the application | Java / Python |
| 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 AI | IBM DB2. |
| 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. | IoT Model | Purpose of AI Model is for integrating the sensors with a user interface. | IBM AI Platform |
| 10. | Infrastructure (Server / AI) | Application Deployment on Local  System / AI Local Server  Configuration AI Server  Configuration | Local, Kubernetes, etc. |

5.3 User Stories:

Functional Requirements:

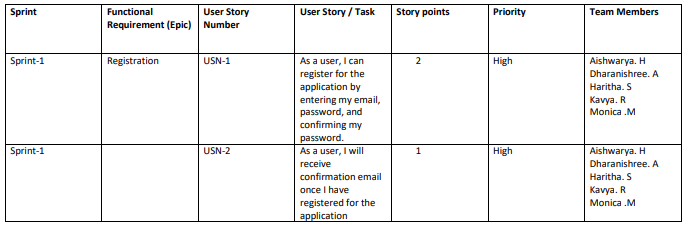


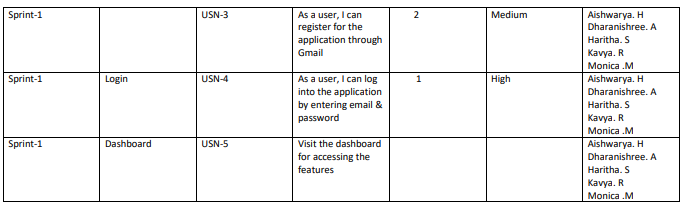
Non-Functional Requirements:



6. PROJECT PLANNING&SCHEDULING

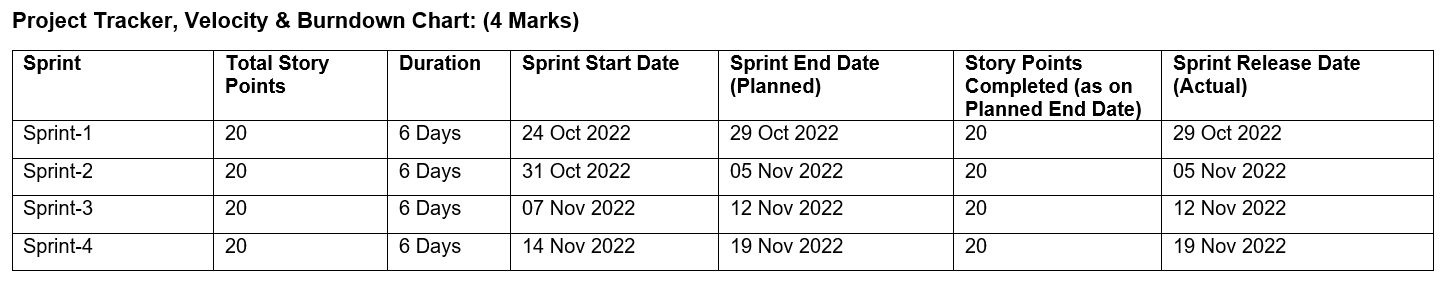
6.1 Sprint Planning & Estimation:





6.2 Sprint Delivery Schedule:

Project Planning Phase Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

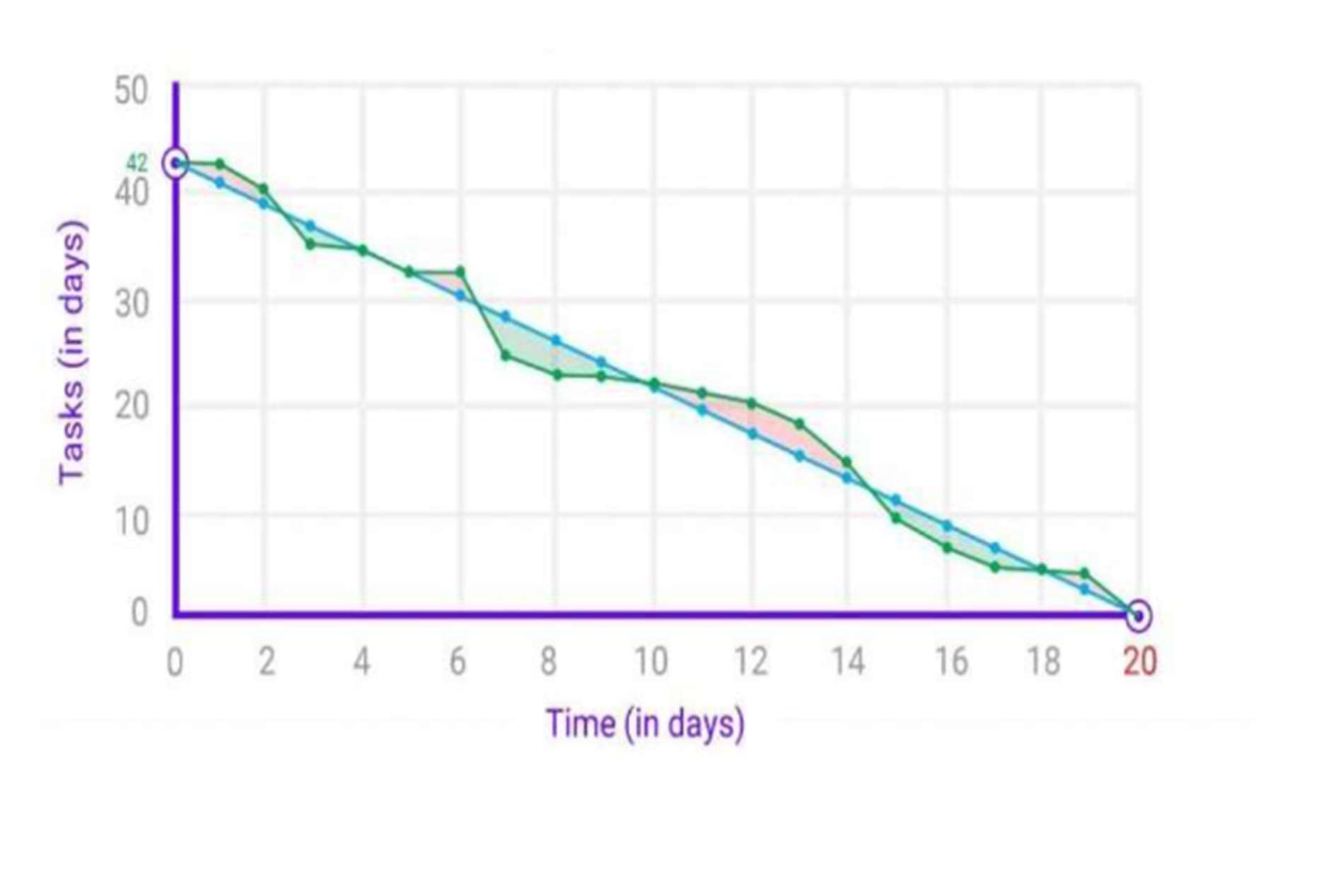


Velocity: Imagine we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint).

Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day) AV=sprint duration/Velocity =20/6=3.33

Burndown Chart:

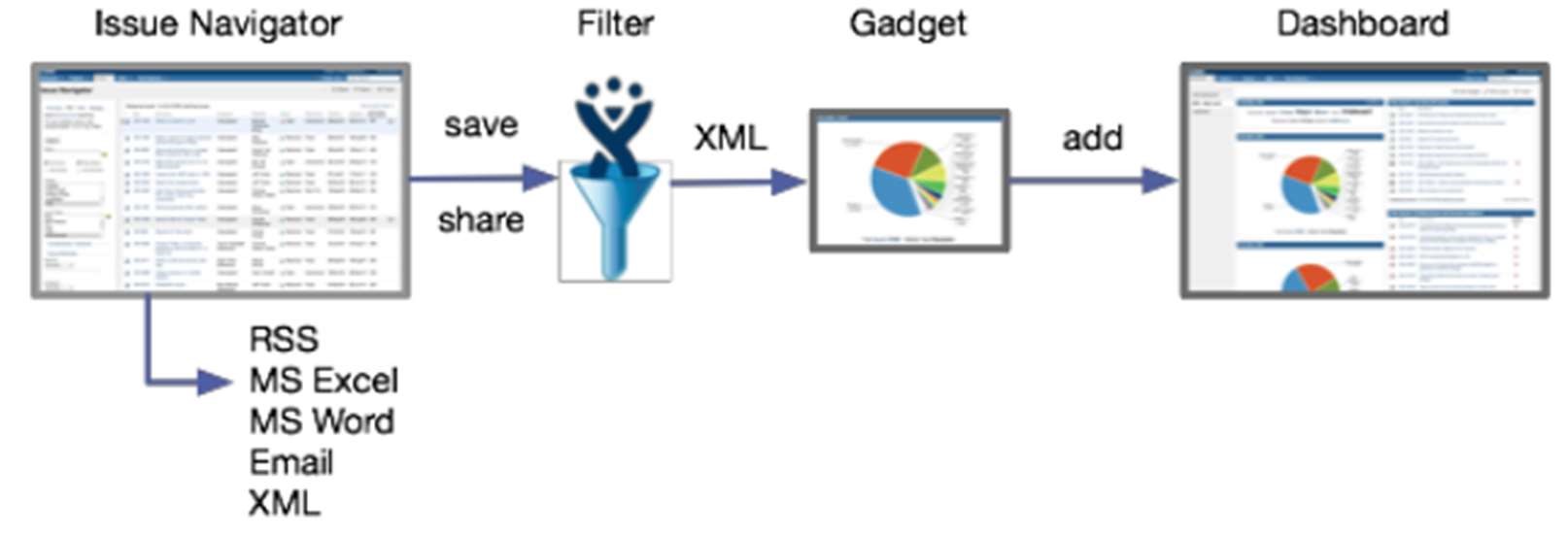
A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 Reports from JIRA:

JIRA offers reporting in a number of different formats. Project reports that are available from the home screen of the selected project, Gadgets that can be added and arranged in Dashboards and for each filter, the issue navigator offers various output formats that can be used in third party reporting software. Additionally, we will mention some advanced methods that customers have been using. In JIRA, a project will automatically offer standard reports available to the user without any necessary configuration. These standard reports comprise a wide range of reporting applications such as time tracking, workload and also abstract reports like Pie Charts that can be used in various ways.





## 7. CODING & SOLUTIONING

7.1 Feature 1:

Index.html

<html>

<head>

<title>Digit Recognition WebApplicationS</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&display=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='C:\IBM Project') }}">

<!-- 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>

<style>

#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: greenyellow;

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%;

}

#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;

}

.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%;

}

}

</style>

<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 : PNT2022TMID37246</div>

</h1>

<section id="title">

<h4 class="heading">Handwritten Digit Recognition Website</h4>

<br><br>

<p>

The website is designed to predict the handwritten digit(0-9).

</p>

<p>

Handwriting recognition is one of the compelling research works going on because every individual in this world

has their own style of writing. It is the capability of the computer to identify and understand

handwritten digits or characters automatically. Because of the progress in the field of science and technology,

everything is being digitalized to reduce human effort.</p>

<br>

<p> Hence, there comes a need for handwritten digit recognition in many real-time applications.

MNIST data set is widely used for this recognition process and it has 70000 handwritten digits.

We use Artificial neural networks to train these images and build a deep learning model.

Web application is created where the user can upload an image of a handwritten digit.

This image is analyzed by the model and the detected result is returned on to UI</p>

</section>

<section id="content">

<div class="leftside">

<form action="/predict" method="POST" enctype="multipart/form-data">

<label>Select a 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">&nbsp Clear &nbsp</button>

</div>

</form>

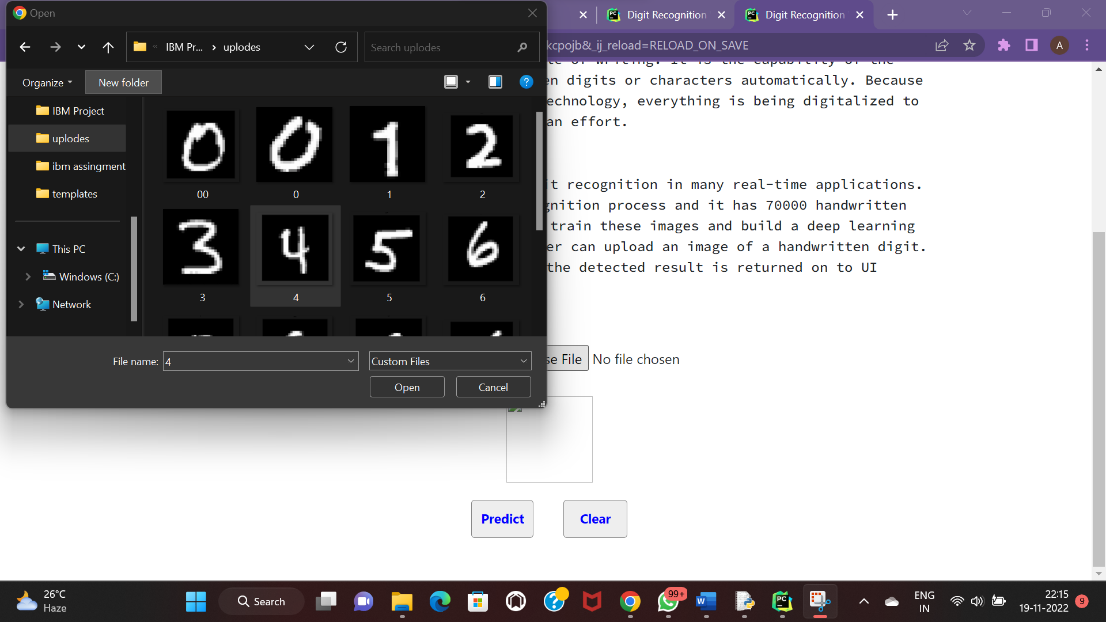
</div>

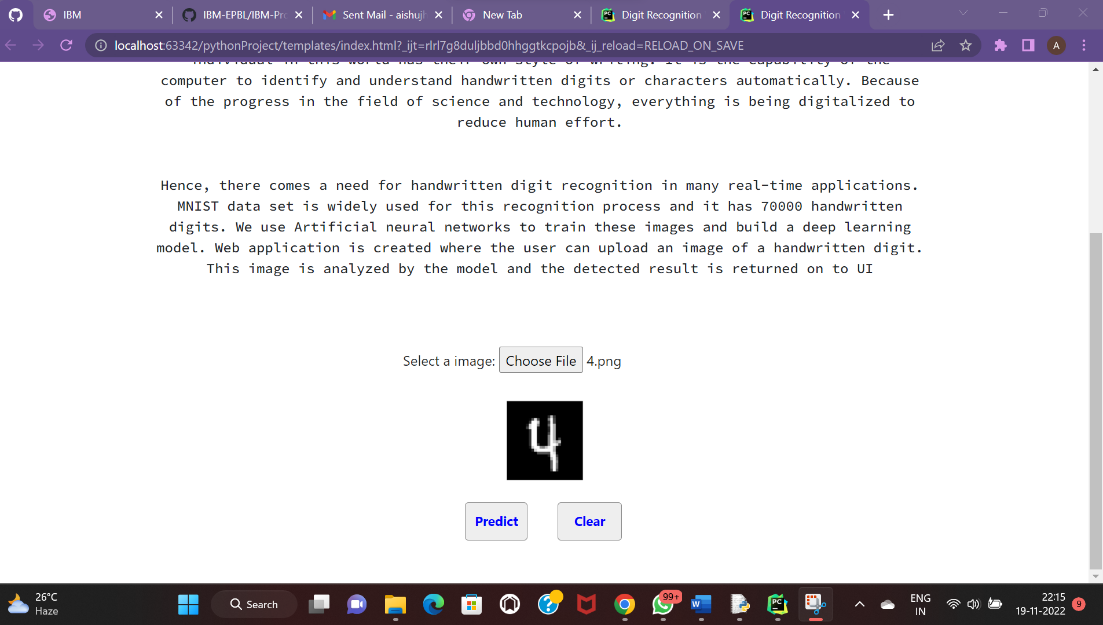
</section>

</body>

</html>







7.2 Feature 2:

Predict.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Prediction</title>

</head>

<style>

body{

background-image: url('C:/IBM Project/templates/image/num5.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>

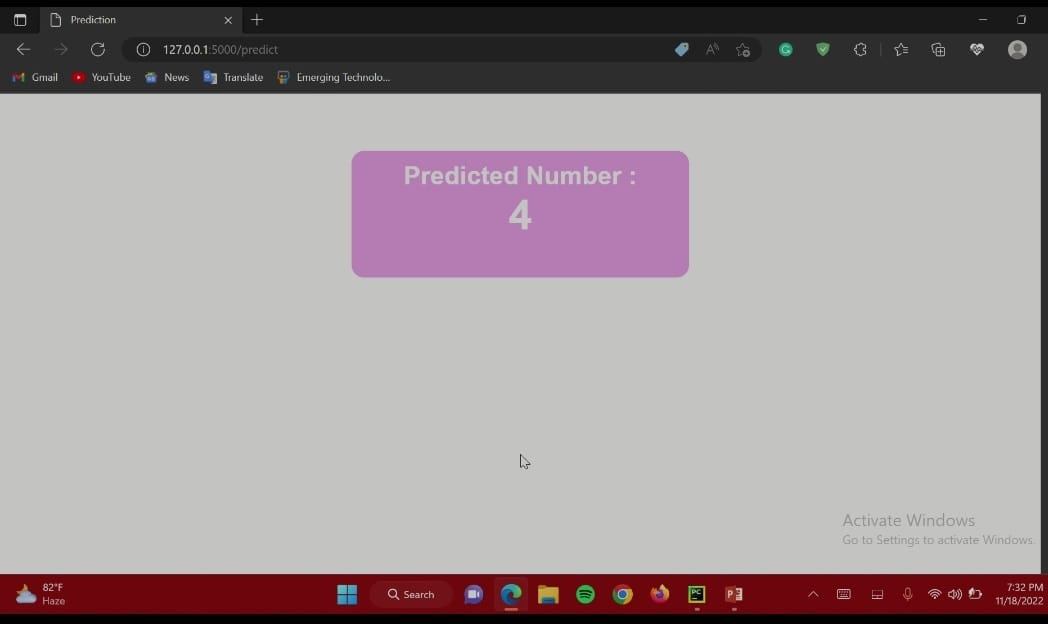
<div id="rectangle">

<h1 id="ans">Predicted Number : {{num}}</h1>

</div>

</body>

</html>



## 8. TESTING

8.1 Test Cases:

Importing the Required Libraries

import numpy

import matplotlib.pyplot as plt from keras.utils import np\_utils from tensorflow.keras.datasets i mport mnist from tensorflow.keras.models

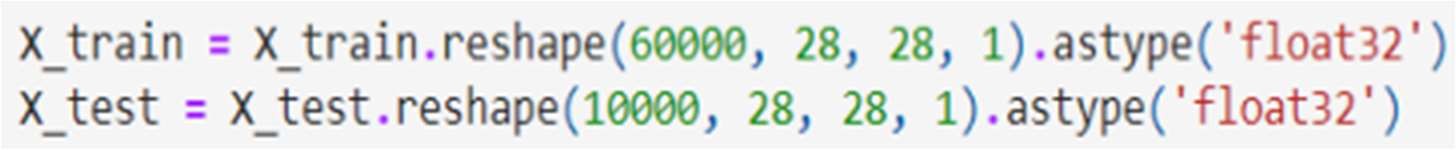
import Sequential from tensorflow.keras.layers import Conv2D, Dense, Flatten from tensorflow.keras.optimizers import Adam import numpy as np import pandas as pd from tensorflow.keras.models

import load\_model from PIL import Image, ImageOp

Loading the Data:

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

Reshaping the Data:

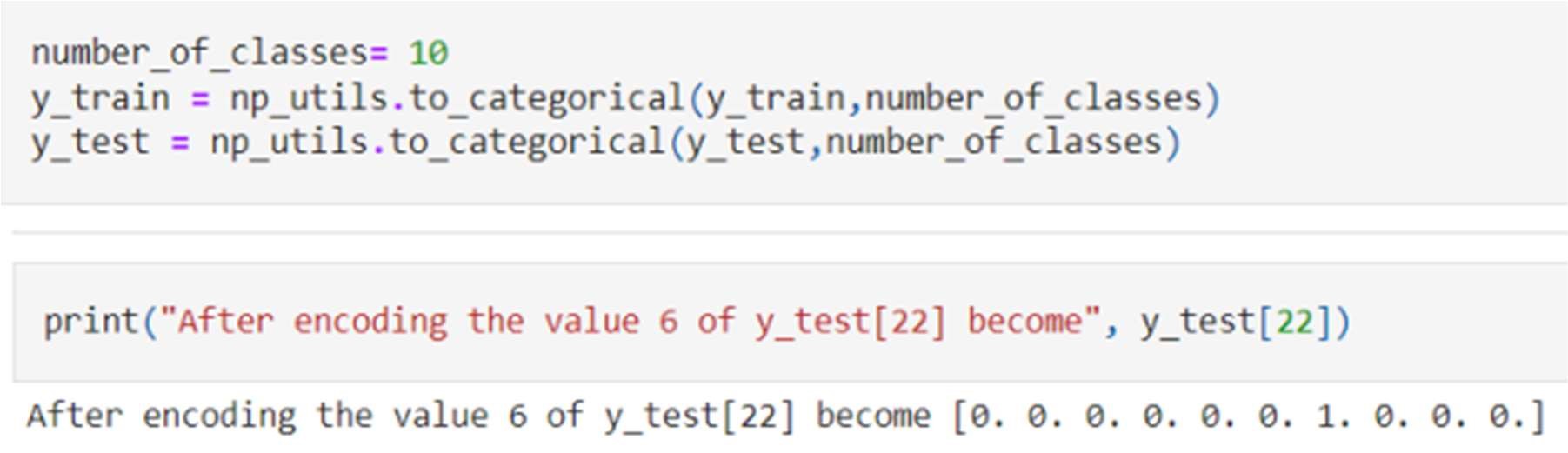


Analyzing the Data: print(X\_train.shape) print(X\_test.shape) (60000, 28, 28)

(10000, 28, 28) X\_train[0]

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]

Apply One Hot Encoding:



8.2 User Acceptance Testing:

Unit Testing:

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

Integration Testing:

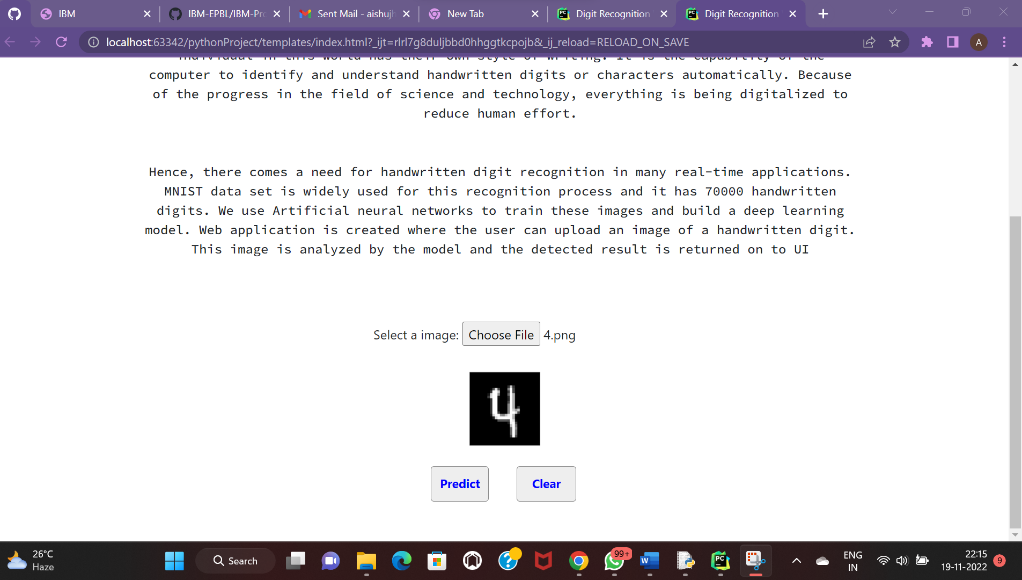
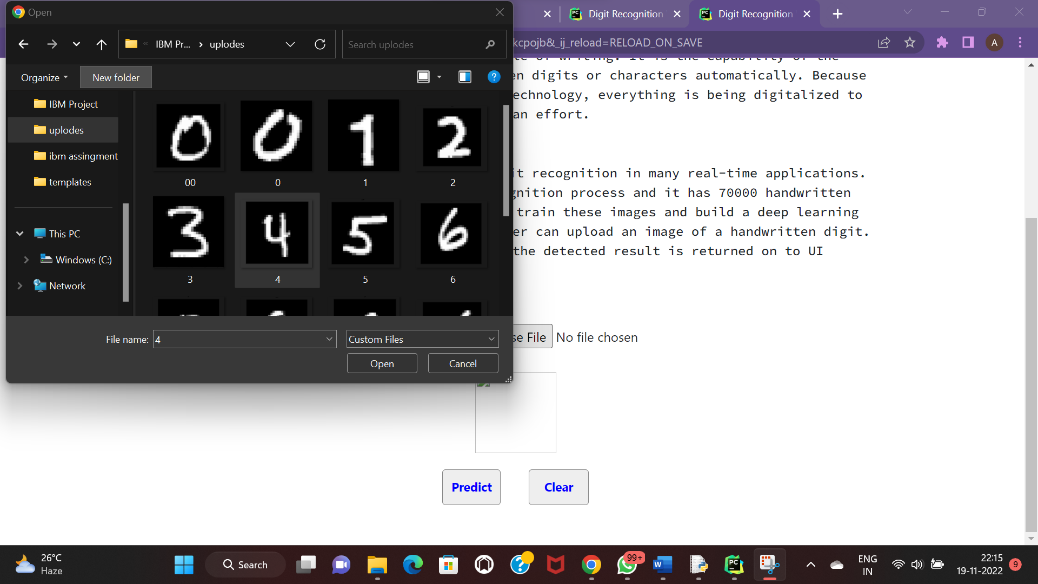
Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

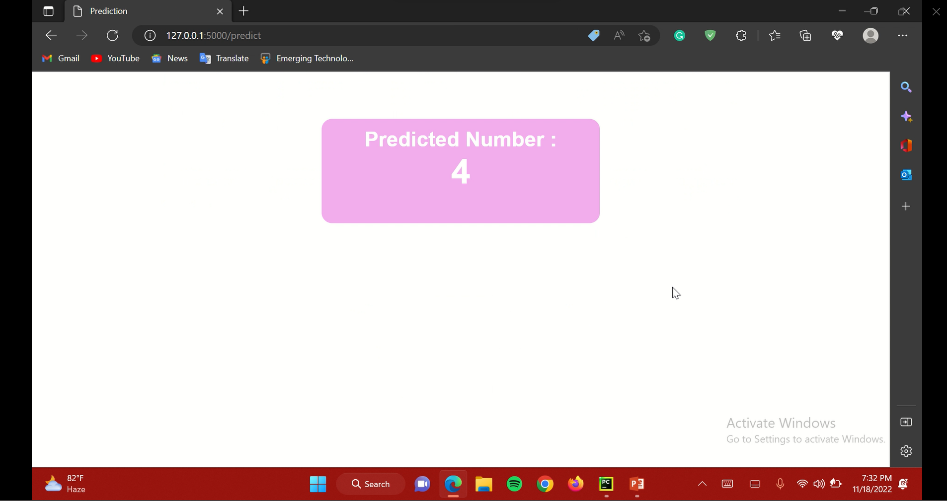
Validation Testing:

To uncover functional errors, that is, to check whether functional characteristics confirm to specification or not specified.

System Testing:

Once individual module testing completed, modules are assembled to perform as a system. Then the top down testing, which begins from upper level to lower level module testing, has to be done to check whether the entire system is performing satisfactorily.





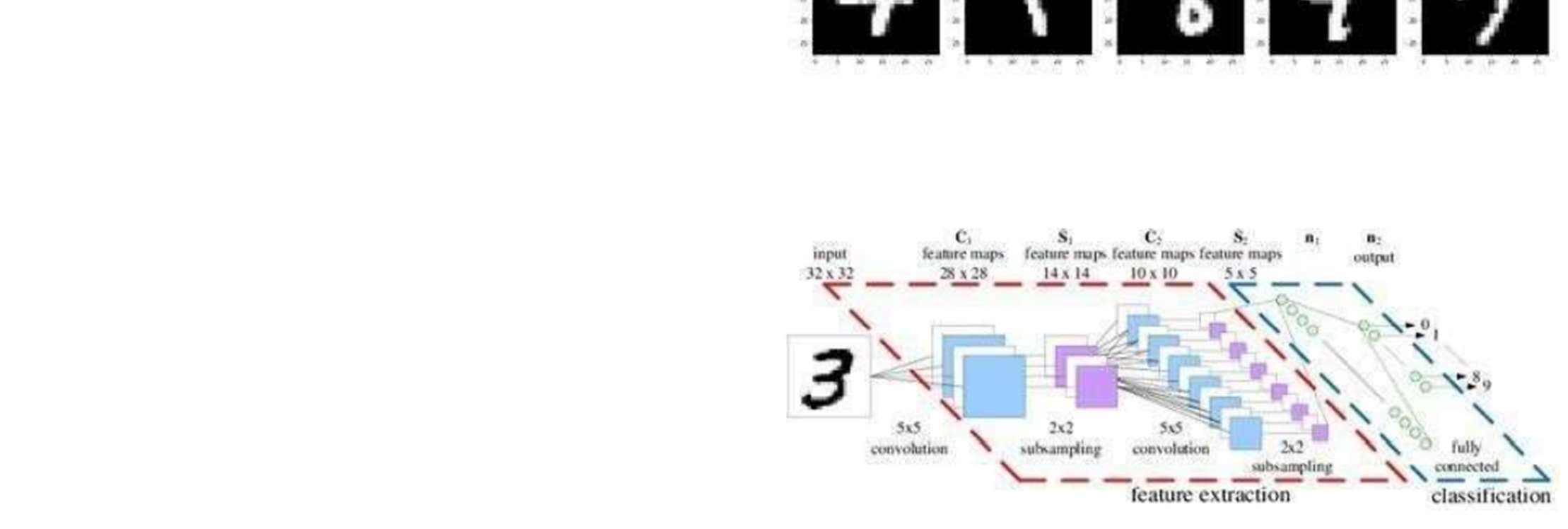
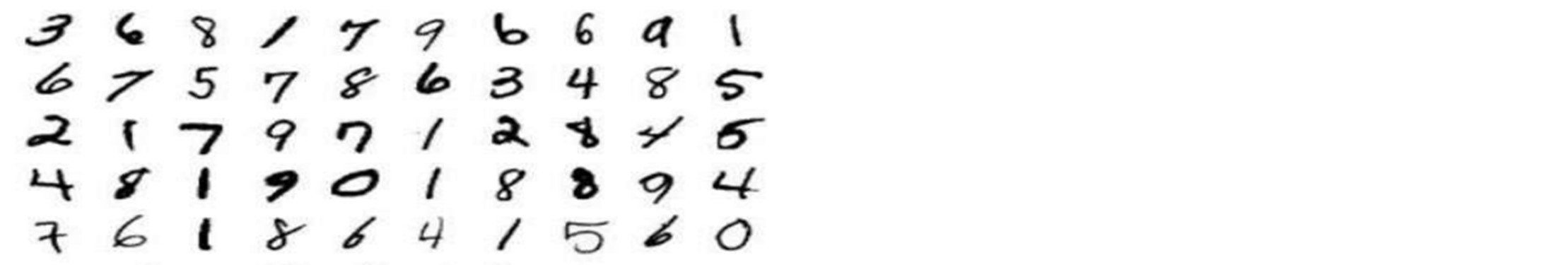
## 9. RESULTS

9.1 Performance Metrics:

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.

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.

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.



## 10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

* Image Segmentation:

Image segmentation is the process of partitioning an image into multiple segments. Image classification is the process of predicting a specific class, or label, for something that is defined by a set of data points

* Character Recognition:

The handwritten character image is converted into printed text by sending the image to the model. In this model, each character is recognized after preprocessing and they are converted into readable text.

* Less Complicated.
* Easy to process
* Accuracy is more
* User friendly
* Consistent Response and Availability

DISADVANTAGES:

* Provides inappropriate result when handwritten image is not clear
* Technology issues
* Data should be some what clear for prediction

11. CONCLUSION

Handwriting recognition is very useful software that really helps to safe and keeps data and documents well. But at time it also has its disadvantage such as that sometimes if fails to read certain people’s handwriting and due to this many people do not prefer to use the handwriting recognition software that much. Even though handwriting recognition has its disadvantages but still it is growing rapidly in the technology world. Handwriting recognition is used when there are certain people who prefer writing on the screens rather than writing it on a paper. In this project, Handwritten Digital Recognition is used In-depth learning strategies have been developed. Many widely used machine learning algorithms, RFC and CNN trained and tested on the same data into find comparisons between dividers. Use these are deeper learning methods, the higher the level of accuracy can be found. Compared to other research methods, this method focuses on which category works best for developing more than 99% separation accuracy models. Use Keras as backend and Tensorflow as software, CNN The model is able to provide about 98.72% accuracy. In the first one test, CNN provides 98.72% accuracy, while KNN provides 96.67% accuracy.

12. FUTURE SCOPE

Handwriting recognition is undoubtedly one of the most challenging areas of the pattern recognition. The goal of the project is to classify numeric samples which are mostly saved as digital images. Several pattern recognitions approaches have been applied to both online and off line handwriting recognition on the basis of unique patterns. The process of recognition consists of several steps such features extraction and recognition with voice alert. Python has a special toolbox, called neural network toolbox which makes the implementation less difficult but the knowledge of theory is needed. We can train these networks with preferred parameters. Artificial Neural Network approach for character recognition is now gaining importance because of CNN's high fault tolerance and parallel architecture. Comparing the performance of the proposed system with the eminent results of existing techniques, it is concluded that the new CNN based proposed system has achieved comparable performance to most of the existing techniques as well as it has exceeded the performance of others. Hence our results are comparable to state-of-the-art and this research will contribute positively to the research effort in the field of handwritten digit recognition. In future, other feature extraction methods and classification schemes will be considered for the digit recognition system.

13. APPENDIX

Source Code:

Importing the Required Libraries

import numpy

import matplotlib.pyplot as plt from keras.utils import np\_utils from tensorflow.keras.datasets i mport mnist from tensorflow.keras.models

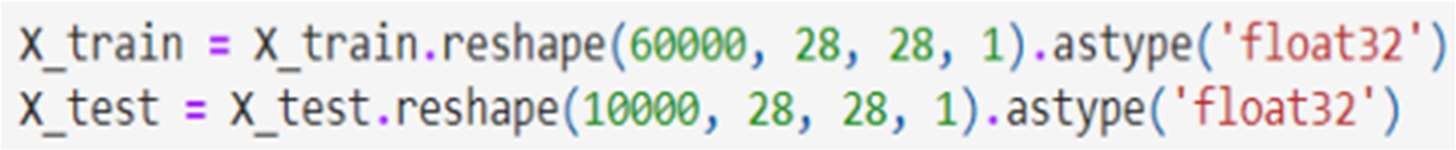
import Sequential from tensorflow.keras.layers import Conv2D, Dense, Flatten from tensorflow.keras.optimizers import Adam import numpy as np import pandas as pd from tensorflow.keras.models

import load\_model from PIL import Image, ImageOp

Loading the Data:

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

Reshaping the Data:

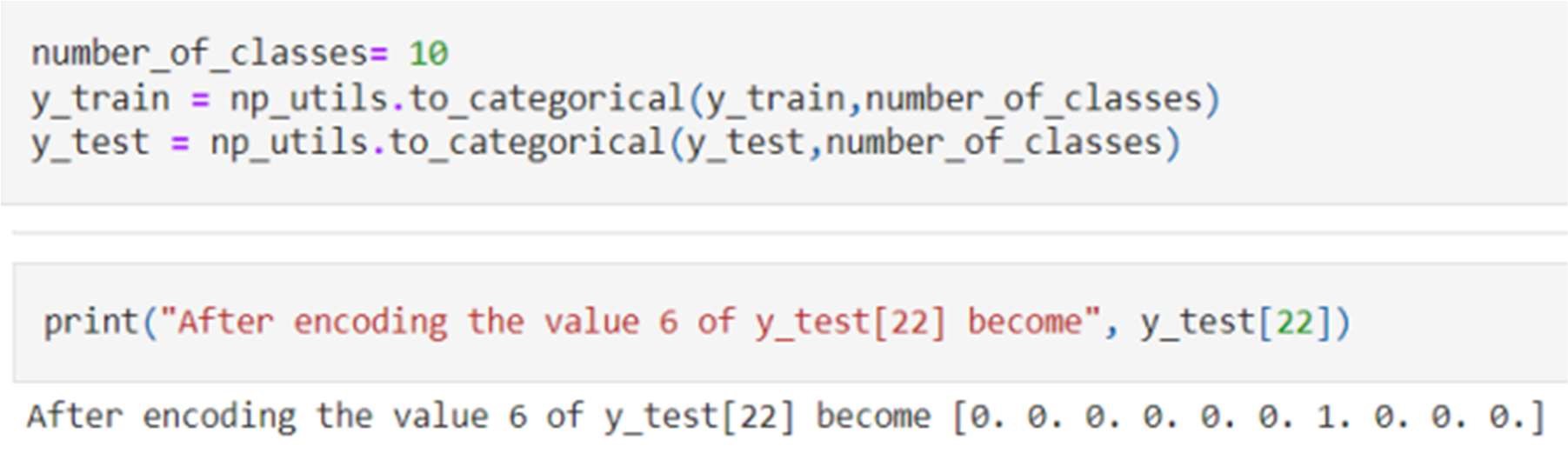


Analyzing the Data: print(X\_train.shape) print(X\_test.shape) (60000, 28, 28)

(10000, 28, 28) X\_train[0]

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]

Apply One Hot Encoding:



from flask import Flask, jsonify, render\_template, request from PIL import Image import numpy as np

# from tensorflow.keras.models import load\_model

import tensorflow as tf import numpy as np import os @app.route('/') @app.route('/index') def index():

return render\_template('home.html')

@app.route('/recognise') def recognize():

return render\_template('recongise.html')

@app.route('/predict' , methods=['GET','POST']) def predict():

data = request.files if request.method == 'POST': img = Image.open(data['file'].stream).convert("L")

img = img.resize((28,28)) im2arr = np.array(img) im2arr = im2arr.reshape(1,28,28,1) y\_pred = model.predict(im2arr) maxi = np.amax(y\_pred) classes\_x=np.argmax(y\_pred,axis=1)

# print(type(classes\_x)) return jsonify(str(classes\_x)) # for idx, x in np.ndenumerate(y\_pred):

# if x == maxi:

# print(idx)

# return jsonify(idx)

# print(list(y\_pred).index(max(y\_pred)))

return render\_template('recongise.html')

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-31584-1660203060

Demo Link:

https://drive.google.com/drive/folders/1\_6HdpvFOE8\_yvhrUvyz9ONTLAz8pzuFT?usp=share\_link