A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

Submitted by

Team Id : PNT2022TMID53487

Akshaya K B (190801004)

Baraniya S (190801008)

Mrithula Angeline M J (190801046)

Poonam B N (190801053)

TABLE OF CONTENTS:

CHAPTER1

INTRODUCTION

1.1 PROJECTOVERVIEW:

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web-based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc. Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits.

Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

1.2 PURPOSE:

The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0-9). The goal of our work is to create a model that will be able to recognize and classify the handwritten digits from images by using concepts of Convolution Neural Network. Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual’s handwriting. With high accuracy rates, the model can solve a lot of real life problems.

The main applications are vehicle license-plate recognition, postal letter-sorting services, Cheque truncation system (CTS) scanning and historical document preservation in archaeology departments, old documents automation in libraries and banks, etc. All these areas deal with large databases and hence demand high recognition accuracy, lesser computational complexity and consistent performance of the recognition system.

CHAPTER2

LITERATURE SURVEY

2.1 EXISTINGPROBLEM:

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. 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.

Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES:

Handwritten Digit Recognition using CNN(2019)

Vijayalaxmi R Rudraswamimath 1, Bhavanishankar K2

Digit Recognition is a noteworthy and important issue. As the manually written digits are not of a similar size, thickness, position and direction, in this manner, various difficulties must be considered to determine the issue of handwritten digit recognition. The uniqueness and assortment in the composition styles of various individuals additionally influence the example and presence of the digits. It is the strategy for perceiving and arranging transcribed digits. It has a wide range of applications, for example, programmed bank checks, postal locations and tax documents and so on. The aim of this project is to implement a classification algorithm to recognize the handwritten digits. The after effects of probably the most broadly utilized Machine Learning Algorithms like SVM, KNN and RFC and with Deep Learning calculation like multilayer CNN utilizing Keras with Theano and Tensorflow. Utilizing these, the accuracy of 98.70% utilizing CNN (Keras + Theano) when contrasted with 97.91% utilizing SVM, 96.67% utilizing KNN, 96.89% utilizing RFC was obtained.

Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)

Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective 5 solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fitting the dataset and provides biased predictions.

An Efficient And Improved Scheme For Handwritten Digit Recognition Based On Convolutional Neural Network (2019)

Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq and others

This study uses rectified linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classification accuracy. The system's training takes into account the time factor as well. The system is also tested by altering the number of CNN layers for additional accuracy verification. It is important to note that the CNN architecture consists of two convolutional layers, the first with 32 filters and a 5x5 window size and the second with 64 filters and a 7x7 window size. In comparison to earlier proposed systems, the experimental findings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020)

Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun

This paper's primary goal was to enhance handwritten digit recognition ability. To avoid difficult pre-processing, expensive feature extraction, and a complex ensemble (classifier combination) method of a standard recognition system, they examined different convolutional neural network variations. Their current work makes suggestions on the function of several hyper-parameters through thorough evaluation utilizing an MNIST dataset. They also confirmed that optimizing hyper-parameters is crucial for enhancing
CNN architecture performance. With the Adam optimizer for the MNIST database, they
were able to surpass many previously published results with a recognition rate of 99.89%. Through the trials, it is made 3 abundantly evident how the performance of handwritten digit recognition is affected by the number of convolutional layers in CNN architecture. According to the paper,

evolutionary algorithms can be explored for optimizing convolutional filter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.

A novel method for Handwritten Digit Recognition with Neural Networks(2020)

*Malothu Nagu*1, N Vijay Shankar#2, K.Annapurna*

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 theory, Nearest Neighbor rule, 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.

2.3 PROBLEMSTATEMENTDEFINITION

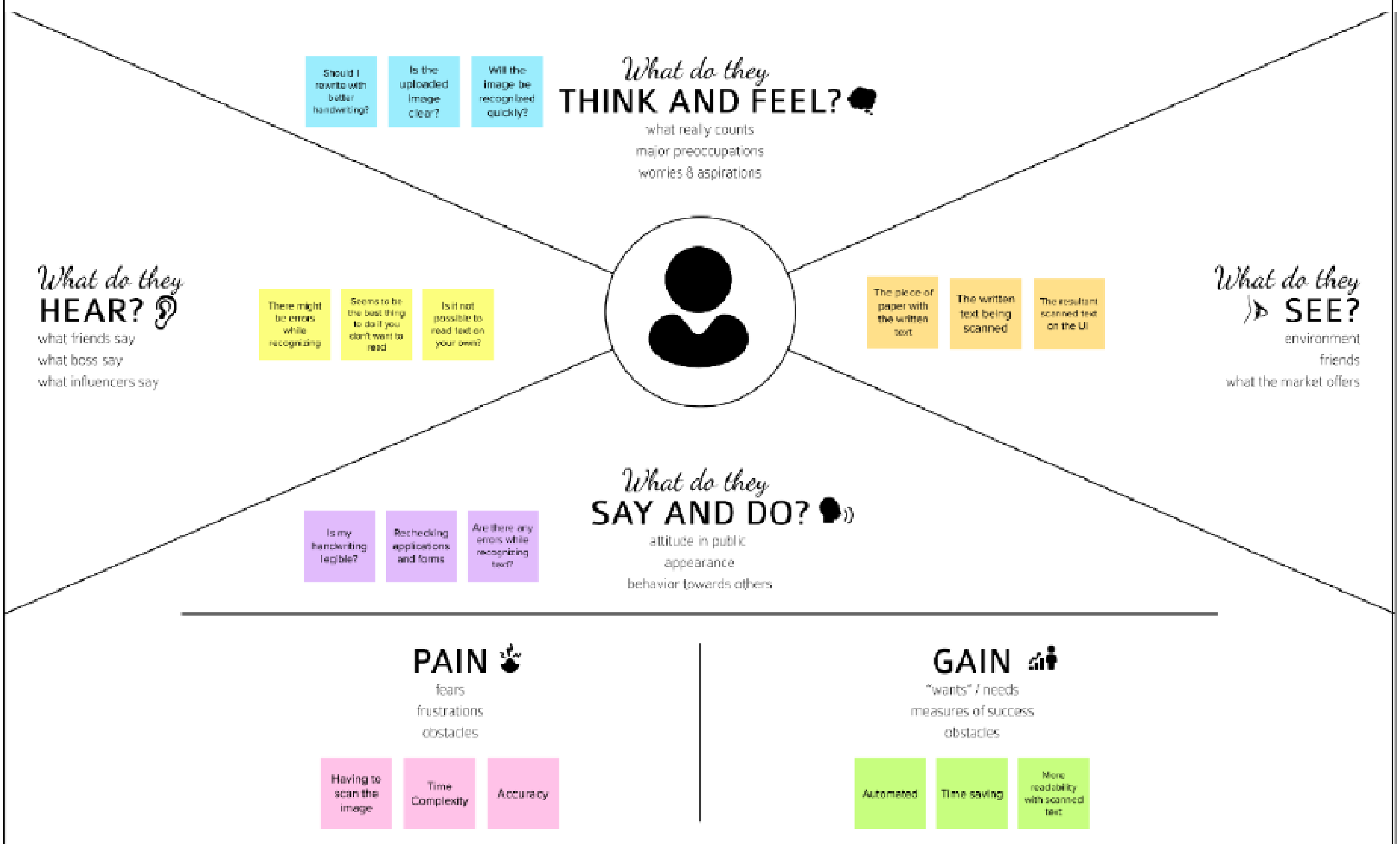
The problem statement is to classify handwritten digits. The goal is to take an image of a handwritten digit and determine what that digit and character is. It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyze images easily. Also, recognize the different elements present in the images.

The handwritten digit recognition is the capability of computer applications to recognize the 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. Convolutional Neural Network model created using Python library over the MNIST dataset to recognize handwritten digits. Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years.

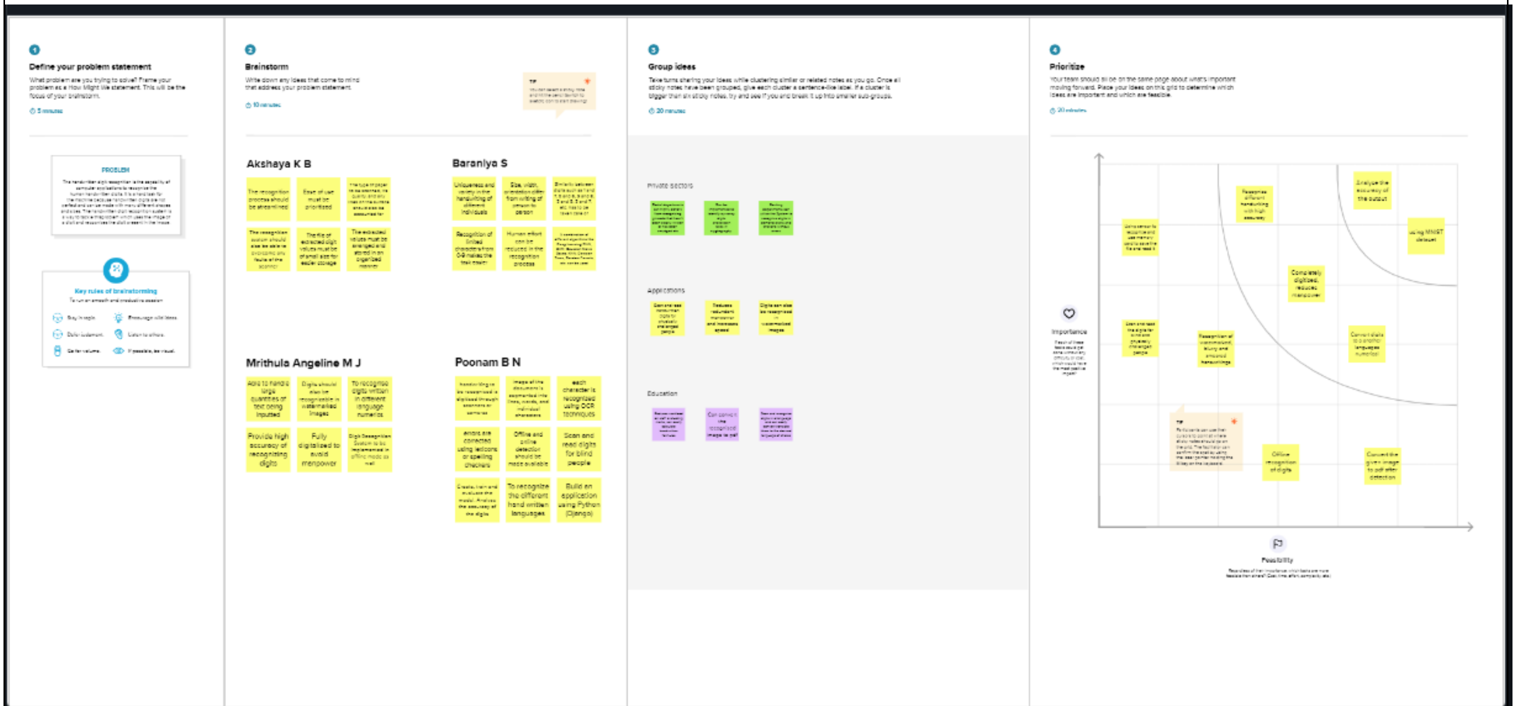
CHAPTER3

IDEATIONANDPROPOSEDSOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION&BRAINSTORMING


Define your problem statement
PROBLEM
Key rules of brainstorming
Brainstorm
Akshaya K B
Baraniya S
Mrithula Angeline M J
Poonam B N
Group ideas
Private sectors
Applications
Education
Prioritize
Importance
Feasibility

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

Who is your customer?
i.e., working parents of 0–5-year-old kids

People across the globe, especially customers who deal with handwritten digits, for example, banking sectors, schools and colleges.

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e., spending power, budget, no cash, network connection, available devices.

Errors in detection, image clarity, network connectivity issues might pose a problem

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem

or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e., pen and paper are an alternative to digital notetaking

Google drive provides the service to convert handwritten number images to digital numbers, but there are very few widely used software's to detect handwriting.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Handwritten digits can be difficult to understand and interpret at times, there must be features to help the visually challenged. It might also cause errors when it comes to bad handwritings. So, there must be improved accuracy when it comes to prediction.

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

Different people might have different styles, size, orientation of handwriting, lack of accuracy in some test cases.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?

i.e., directly related: find the right solar panel installer, calculate

Upload the image and obtain the result in a click of a button to gain efficient results.

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

3. TRIGGERS

TR

What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.

Faster, accurate and highly efficient prediction of digital numbers from handwritten images of numbers.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e., lost, insecure > confident, in control - use it in your communication strategy & design.

Before: The customers feel annoyed the handwritten digits are not legible or when the paper is wrinkled.

After: The customers feel frustrated for having to scan the images.

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior.

Our application aims to make the task of digit input easier, by reducing it to a simple procedure of scanning the paper the digits are written in, while the system handles the extraction and storage of the values. This avoids any chance of manual input errors during the process.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Online: Using software that is available on the internet, all the uploading and prediction of results will be done in online mode.

Offline: Obtaining assistance from people in order to recognize the digits.

NFR-3	**Reliability**	The Database is frequently updated with handwriting of different styles and size and will rollback when any update fails.
NFR-4	**Performance**	The proposed system is advantageous as it uses fewer features to train the neural network, which results in faster convergence.
NFR-5	**Availability**	The system functionality and services are available for use with all operations.
NFR-6	**Scalability**	The website traffic limit must be scalable enough to support 2 lakhs users at a time

CHAPTER5

PROJECT DESIGN

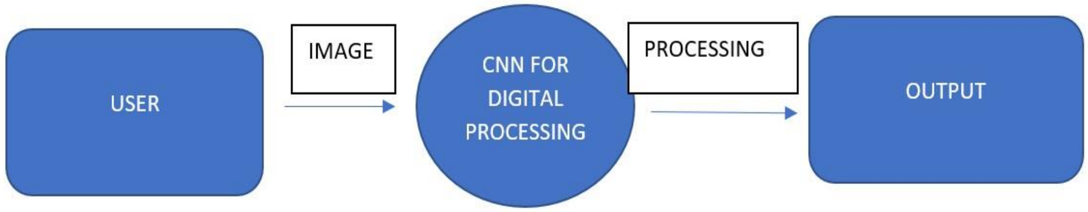
5.1 DATAFLOWDIAGRAM

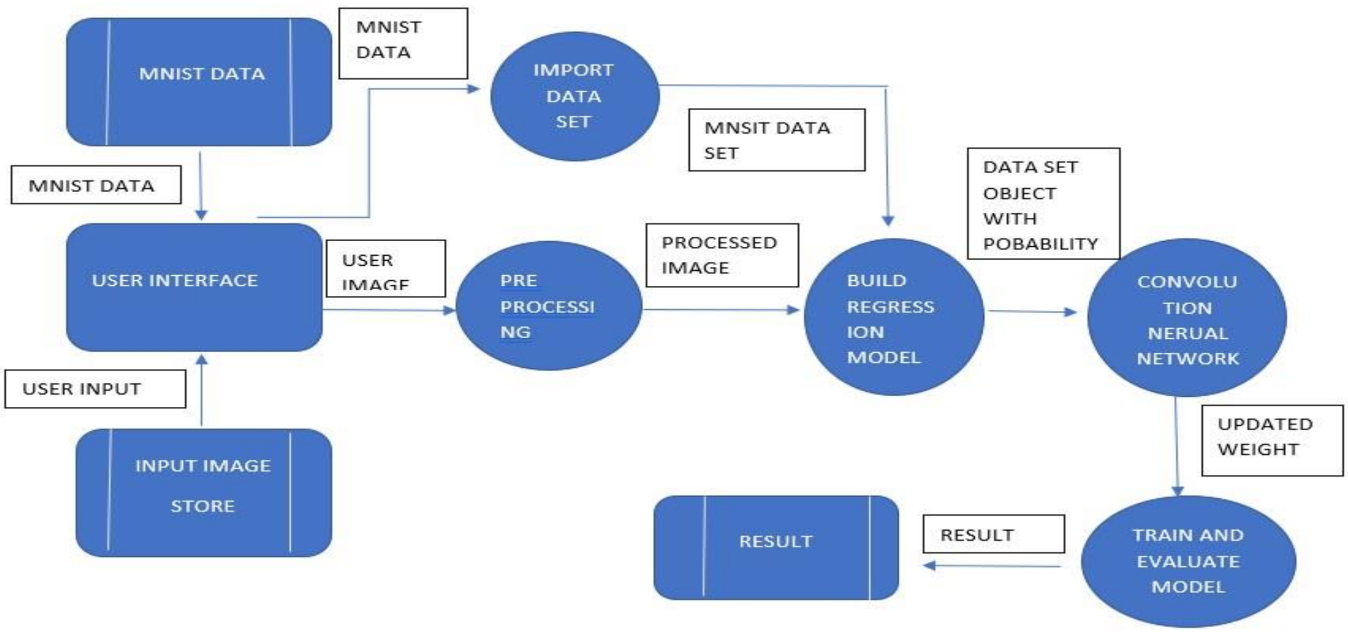
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

DFD Level-0

The DFD Level-0 consists of two external entities, the UI and the Output, along with a process, representing the CNN for Digit Recognition .Output is obtained after processing.

DFD





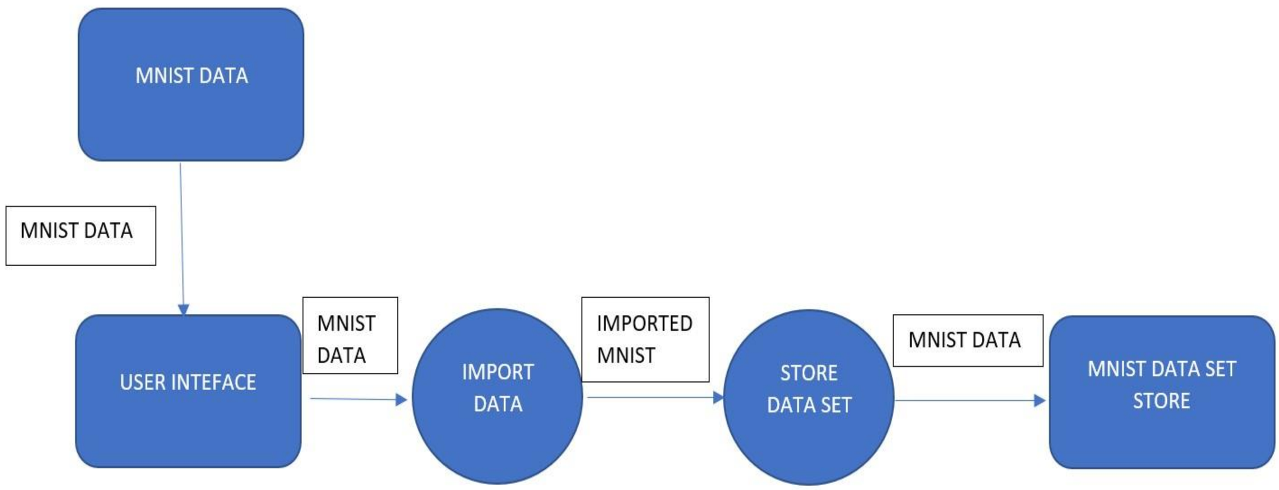
Level-1

The DFD Level-1 consists of 2 external entities, the GUI and the Output, along with five process blocks and 2 data stores MNIST data and the Input image store, representing the internal workings of the CNN for Digit Recognition System. Process block imports MNIST data from library. Process block imports the image and process it and sends it to block where regression model is built. It sends objects with probabilities to CNN where weights are updated and multiple layers are built. Block trains and evaluates the model to generate

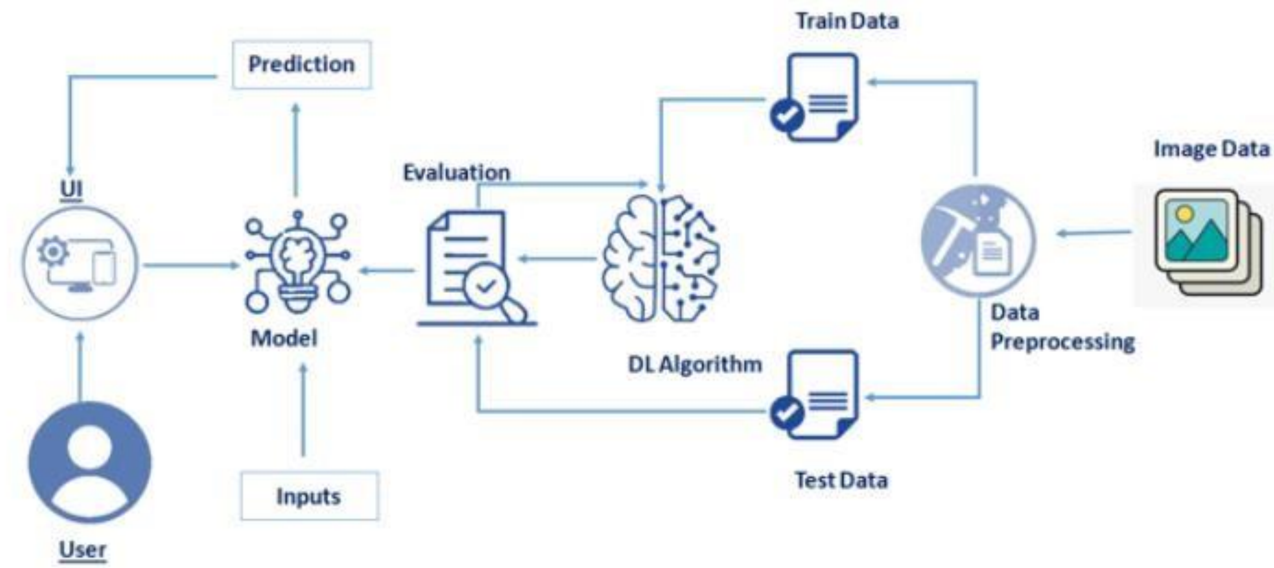
output.

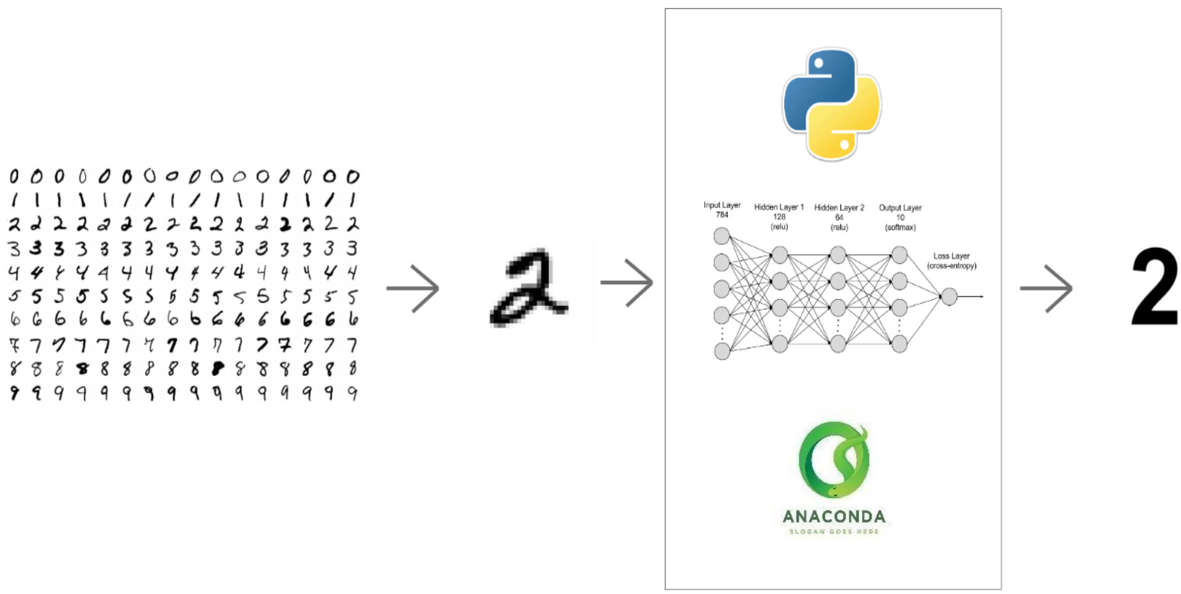
DFD Level-2

The DFD Level-2 for import data(figure 4) consists of two external data and one entity UI along with three process blocks, representing the three functionalities of the CNN for Digit Recognition System. It imports data from MNIST data store and stores on the system.



5.2 SOLUTION&TECHNICALARCHITECTURE





MNIST DATASET PROCESSING WITH PYTHON

5.4 USERSTORIES

User Type	Functional Requirement	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer	Building the application	USN-1	As a user, I should be able to access the application from any where and use on any devices	User can access the application using the browser on any device	High	Sprint-4
	Uploading Image	USN-2	As a user, I should be able to upload images to predict the digits	User can upload images	High	Sprint-2
	Viewing the Results	USN-3	As a user, I should be able to view the results	The result of the prediction is displayed	High	Sprint-3
	Viewing Other Prediction	USN-4	As a user, I should be able to see other close predictions	The accuracy of other values must be displayed	Medium	Sprint-1
	Usage Instruction	USN-5	As a user, I should have a usage instruction to knowhow to use the application	The usage instruction is displayed on the homepage	Medium	Sprint-3

CHAPTER 6 PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Mrithula Baraniya Poonam Akshaya
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Mrithula Baraniya Poonam Akshaya
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Mrithula Baraniya Poonam Akshaya
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Mrithula Baraniya Poonam Akshaya
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Mrithula Baraniya Poonam Akshaya
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Mrithula Baraniya Poonam

						Akshaya
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Mrithula Baraniya Poonam Akshaya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High	Mrithula Baraniya Poonam Akshaya
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Mrithula Baraniya Poonam Akshaya
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Mrithula Baraniya Poonam Akshaya
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High	Mrithula Baraniya Poonam Akshaya

Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Mrithula Baraniya Poonam Akshaya

6.2 SPRINTDELIVERYSCHEDULE

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	80	6 Days	24 Oct 2022	31 Oct 2022	20	08 Nov 2022
Sprint-2	80	6 Days	31 Oct 2022	5 Nov 2022	20	08 Nov 2022
Sprint-3	80	6 Days	5 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	80	6 Days	12 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORT FROM JIRA

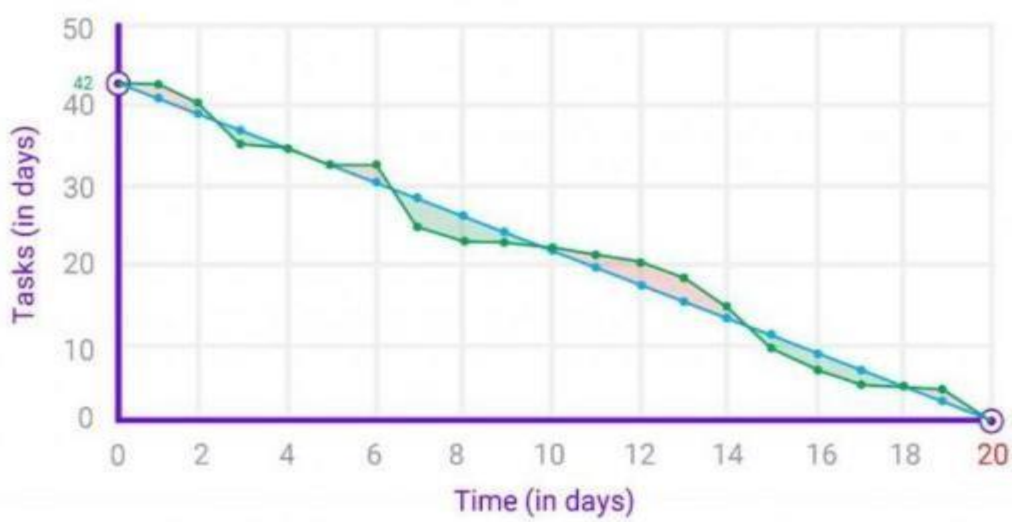
Velocity:

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 = 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 developmen t methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CHAPTER 7
CODING & SOLUTION

7.1 FEATURE 1 –FLASK FILE UPLOADING

Handling file upload in Flask is very easy. It needs an HTML form with its enctype attribute set to ‘multipart/form-data’, posting the file to a URL. The URL handler fetches file from request.files[] object and saves it to the upload folder.

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

UPLOAD_FOLDER = 'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-
recognitionsystem/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")  # convert image to monochrome
img = img.resize((28, 28))  # resizing of input image

im2arr = np.array(img)  # converting to image
im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1)  # printing our Labels
 return render_template('predict.html',
num=str(num[0])) if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

7.2 FEATURE 2 –UPLOAD IMAGE WITH PREVIEW

A preview refers to a feature that lets you glimpse or view something in part or whole without it being opened. A picture preview would show a small version of the picture and give you a good idea what each picture is without opening each picture it is a useful feature created using JavaScript.

```
<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">  Clear  </button>
</div>
</form>
</div>
</section>
```

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

7.3 FEATURE 3 – CLEAR IMAGE

This feature can be used to clear the image if we uploaded a wrong image or if we need to change the image. The clear button clears both the image value and the preview of the image in script tag.

```
<script>


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

```
</scrip>
```

CHAPTER8
TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements on the Home Page	The Home page must be displayed properly	Working as expected	PASS
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 2560x1801 And 768x630	FAIL
HP_TC_003	Functional	Home Page	Check if the 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 the user cannot upload unsupported files	The application should not allow the user to select an 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

8.2.1 DEFECT ANALYSIS

Resolution	Severity1	Severity2	Severity3	Severity4	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

8.2.2 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

CHAPTER9
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**

CHAPTER10
ADVANTAGES&DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device
- Neural Network is used to train and identify written digits for greater efficiency.
- The accuracy rate is very high.

- Speed of data entry
- It is much easier to dictate the machine than to write
- Easier data retrieval

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors
- There is a wide range of handwriting – good and bad.
- It is tricky for programmers to provide enough examples of how every character might look.
- Customers must try with clear image and neat handwriting to get accuracy in digits.
- Unclear image will not give accurate results.

CHAPTER11
CONCLUSION

Convolutional Neural Network (CNN) adds its significant improvement to the Manuscript Document Recognition System. This paper tells us the effectiveness of CNN-based classification of data and pre-processing methods. Our model clearly sees handwriting and achieves outgoing predictions of up to 82.16% and accurate predictions of up to 69.16%. However the model can be continuously developed using multiple training samples. This will help the model to learn as well as the generalize better. There are many images in the training set that are completely invisible to the human eye.

This project demonstrated a web application that uses machine learning to recognize handwritten numbers. Flask, HTML, CSS, JavaScript ,and a few other technologies were used to create this project. The model predicts the handwrittendigitusingaCNNnetwork.Duringtesting,themodelachieveda99.61%recog nition rate.Theproposedprojectisscalableandcaneasilyhandleahugenumberofusers.Sinceitis awe bapplication,itiscompatiblewithanydevicethatcanrunabrowser.Thisprojectisextremel yuse fulinreal-worldscenariossuchasrecognizingnumberplates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on.

Through extensive evaluation using a MNIST dataset, the present work suggests the role of various hyper-parameters. Fine tuning of hyper-parameters is essential in improving the performance of CNN architecture. We achieved a recognition rate of 99.89% with the Adam optimizer for the MNIST database, which is better than all previously reported results. The effect of increasing the number of convolutional layers in CNN architecture on the performance of handwritten digit recognition is clearly presented through the experiments.

CHAPTER12

FUTURESCOPE

This project can be enhanced with a great field of machine learning and artificial intelligence. The world can think of a software which can recognize the text from a picture and can show it to the others, for example a shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total English alphabet because there will be more and many more training sets and testing values that the neural network model will not be enough to detect. Think of a AI modeled car sensor going with a direction modeling in the roadside, user shall give only the destination.

All of these enhancement is an application of the texture analysis where advanced image processing, Neural network model for training and advanced AI concepts will come. These applications can be modeled further .As this project is fully done by free and available resources and packages this can be also a limitation of the project. The fund is very important because all machine learning libraries and advanced packages are not available for free. Unless of those the most of the visualizing platforms like on which developers are doing some works like Watson Studio or Aws. These all are mainly paid platforms where a lot of ML projects are going on.

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:

- Add support to detect from digits multiple images and save the results
- 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 andreducetheworkloadonmanyworkers,enhancingoverallworkefficiency.

APPENDIX

SOURCECODE

MODELCREATION:

jupyter Handwritten-Digit-Recognition Last Checkpoint: 10/20/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Importing the required libraries

In [46]:
```
import numpy
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona L Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt   #used for data visualization
```

load data

In [47]:
```
(x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the mnist data into train and test
```

In [48]:
```
print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels
print (x_test.shape)
```

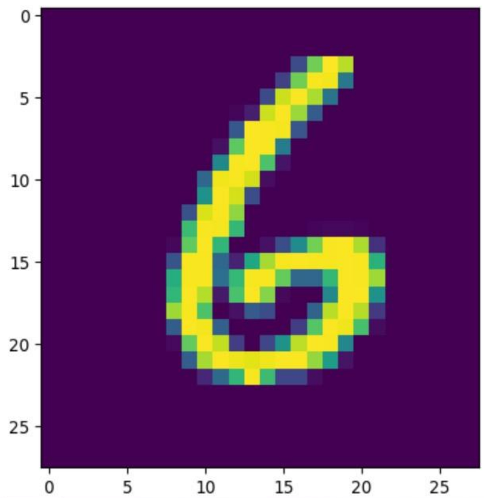
```
(60000, 28, 28)
(10000, 28, 28)
```

In [49]:
```
x_train[0]
```

Out[49]:
```
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],
```

In [50]:
```
plt.imshow(x_train[6000])      #ploting the index=image
```

Out[50]: <matplotlib.image.AxesImage at 0x222c71e0250>



Reshaping Dataset

In [52]:
```
#Reshaping to format which CNN expects (batch, height, width, channels)
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 [53]:
```
number_of_classes = 10  #storing the no of classes in a variable
```

In [54]:
```
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 [55]:
```
#create model
model=Sequential ()
```

In [56]:
```
#adding model Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
```

In [57]:
```
#flatten the dimension of the image
model.add(Flatten())
```

In [58]:
```
#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling the model

In [59]:
```
#Compile model
model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

In [62]:
```
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

Train the model

In [63]:
```
#fit the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 88s 47ms/step - loss: 0.2299 - accuracy: 0.9506 - val_loss: 0.0899 - val_accuracy:
0.9730
Epoch 2/5
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0682 - accuracy: 0.9796 - val_loss: 0.0765 - val_accuracy:
0.9780
Epoch 3/5
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0493 - accuracy: 0.9838 - val_loss: 0.0888 - val_accuracy:
0.9755
Epoch 4/5
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0357 - accuracy: 0.9887 - val_loss: 0.0835 - val_accuracy:
0.9791
Epoch 5/5
1875/1875 [==============================] - 89s 48ms/step - loss: 0.0288 - accuracy: 0.9907 - val_loss: 0.1145 - val_accuracy:
0.9708
```

Out[63]:
```
<keras.callbacks.History at 0x222d90d9db0>
```

Observing the metrics

In [64]:
```
# Final evaluation of the model
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.1144733875989914, 0.97079998254776]
```

Test The Model

In [65]:
```
prediction=model.predict(x_test[6000:6001])
print(prediction)
```

```
1/1 [==============================] - 0s 84ms/step
[[1.4689527e-14 1.8748821e-17 2.3109615e-13 9.5624716e-07 3.1683821e-04
  2.4785629e-07 6.1842380e-18 6.2185841e-06 8.8746788e-07 9.9967492e-01]]
```

Save The model

In [70]:
```
# Save the model
model.save('models/mnistCNN.h5')
```

CNNPREDICTION:

```
In [2]: from tensorflow.keras.models import load_model
        from keras.preprocessing import image
        from PIL import Image
        import numpy as np

In [3]: model = load_model("mnistCNN.h5")

In [4]: img = Image.open("C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-system/data/1.png").convert("L") # convert i
        img = img.resize( (28,28) ) # resizing of input image

In [5]: img
```

Out[5]:

```
In [6]: im2arr = np.array(img) #converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement

In [7]: pred = model.predict(im2arr)
        print(pred)

        1/1 [==============================] - 0s 182ms/step
        [[2.5381066e-09 4.9758598e-01 6.5878254e-07 3.7901787e-06 5.3061078e-05
          2.5423644e-06 2.0804979e-10 9.8954014e-02 1.2672696e-03 4.0213263e-01]]
```

TRAIN THE MODEL ON IBM:

```
In [1]: from ibm_watson_machine_learning import APIClient
        credentials ={
            "url":"https://jp-tok.ml.cloud.ibm.com",
            "apikey":"BHyalu2c7JN6n9cnvAVULvSKRYFVLMQ_m51toZ9Yk0nS"
        }
        client = APIClient(credentials)

In [2]: def guid_from_space_name(client,deploy):
          space = client.spaces.get_details()
          return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])

In [3]: space_uid = guid_from_space_name(client,'digitrecognition')
        print("Space UID = " + space_uid)

        Space UID = aa24227a-9f01-493f-90e6-1b6132057fc6

In [4]: client.set_default_space(space_uid)

Out[4]: 'SUCCESS'

In [6]: client.repository.download("97d463b1-45ee-47f7-b8af-aed338794ce1", 'DigitRecog_IBM_model.tar.gz')

        Successfully saved model content to file: 'DigitRecog_IBM_model.tar.gz'

Out[6]: 'C:\\Users\\Dell\\PycharmProjects\\A-novel-method-for-digit-recognition-system/DigitRecog_IBM_model.tar.gz'
```

HOME PAGE(HTML) – index.html

```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&displa
y=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="sha384q8i/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: PNT2022TMID27424</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.
</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 analysed 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">  Clear  </button>
</div>
</form>
</div>
</section>


</body>


</html>
```

HOME PAGE(CSS) – 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: 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%;
  }
}
```

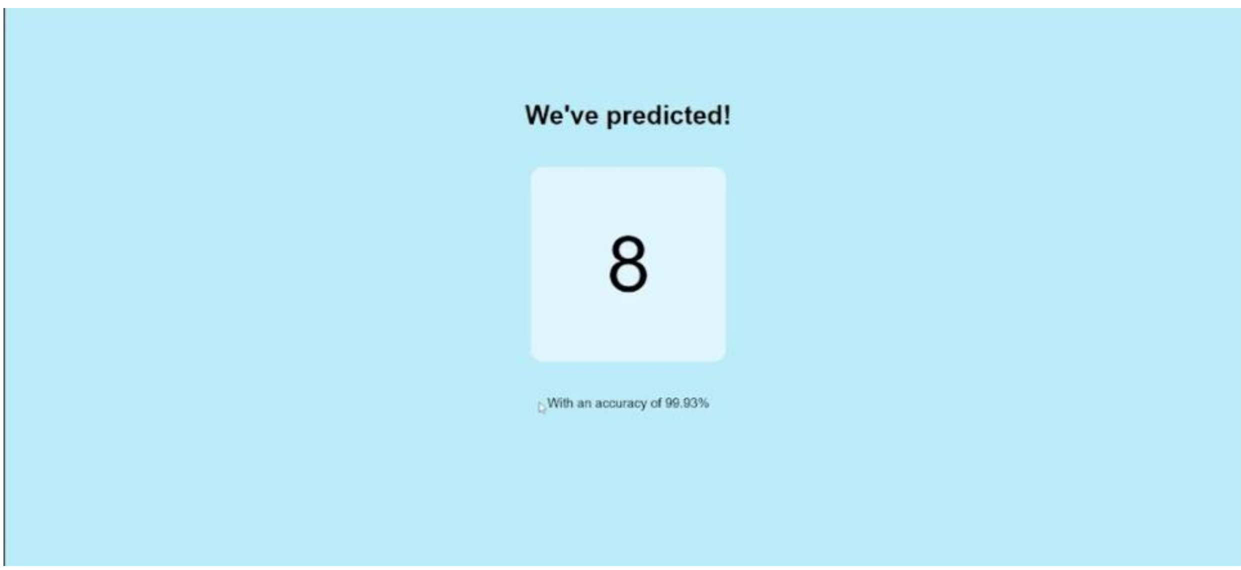
PREDICTPAGE(HTML) – predict.html

```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>
```

FLASK APP - app.py

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

UPLOAD_FOLDER = 'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-
recognitionsystem/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")  # convert image to monochrome
img = img.resize((28, 28))  # resizing of input image

im2arr = np.array(img)  # converting to image
im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

pred = model.predict(im2arr)


        num = np.argmax(pred, axis=1)  # printing our Labels

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

SCREENSHOTS



GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-25783-1659973203

Team Id : PNT2022TMID53487

Project Name : A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

Team Members :

Akshaya K B (190801004)

Baraniya S (190801008)

Mrithula Angeline M J (190801046)

Poonam B N (190801053)