A Novel Method for Handwritten Digit Recognition System

TEAM MEMBERS

Chalini P - SSNCE 195001026
Hema M - SSNCE 195001037
Dhivya Lakshmi. RM - SSNCE 195001028
Anza Prem - SSNCE 195001020

TEAM ID - PNT2022TMID53039

Git Repo Link - https://github.com/IBM-EPBL/IBM-Project-9293-1658991853

PROJECT REPORT

1. **INTRODUCTION:**

1.1 Project Overview

The Handwritten digit Recognition system is a technology that is of great use to all of us. The novel method we have proposed will detect handwritten digits ranging between 0-9 from the MNIST dataset. By using convolutional neural networks and integrating that with the web application using flask, we are able to recognise handwritten digits with high accuracy.
This efficient method can be used for a variety of purposes that can be tedious when done by us. Some of the applications include recognising digits from bank cheques, old documents, images etc. This system can scan the numbers and document them for easy access. This can be used in banks or offices that require physical data to be stored digitally.

1.2 Purpose

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. Our purpose is to automate the process of scanning the information on old manuscripts, bank cheques, letters, images, etc and storing them digitally for future reference.

2. LITERATURE SURVEY:

2.1 Existing Problem

- The postman needs a way to interpret handwritten postal addresses so that he delivers letters to the correct addresses.
- The banker needs a way to understand handwritten bank cheques so that the right amount can be transferred to the right party.
- Authorities need a way to verify signatures to validate certain documents.
- Historians and writers need a way to interpret old handwritten papers and letters so that they can learn more about what happened in the past.

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 digitized to reduce human effort. It's better to have different approaches to the same problem because the use cases will differ in today 's examples.

2.2 Reference

Paper 1

Hybrid CNN-SVM Classifier for Handwritten Digit Recognition —2020

Savita Ahlawata, Amit Choudharyb

https://reader.elsevier.com/reader/sd/pii/S1877050920307754?token=255634CBFA4B29759D50EF7B36744C6A9188E2C8753392E7248C6B789632042A5517FDD577C664653923CB4293189277&originRegion=eu-west-1&originCreation=20220919085416

The aim of this paper is to develop a hybrid model of a powerful Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) for recognition of handwritten digit from MNIST dataset. The proposed hybrid model combines the key properties of both the classifiers. In the proposed hybrid model, CNN works as an automatic feature extractor and SVM works as a binary classifier.

The MNIST dataset of handwritten digits is used for training and testing the algorithm adopted in the proposed model. The MNIST dataset consists of handwritten digits images which are diverse and highly distorted. The receptive field of CNN helps in automatically extracting the most distinguishable features from these handwritten digits. The experimental results demonstrate the effectiveness of the proposed framework by achieving a recognition accuracy of 99.28% over MNIST handwritten digits dataset.

Paper 2

Deep capsule network for recognition and separation of fully overlapping handwritten digits –2021

HonggeYaoa | YuxingTan | Chunqiu Xu | Jun Yu | Xiaojun Baia

https://www.sciencedirect.com/science/article/pii/S0045790621000513

The recognition and separation of fully overlapping handwritten digits is an effective means to test the recognition ability of the network. It is also the basis of separating overlapping complex handwritten characters. This paper constructs a deep capsule network, FOD_DCNet, for the recognition and separation of fully overlapping handwritten digits. Firstly, we used small sized convolution kernels to extract features, which is conducive to extracting fine-grained features while reducing training parameter; secondly, we expanded the capsules dimension to express the extracted features to avoid the loss and omission of features; thirdly, we proposed "series dual dynamic routing collocation" to optimize the routing classification function. Compared with CapsNet, our FOD_DCNet reduces the number of iterations of each route, and increases the classification efficiency. The classification accuracy of FOD_DCNet can reach 93.53%, which is 5.43% higher than CapsNet and its parameter amount is only 55.61% of CapsNet.

Paper 3

Hierarchical Convolutional Neural Network for Handwritten Digits Recognition–2019

Zufar Kayumov, Dmitrii Tumakov, Sergey Mosin

https://www.sciencedirect.com/science/article/pii/S1877050920311881

The application of a combination of convolutional neural networks for the recognition of handwritten digits is considered. Recognition is carried out by two sets of the networks following each other. The first neural network selects two digits with maximum activation functions. Depending on the winners, the next network is activated, which selects one digit from two. The proposed algorithm is tested on the data from MNIST. The minimal handwriting recognition error was estimated with this approach.

Paper 4

A Comparative Study on Handwriting Digit Recognition Using Neural Networks-2017

Mahmoud Abu Ghosh, Ashraf Yunis Maghari

https://www.researchgate.net/publication/321121793_A_Comparative_Study_on_Handwriting_Digit_Recognition_Using_Neural_Networks

The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). In this paper, the three NN approaches are compared and evaluated in terms of many factors such as accuracy and performance. Recognition accuracy rate and performance, however, is not the only criterion in the evaluation process, but there are interesting criteria such as execution time. Random and standard dataset of handwritten digit have been used for conducting the experiments. The results show that among the three NN approaches, DNN is the most accurate algorithm; it has 98.08% accuracy rate. However, the execution time of DNN is comparable with the other two algorithms. On the other hand, each algorithm has an error rate of 1-2% because of the similarity in digit shapes, specially, with the digits (1,7), (3,5), (3,8), (8,5) and (6,9).

2.3 Problem Statement Definition

<u>A Novel Method For Handwritten Digit Recognition System</u>

A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach in a website where the user can upload an image and the results will be displayed immediately.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation and Brainstorming

Brainstorming

Hema

- An app to upload images
- Use AI
- App that can be used for signature verification
- Choose appropriate ML Model
- Scan using smart phone to recognize
- App that can recognize postal code, zip code

Chalini

- An app that can click pics of handwritten text
- An app that can scan handwritten letters, papers
- Use deep learning models
- Use ANN
- Use Flask for building app
- An app that processes bank cheques

Dhivya

- An app that recognizes handwritten digits
- Use Django, HTML, CSS for web app
- Integrate web app with ML model
- Use CNN as the DL Model
- An app that enhances images before recognition
- Use large dataset for training and testing

Anza

- An app that can recognize any handwriting
- Use pattern recognition
- Mobile app that can be used anywhere, anytime
- An app that can be used for postal address verification
- Use RNN
- Use Bayesian Decision theory, Nearest Neighbor rule

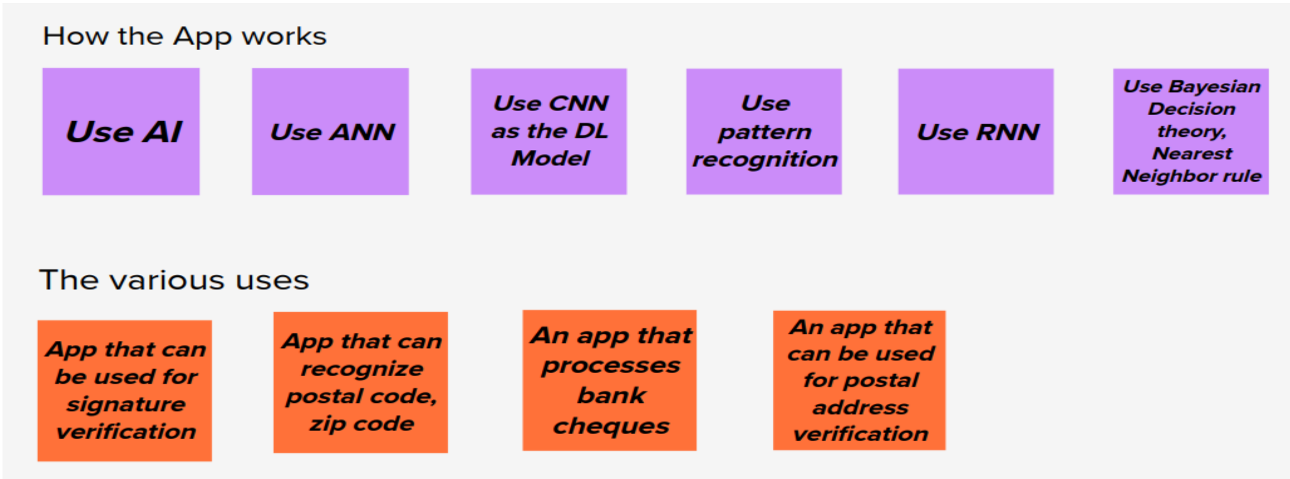
Grouping Ideas

What the App does

- An app to upload images
- An app that can click pics of handwritten text
- An app that can scan handwritten letters, papers
- An app that recognizes handwritten digits

How the App is built

- Use Flask for building app
- Use Django, HTML, CSS for web app
- Integrate web app with ML model


How the App works
Use AI
Use ANN
Use CNN as the DL Model
Use pattern recognition
Use RNN
Use Bayesian Decision theory, Nearest Neighbor rule
The various uses
App that can be used for signature verification
App that can recognize postal code, zip code
An app that processes bank cheques
An app that can be used for postal address verification

Prioritize


An app to upload images
Use CNN as the DL Model
An app that recognizes handwritten digits
An app that can scan handwritten letters, papers
Use Flask for building app
Integrate web app with ML model
An app that can click pics of handwritten text
Use Bayesian Decision theory, Nearest Neighbor rule
Use pattern recognition
Importance
Feasibility

3.3 Proposed Solution

- **Problem Statement:**
A Novel Method for Handwritten Digit Recognition System.

- **Idea/Solution description:**
Using Deep learning techniques such as Convolutional Neural Network to recognize handwritten digits. A web application that takes images of handwritten digits as input and recognizes them. The application is built using Flask Framework.

- **Novelty/Uniqueness:**
Usage of upcoming technologies like Deep learning and Artificial Neural Network. The current system is more efficient and reliable than the ones that were traditionally used.

- **Social Impact/Customer Satisfaction:**
People can now easily interpret postal code, zip code, bank cheques, handwritten letters, papers, etc. People are now more confident while dealing with handwritten texts/digits. This application can be applied on a lot more Use-cases.

- **Business Model/Financial Benefit:**
The application can be built free of cost with currently available technologies and dataset.

- **Scalability of solution:**
The application can be further enhanced to recognize handwritten text as well.
This can be applied in a lot of instances like signature verification, etc.
The application can be developed in such a way that it can scan a handwritten digit/text and immediately display what has been recognized.
Can make the whole process quicker and easier.

3.4 Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Postmen, people delivering couriers.
People working in banks.
Historians, writers, etc trying to interpret old handwritten letters, papers.

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.

The available solution is not widely known and used.
The solution has to be made easily available for everyone to use.

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 is an alternative to digital notetaking

Character extraction->character recognition
Feature extraction

Modern methods are more efficient and reliable than the above mentioned traditional methods.

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.

A method to recognize handwritten digits.

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?
i.e. customers have to do it because of the change in regulations.

The need to recognize handwritten digits and text.
The need to interpret postal address, bank Cheques, old handwritten letters, papers and for signature verification.

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 usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Find if there are already available methods to recognize handwritten digits and texts, and if yes try to make use of them.

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

Identify strong TR & EM

3. TRIGGERS TR

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

Upcoming technologies like Deep learning and ANN.

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: confused, helpless
After: confident, hopeful

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 behaviour.

A web application that takes images of handwritten digits and recognizes them.
The underlying technology is CNN and the app is built using Flask Framework.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

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

Automatic conversion of text as it is written on a specialized digitizer or PDA.

8.2 OFFLINE

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

Scan the handwritten digit or text and then use that for recognition.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	Handwritten digit recognition - identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorize them into ten established classifications (0-9).
FR-2	Website - It should be easily accessible through a web application and should be convenient to use. Web hosting makes the code, graphics, and other items that make up a website accessible online.
FR-3	To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. Get the training and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. It is a virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Dataset: MNIST dataset is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 Non-Functional Requirements

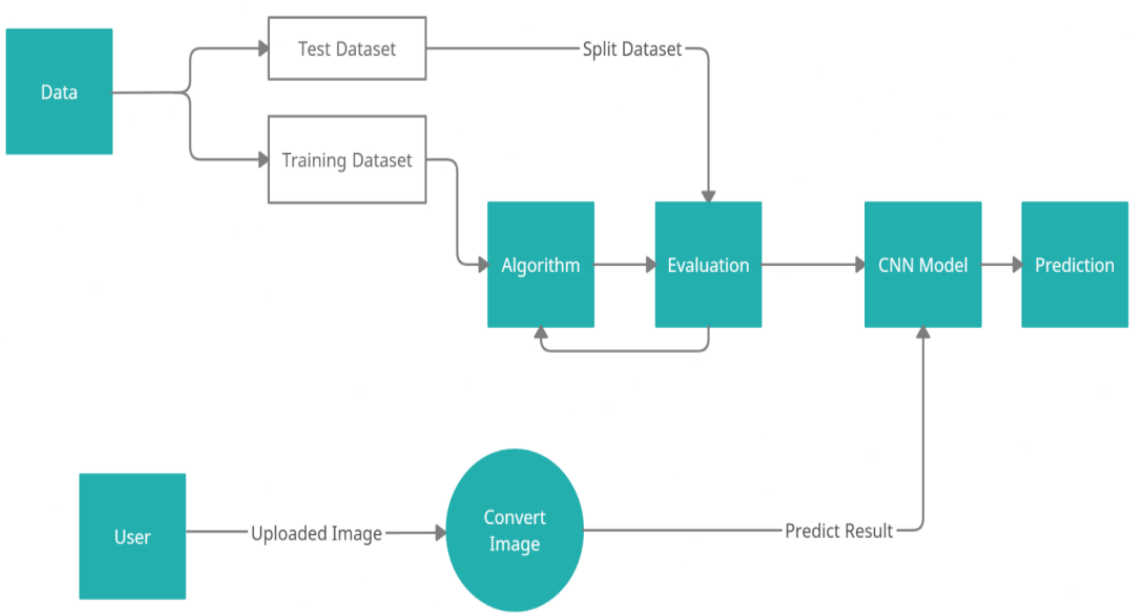
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Applications for digit recognition include filling out forms, processing bank checks, and sorting mail. Also can be used in the documentation of large numbers of old bank documents etc.
NFR-2	**Security**	The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit.

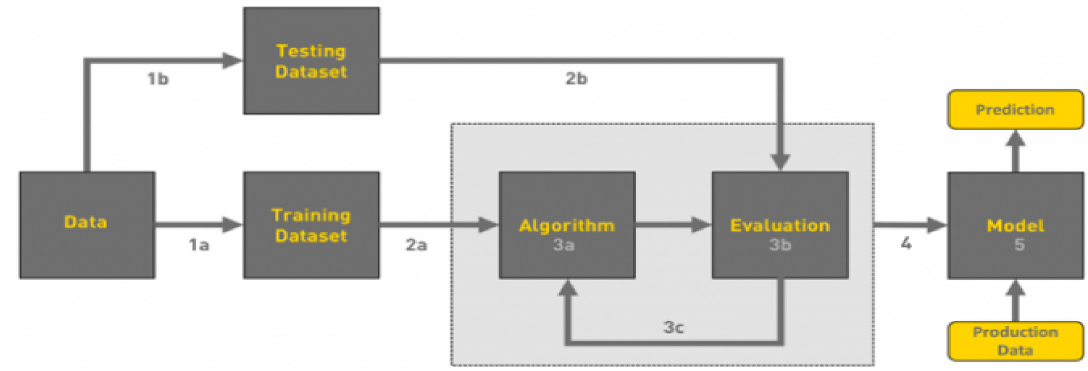
NFR-3	**Reliability**	Numerous techniques and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, etc., can be used to recognise handwritten numbers. The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances.
NFR-4	**Accuracy**	With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.
NFR-5	**Availability**	The website should be available 24/7 in android or iOS or windows. It should be easily accessible by the user at all times.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution and Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Image upload	USN-1	User can upload image using the input component	user can upload only images	High	Sprint-1
Customer (Web user)	image upload	USN-2	adding upload button to send image for prediction	button is enabled only when image is uploaded	High	Sprint-1
Customer (Web user)	results	USN-3	graph to show the prediction		High	Sprint-4
Customer (Web user)	training	USN-4	importing keras datasets and integrating it		High	Sprint-2
Customer (Web user)	training	USN-5	model training	image dataset only	High	Sprint-2
Customer (Web user)	training	USN-6	model testing	image dataset only	High	Sprint-3
Customer (Web user)	results	USN-7	result formatting		High	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation

MILESTONE AND ACTIVITY LIST

SPRINT	TASK	STORY POINTS	PRIORITY	MEMBERS
SPRINT 1	GET DATASET	2	High	Anza
	DATA PREPROCESSING	3	High	Chalini
	FEATURE EXTRACTION	3	Medium	Hema
	PREPARE TRAINING AND TESTING DATASET	2	High	Dhivya
SPRINT 2	EXPERIMENT ON DIFFERENT MODELS	1	medium	Dhivya
	CREATE MODEL	3	High	Chalini and Dhivya
	TRAIN THE MODEL	2	HIgh	Anza
	TEST THE MODEL	2	High	Hema
SPRINT 3	SETUP CLOUD SERVICE AND DATABASE	3	High	Anza
	TRAIN MODEL ON IBM	2	High	Hema and Chalini
	BEGIN DEVELOPMENT OF WEBSITE	2	High	Dhivya and Hema
	PYTHON CODE FOR APPLICATION	3	High	Chalini and Dhivya
SPRINT 4	FINISH THE WEBSITE	2	High	Dhivya
	INTEGRATE MODEL WITH APPLICATION	3	High	Chalini and Hema
	TEST THE APPLICATION	2	High	Anza

6.2 Sprint Delivery Schedule

SPRINT DELIVERY PLAN

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT RELEASE DATE
SPRINT 1	10 POINTS	2 DAYS	NOV 7
SPRINT 2	8 POINTS	3 DAYS	NOV 10
SPRINT 3	10 POINTS	6 DAYS	NOV 16
SPRINT 4	7 POINTS	3 DAYS	NOV 19

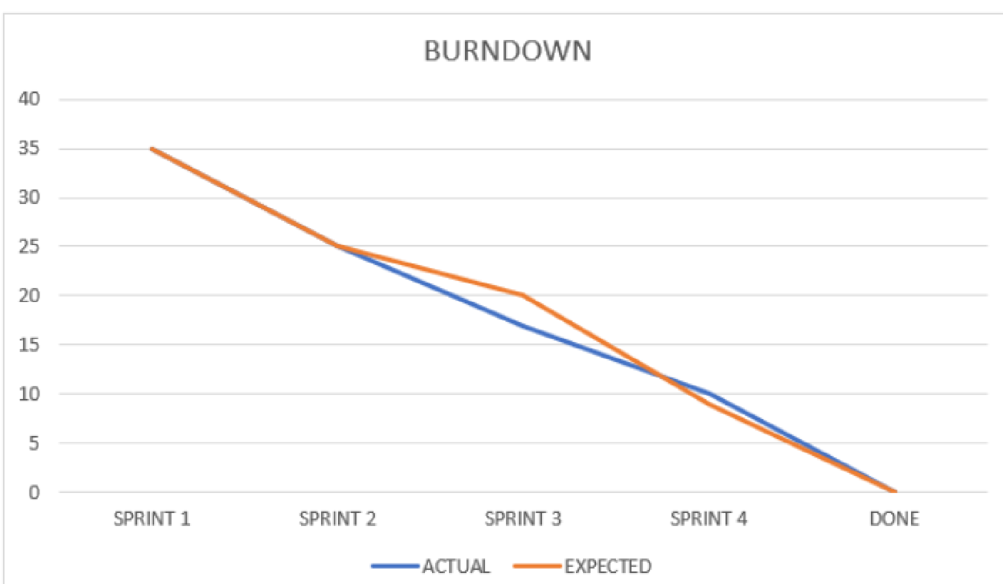
VELOCITY

Velocity = $\frac{\sum sprint\ points}{Total\ sprints}$

= (10+8+10+7)/4 = 35/4
= 8.75

Av velocity = velocity /avg duration
= 8.75 / 3.5
= 2.5

BURNDOWN GRAPH



7. CODING AND SOLUTIONING

7.1 Feature 1

Data Acquiring and Model Building

- Import the MNIST dataset using keras.
- Analyze and reshape the data and apply one-hot encoding.
- Build the model by adding CNN layers.
- Train and test the model and observe the metrics

Code:

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

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

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

```
model=Sequential()

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


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

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

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

```
#fitting the model

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

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

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

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

7.2 Feature 2

Application Building

- Create html pages for the web application. Style the pages using css.
- Home page and a page to upload an image of a handwritten digit.
- Integrate the frontend with the model using flask as the backend.
- Run the application and check the output.

CODE:

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hand Written Digit Recognizer</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <nav class="nav" style="font-size:larger;font-weight:bolder;text-decoration: none;">
    <span>
        <a href="/index">Home</a>
    </span>
    <span>
        <a href="/web">Recognizer</a>
    </span></nav>
    <p style="margin:50px 200px;font-size: medium;font-weight: 300;font-family: Verdana, Geneva, Tahoma, sans-serif;">
    The Handwritten digit Recognition system is a technology that is of great use to all of us.
    The novel method we have proposed will detect handwritten digits ranging between 0-9 from the MNIST dataset.
    By using convolutional neural networks which  is integrated with the web application using flask,
    we are able to recognise handwritten digits with high accuracy.
    This efficient method can be used for a variety of purposes that can be tedious when done manually by humans.
    Some of the applications include recognising digits from bank cheques, old documents, images etc.
    This system can scan the numbers and document them for easy access.
    This can be used in banks or offices that require physical data to be stored digitally.
    </p>
</body>
</html>
```

web.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
```

```html
<body>
    <h2>Handwritten Digit Recognition</h2>
    <div class="separator">
        <div class="col left">
            <!--<h3>Input</h3>-->
            <form method="POST" action="/predict" enctype="multipart/form-data">
                <br>
                <br>
                <br>
                <br>
                Upload an image:
                <br>
                <br>

                <input type="file" name="file">

                <button type="submit">Submit</button>

            </form>
        </div>
        <div class="col right">
            <h3>Results</h3>
            {% if predicted %}

            <p>{{ predicted }}</p>


            {% endif %}
        </div>
    </div>
</body>
</html>
```

Flask: app.py

```python
#Flask-framework to run/serve application
from PIL import Image
import numpy as np
from tensorflow.keras.models import load_model
import tensorflow as tf
from flask_cors import CORS
import flask
import pandas as pd
from flask import request, render_template

app = flask.Flask(__name__, static_url_path='')

@app.route('/', methods = ['GET'])
def home():
    return render_template('index.html')

@app.route('/web', methods=['GET'])
def recognize():
    return render_template('web.html')

@app.route('/index', methods=['GET'])
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
```

```python
def predict():
    if request.method == 'POST':
        img = Image.open(request.files['file'].stream).convert("L")
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)
        model = load_model('models/mnistCNN.h5')
        predicted = model.predict(im2arr)
        predicted = np.argmax(predicted, axis=1)
    return render_template('web.html', predicted=str(predicted[0]))

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8000, debug=True)
```

8. TESTING

8.1 Test Cases

TEST ID	TEST CASE	STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
T01	Load website	1 .click home	Click home	home	home	Pass
T02	recognizer	1.click recognizer	Click recognizer	Recognizer page	Recognizer page	Pass
T03	Upload image	1.get image from file 2. Upload file from device	file	File accepted	File accepted	Pass
T04	Result	After uploading result to be displayed	Click submit	Result displayed	Result displayed	Pass

8.2 User Acceptance Testing

TEST ID	TEST CASE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
T01	Upload image	User Handwritten digit image file	5	5	Pass
T02	Recognising Input	Click submit	7	7	Pass
T03	Upload image	User Handwritten digit image file	9	9	Pass
T04	Recognising Input	Click submit	6	6	Pass

9. RESULTS

9.1 Performance Metrics

TEST LOSS	TEST ACCURACY
0.139	96.95%

```
#final evaluation of the model

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

```
Metrics(Tesr loss and Test Accuracy):
[0.13925771415233612, 0.9695000052452087]
```

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

```
1/1 [==============================] - 0s 267ms/step
[[2.00554035e-11 3.09320120e-17 3.29157037e-14 4.59538130e-10
  2.57778689e-20 4.48461340e-15 3.88025579e-20 1.00000000e+00
  7.38416446e-12 3.22724292e-09]
 [2.27445784e-10 3.75968310e-12 1.00000000e+00 6.89282819e-14
  9.86108346e-17 3.51513529e-19 1.02042966e-11 2.09608569e-20
  4.32586231e-14 6.42716731e-24]
 [5.93559219e-13 1.00000000e+00 9.82012693e-10 2.65289066e-13
  6.25340002e-09 2.25731521e-12 2.09223375e-10 3.44775763e-08
  2.75011992e-11 8.85743128e-12]
 [1.00000000e+00 5.15908509e-21 5.68719334e-14 1.36535135e-17
  3.80047967e-18 5.10071044e-14 1.19628552e-09 2.12989452e-16
  3.06164057e-13 7.82308419e-12]]
```

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

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

10. ADVANTAGES AND DISADVANTAGES

Advantages:

- ➔ Automating the process of digitizing old handwritten documents for future reference.
- ➔ Allows people to write on the cell phones using stylus and then the phone software translates the written words to the phone in text.
- ➔ Damaged or corrupted historical papers (Eg:genealogical information, written manuscripts, etc.) can be transformed to a text document format which can also be said as readable electronic format. By this way, historical facts can be stored, reviewed and shared easily too many people.

Disadvantages:

- ➔ Due to unnatural letters strokes, sometimes the software predict the letters wrongly.
- ➔ Since not everyone's handwriting is the same, it fails to read certain people's handwriting.

11. FUTURE SCOPE

1. Develop the system further to accommodate more that single digit s that it can be used for digitalising old account records and other outdated documents.
2. Improve the accuracy so that it can be included in more phones to recognise text written using stylus.
3. Further, the model can be modified to recognise text data too.

12. CONCLUSION

The main objective of this project was to effectively recognise handwritten digits. We have used the MNIST dataset for training and testing along with few user inputs. The proposed CNN algorithm addresses the feature extraction and correct classification approach and well in terms of accuracy and time complexity. The overall highest accuracy 96.95% is achieved in the recognition process. The system has multiple future scopes and can be further improved by training using different approaches and use cases.