A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

A PROJECT REPORT

Submitted by

312419104017 ARTHIYA A P
312419104022 ATHERAI C
312419104027 BUJITHA RA
312419104035 DHARANI V

TEAM ID :PNT2022TMID28048

In partial fulfillment for the award of the certification of

IBM PROJECT

IN

Artificial Intelligence

1. INTRODUCTION

1.1 Project Overview

Handwriting recognition algorithms have depended heavily on existing information and customized features. It is difficult to train an optical character recognition (OCR) system using these requirements. Deep learning approaches are the main focus of handwriting recognition research, which has recently produced ground-breaking results. However, the exponential increase in the volume of handwritten information and the accessibility of enormous computing capacity necessitate an enhancement in recognition rate and warrant additional study. Convolutional neural networks (CNNs) are the most successful method for resolving handwriting recognition issues because they are very good at understanding the layout of handwritten characters and words in ways that facilitate the automatic extraction of distinctive features. In the proposed study, we explore different design alternatives for CNN-based handwritten digit identification, including layer count, receptive field, stride size, padding, kernel size, and dilution. In order to attain accuracy even greater than ensemble architectures, as well as decreased operational complexity and expense, a CNN architecture is presented. Additionally, we provide a suitable arrangement of learning parameters for creating a CNN that enables us to set a new absolute record for categorizing MNIST handwritten digits.

1.2 Purpose

One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include form data entry, bank check processing, postal mail sorting, and others. The capacity to create an effective algorithm that can recognise handwritten digits given by users via a tablet, scanner ,and other digital devices is at the core of the issue.

2. LITERATURE SURVEY

2.1 Existing problem

S. No	Paper name, Author & publication year	Proposed framework	Algorithm Used	Limitations & Future work
1	"Multi-Language Handwritten Digits Recognition based on Novel Structural Features" - Jaafar M. Alghazo et al. (2019)	The proposed approach has been evaluated on six prominent languages. 65 local structural characteristics are recovered in total, and multiple classifiers are utilized to assess number recognition.	AIRS, MLP,Logistic Regression, Random Forest	Fuzzy logic might be used in the future to eliminate misunderstanding between various digits.
2	"Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)" Savita Ahlawat,Amit Choudhary et al. (2020)	The goal of the proposed work is to investigate different design alternatives, such as the number of layers, stride size, receptive field, kernel size for CNN-based handwritten digit recognition .In order to enhance the performance of handwritten digit recognition, several SGD optimization strategies were used. Uses pure CNN.	CNN, SGD	Ensemble CNN models can give higher accuracy than pure CNN models.

3	“An Effective and Improved CNN-ELM Classifier for Handwritten Digits Recognition and Classification” - Saqib Ali, Jianqiang Li et al. (2020)	The proposed model is a self-developed handwritten digit recognition system built on the symmetrical CNN-ELM method, which has high training precision and runs quickly. The findings show that the CNN-ELM-DL4J strategy surpasses the traditional CNN models in terms of accuracy and computing efficiency.	CNN-ELM	By expanding/changing the dataset images and/or further configuring the network with the right parameters, experimental results can be made more effective and efficient.
4	“Combination of Convolutional Neural Network Architecture and its Learning Method for Rotation-Invaria nt Handwritten Digit Recognition” Kazuya Urazoe et al. (2020)	The proposed model introduced a multi-task learning (MTL) algorithm for rotation invariance. This letter describes how our past work may be applied to common CNN designs.	CNN	_
5	“Handwritten Digit Recognition of MNIST dataset using Deep Learning state-ofthe-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) ” - Drishti Beohar et al. (2021)	In this research, the two primary Deep Learning algorithms, Artificial Neural Network and Convolutional Neural Network, are compared in terms of feature extraction and classification phases of recognition. On the MNIST dataset, the algorithms were trained utilizing categorical crossentropy loss and the ADAM optimizer.	ANN, CNN	The drawback of the CNN over the ANN is that it requires more time and CPU resources.

6	“An improved faster-RCNN model for handwritten character recognition” Saleh Albahli et al. (2021)	Introduced an enhanced Faster-RCNN framework using DenseNet-41for feature computation, which improves performance in identifying tiny objects while reducing training and testing time and complexity.	DenseNet-41, Deep learning, Faster-RCNN	Was tested over a simple dataset like MNIST dataset.
7	“A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach “ - Ali Abdullah Yahya et al. (2021)	The feature extraction task is carried out by the convolution layers, maxpooling layers, and batch normalization layers in the proposed CNN architecture. The feature mappings learnt from these layers are then used to categorize the picture score in the input layer.	CNN	There are about 1% of the values that occur outside the main diagonal, implying that the method still yields some mislabeled components. The primary reason for failure is that images vary in size and style, making categorisation even more difficult.
8	“Realtime Handwritten Digit Recognition Using Keras Sequential Model and Pygame” - K. Senthil Kumar et al. (2021)	In this research, CNN and Pygame are used to construct an expanded application of handwritten detection, — in other words, real-time validation of handwritten numbers. This project's classifier is a sequential model using a 4-layer CNN.	CNN	Various methods, such as a multilayer perceptron, can be used to improve the overall process.
9	“Handwritten Digit Recognition Using Deep Learning” Gaganashree J. S.Padmashali et al. (2021)	When attempting to tackle this problem, other issues occur. Handwritten numerals do not always have the same size, thickness, direction, or location relative to the edge. The major purpose of this study is to improve the way of defining perceptual patterns and characterization approaches	CNN, ANN	It is difficult to determine which handwriting traits accurately describe each concerned class.

		to include the handwritten digits.		
10	“Convolutional Vision Transformer for Handwritten Digit Recognition” Vanita Agrawal et al. (2022)	A deep learning method that uses attention processes to evaluate the significance of factors	CNN, ViT	To recognize digit strings, the model must be expanded.

2.2 References

[1] J. M. Alghazo, G. Latif, L. Alzubaidi, and A. Elhassan, “Multi-language handwritten digitsrecognition based on novel structural features,” Journal of Imaging Science and Technology, vol. 63, no. 2, 2019, DOI: 10.2352/J.IMAGINGSCI.TECHNOL.2019.63.2.020502.

[2] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved Handwritten DigitRecognition Using Convolutional Neural Networks (CNN),” Sensors 2020, Vol. 20, Page 3344, vol. 20, no. 12, p. 3344, Jun. 2020, DOI: 10.3390/S20123344.

[3] S. Ali, J. Li, Y. Pei, M. S. Aslam, Z. Shaukat, and M. Azeem, “An Effective and ImprovedCNN-ELM Classifier for Handwritten Digits Recognition and Classification,” Symmetry 2020, Vol. 12, Page 1742, vol. 12, no. 10, p. 1742, Oct. 2020, DOI: 10.3390/SYM12101742.

[4] K. Urazoe, N. Kuroki, T. Hirose, and M. Numa, “Combination of Convolutional NeuralNetwork Architecture and its Learning Method for Rotation-Invariant Handwritten Digit Recognition,” IEEJ Transactions on Electrical and Electronic Engineering, vol. 16, no. 1, pp. 161–163, Jan. 2021, DOI: 10.1002/TEE.23278.

[5] D. Beohar and A. Rasool, “Handwritten digit recognition of MNIST dataset using deeplearning state-of-the-art artificial neural network (ANN) and Convolutional Neural Network (CNN),” 2021 International Conference on Emerging Smart Computing and Informatics, ESCI 2021, pp. 542–548, Mar. 2021, DOI: 10.1109/ESCI50559.2021.9396870.

[6] S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, “An improved faster-RCNN model forhandwritten character recognition,” Arab J Sci Eng, vol. 46, no. 9, pp. 8509–8523, Sep. 2021, DOI: 10.1007/S13369-021-05471-4.

[7] A. A. Yahya, J. Tan, and M. Hu, “A Novel Handwritten Digit Classification System Based onConvolutional Neural Network Approach,” Sensors 2021, Vol. 21, Page 6273, vol. 21, no. 18,p. 6273, Sep. 2021, DOI: 10.3390/S21186273.

2.3 Problem Statement Definition

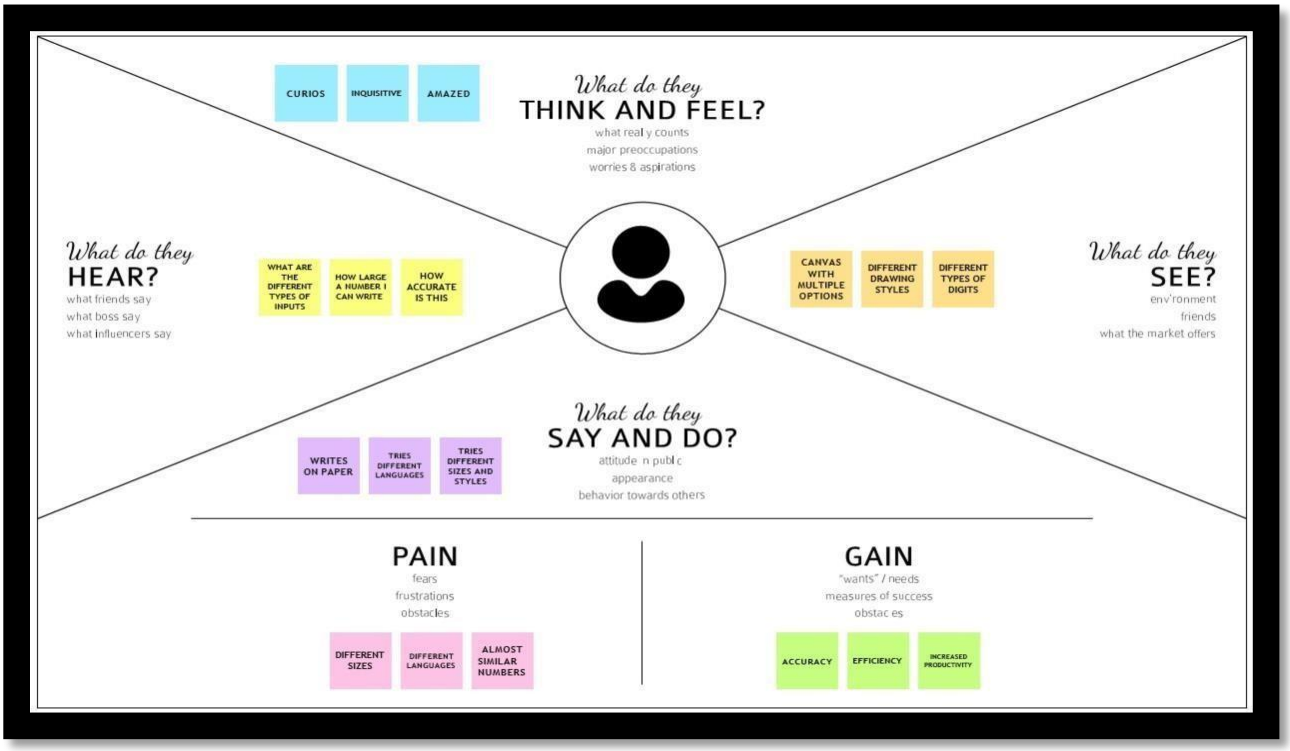
Character handwriting recognition has been around since the 1980s. Handwritten digit recognition using a classifier offers a wide range of applications, including digital digit recognition on PC devices, recognising zip codes on mail, handling bank cheque amounts, numeric portions in structures filled out by hand (for example, tax forms), and so on. There are several difficulties encountered while attempting to address this problem. The digits are not necessarily the same height, width, orientation, or location with respect to the margins. The primary goal was to implement a pattern characterisation approach for perceiving handwritten digits using the MNIST data collection of photographs of handwritten digits (0 - 9). Machine Learning provides a variety of approaches for reducing human effort in detecting manually typed numbers. Deep Learning is a technology that educates computers to do what people do naturally: learning via examples. Human efforts in seeing, learning, recognising, and many other areas can be reduced by using deep learning approaches. The machine learns to do classification tasks from images or the text of any document using deep learning. Models using deep learning can achieve state-of-the-art accuracy, outperforming humans.

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Clerk in bank	Recognize The amount in digit written in check	Due to varied Handwriting Recognising the digits is Difficult	Of the similarities between digits such as 1and 7,3and8,2and7	Confused and Perplexed

PS-2	RTO Inspector	Recognize the number written in number plate	Recognising the number written in number plate	Of blurred image and poor image quality	Overstressed

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

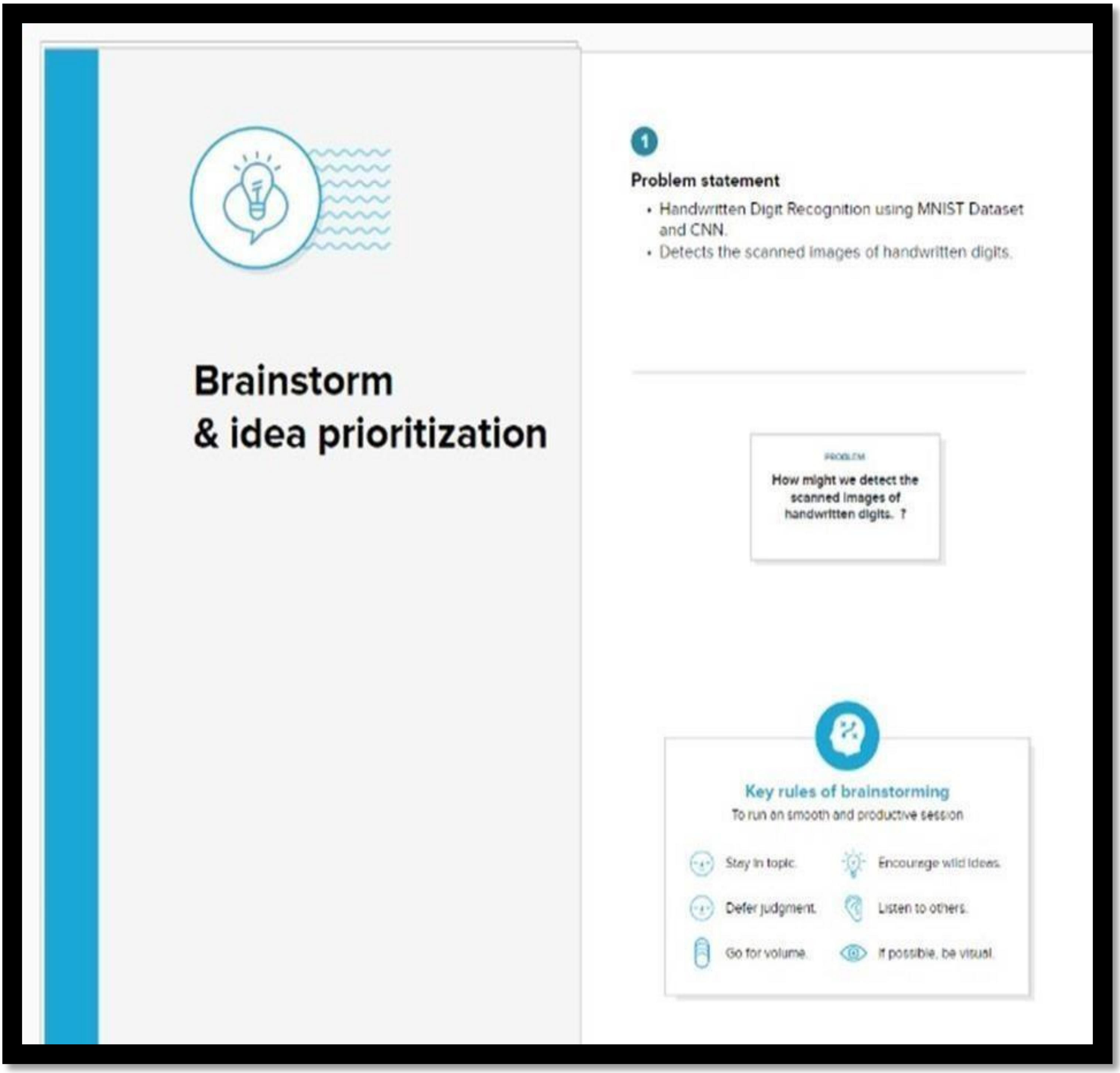


3.2 Ideation & Brainstorming

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

Step-2: Brainstorm, Idea Listing and Grouping

Brainstorm and Idea Listing,Grouping



Brainstorm & idea prioritization

1

Problem statement

- Handwritten Digit Recognition using MNIST Dataset and CNN.
- Detects the scanned images of handwritten digits.

PROBLEM

How might we detect the scanned images of handwritten digits. ?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

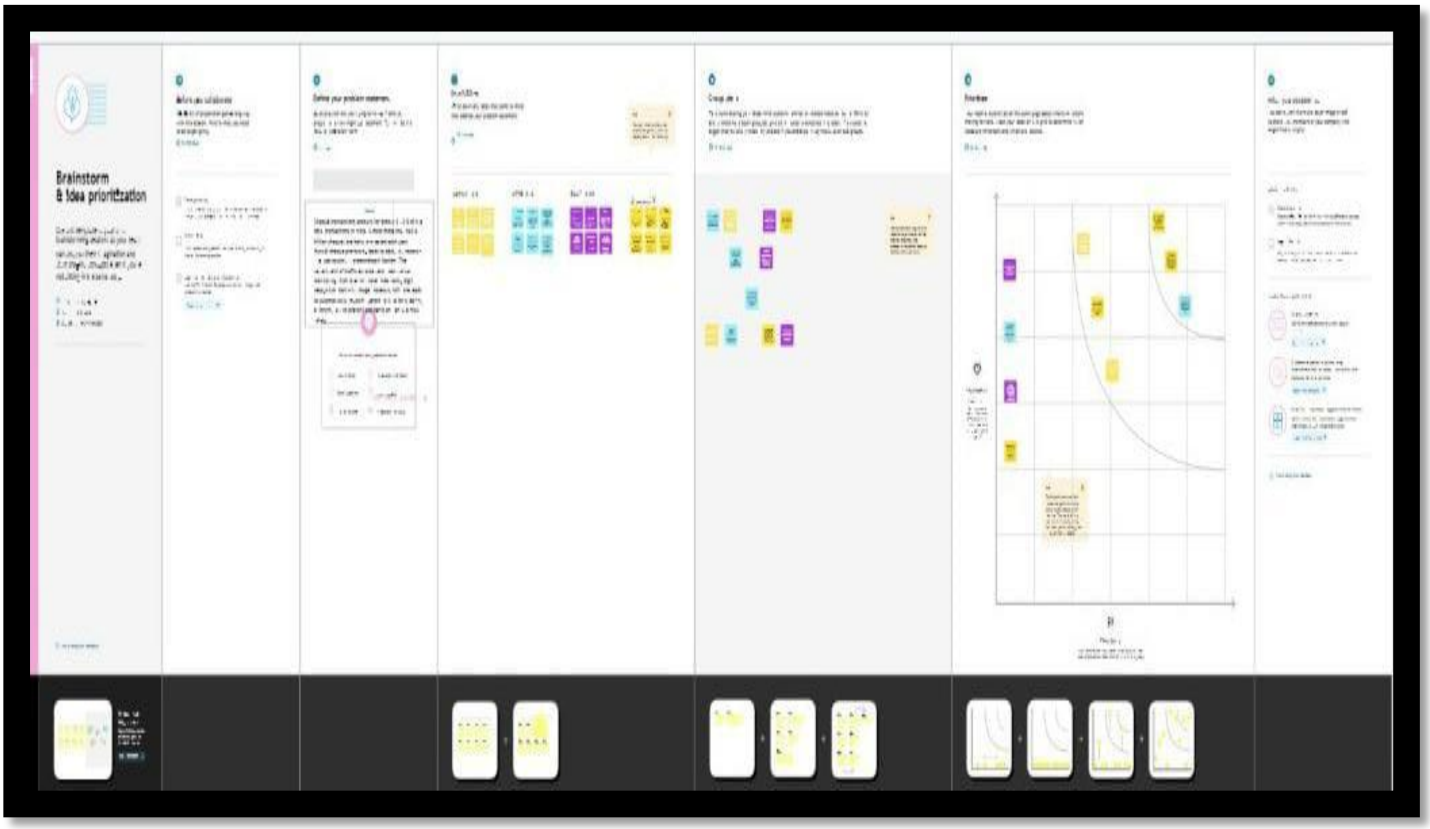
Encourege wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.



Brainstorm
& Idea prioritization

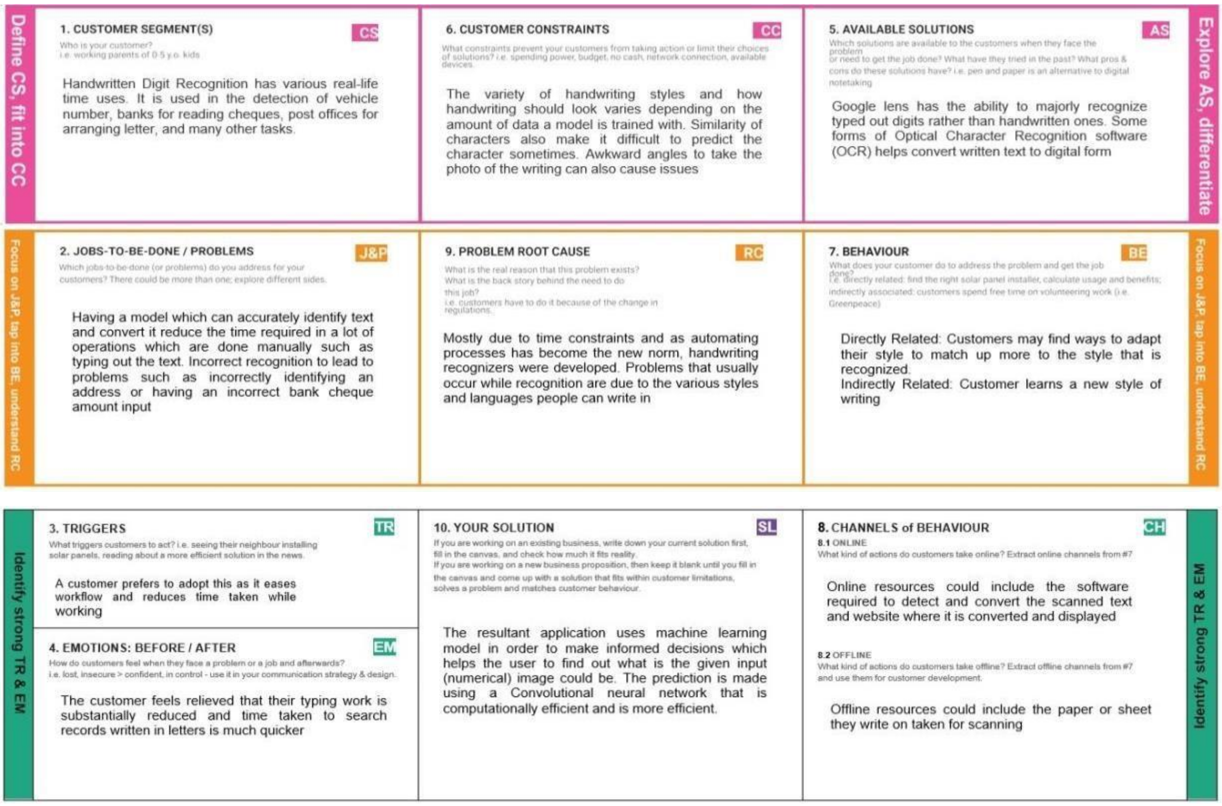
3.2 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Detect the scanned images of handwritten digits.

2.	Idea / Solution description	● Dataset - MNIST ● CNN - vgg16 cnn ● Optimiser - S-GD Using MNIST dataset and vgg16 CNN, we recognise the handwritten digits and optimize the solution.
3.	Novelty / Uniqueness	To achieve a highly reliable and accurate model through hyperparameter tuning and various model optimization techniques
4.	Social Impact / Customer Satisfaction	● Editable ● Easy translation ● Easy to scan and recognise digits ● Easy to use and understand ● Easy to calculate ● Supports multiple languages
5.	Business Model (Revenue Model)	● Data analytics ● Statistics ● Future prediction ● Data entry

6.	Scalability of the Solution	The model is scalable from a dataset training perspective. We can train huge amounts of Hand-written image data by converting them into .npy file format which would facilitate easy storing, retrieving and processing.

3.3 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

Handwritten Digit Recognition has various real-life time uses. It is used in the detection of vehicle number, banks for reading cheques, post offices for arranging letter, and many other tasks.

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 variety of handwriting styles and how handwriting should look varies depending on the amount of data a model is trained with. Similarity of characters also make it difficult to predict the character sometimes. Awkward angles to take the photo of the writing can also cause issues

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

Google lens has the ability to majorly recognize typed out digits rather than handwritten ones. Some forms of Optical Character Recognition software (OCR) helps convert written text to digital form

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.

Having a model which can accurately identify text and convert it reduce the time required in a lot of operations which are done manually such as typing out the text. Incorrect recognition to lead to problems such as incorrectly identifying an address or having an incorrect bank cheque amount input

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.

Mostly due to time constraints and as automating processes has become the new norm, handwriting recognizers were developed. Problems that usually occur while recognition are due to the various styles and languages people can write in

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)

Directly Related: Customers may find ways to adapt their style to match up more to the style that is recognized.
Indirectly Related: Customer learns a new style of writing

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.

A customer prefers to adopt this as it eases workflow and reduces time taken while working

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.

The customer feels relieved that their typing work is substantially reduced and time taken to search records written in letters is much quicker

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.

The resultant application uses machine learning model in order to make informed decisions which helps the user to find out what is the given input (numerical) image could be. The prediction is made using a Convolutional neural network that is computationally efficient and is more efficient.

8. CHANNELS of BEHAVIOUR CH

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

Online resources could include the software required to detect and convert the scanned text and website where it is converted and displayed

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

Offline resources could include the paper or sheet they write on taken for scanning

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)

FR-1	MNIST Dataset	The Modified National Institute of Standards and Technology dataset (MNIST) database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples.
FR-2	Data preprocessing	Improves the image by doing some operations to the input image to prepare it for segmentation.
FR-3	GUI	Enables the user to insert a handwritten image and receive the digits in digital form. designed to facilitate virtualization.
FR-4	Image Data	The ability of a computer to recognise human handwritten digits from various sources, such as images, documents, touch screens, etc., and classify them into ten recognised classes is known as handwritten digit recognition (0-9). This has received a great deal of research in the field of deep learning.
FR-5	Digit Classifier Model	Utilize the MNIST collection of handwritten digits to train a convolutional network to predict a digit from an image. Assemble the data for training and validation first.
FR - 6	Evaluation	Ensure that the digit is correctly recognised by the model and produces the accurate output.

FR - 7	Website	The code, graphics, and other components of a website are made available online by web hosting. Every website is hosted by a server. The amount of server space provided to a website depends on the hosting type. The four primary types of hosting are shared, dedicated, VPS, and reseller.

4.2 Non-Functional requirements

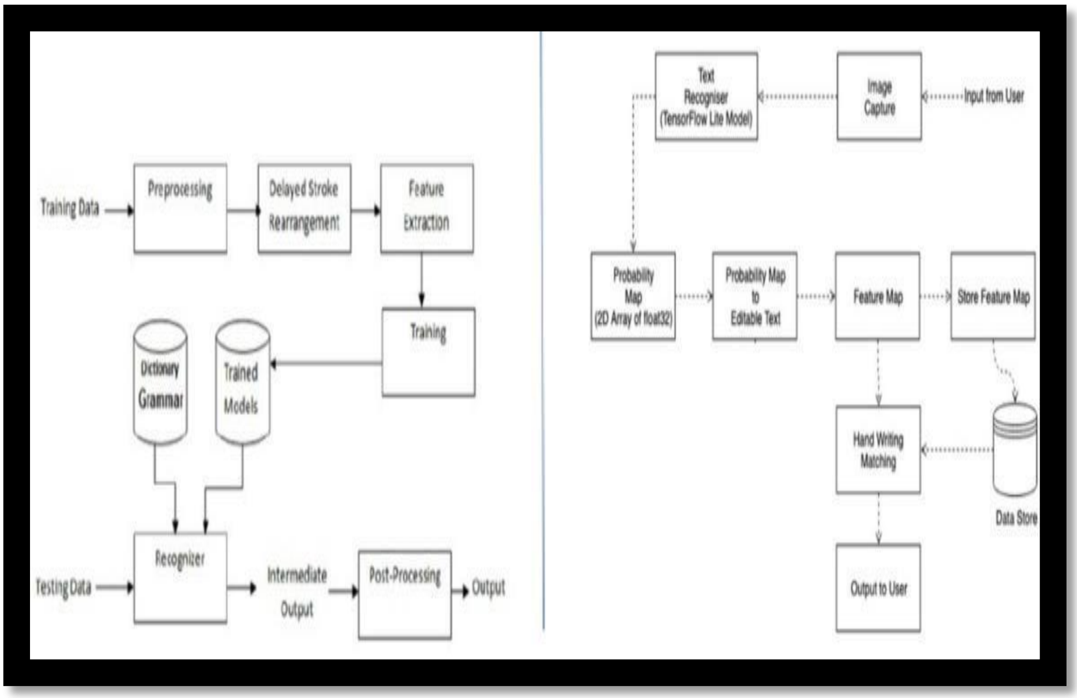
Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	To accurately recognise and comprehend handwritten digits mechanically.
NFR-2	**Security**	● In addition to classifying the digit, the algorithm also generates a full description of the instantiation parameters, which could disclose details like the writing style. ● The generative models are capable of segmentation driven by recognition.

NFR-3	**Reliability**	● The neural network makes use of the samples to automatically determine rules for reading handwritten digits. By increasing the number of training instances, the network may also learn more about handwriting and hence improve its accuracy. ● To recognize handwritten numbers, a variety of methods and algorithms can be employed, including Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc.
NFR-4	**Performance**	High, as deep learning models are created using artificial neural networks that are trained on the training set of images. employing the CNN algorithm for quick prediction.
NFR-5	**Availability**	Anyone can quickly access the system through a web application, making it very accessible for desktop and mobile browsers.
NFR-6	**Scalability**	Works with various other datasets with different languages and writing styles.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

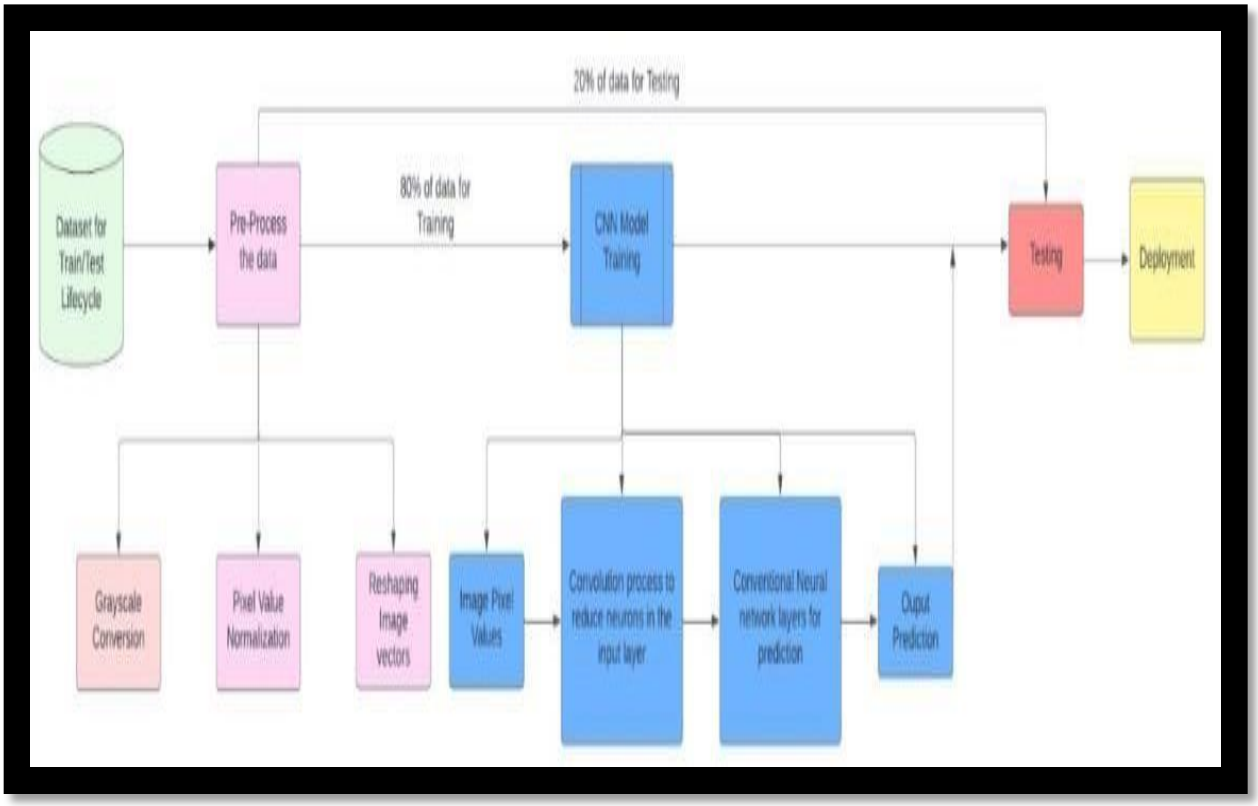


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with the application using the GUI provided, either through real time writing or uploading images from drive, local etc.	Flask, CSS, JavaScript
2.	Application Logic-Pre processing	Pre-processing the data	Python

3.	Application Logic-Model Training and Testing	Training the model on the processed data	Python
4.	Application Logic-3	Deploying the application on IBM cloud and incorporating customer care using Assistant	IBM Watson STT service, IBM Watson Assistant
5.	Database	To store user credentials and image information	MySQL, NoSQL, etc.
6.	Cloud Database	Process data quickly and efficiently during response	IBM DB2, IBM Cloudant etc.
7.	File Storage	Storing the data for training as well as user information	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Machine Learning Model	To recognize digits from handwritten data	Object Recognition Model, etc.

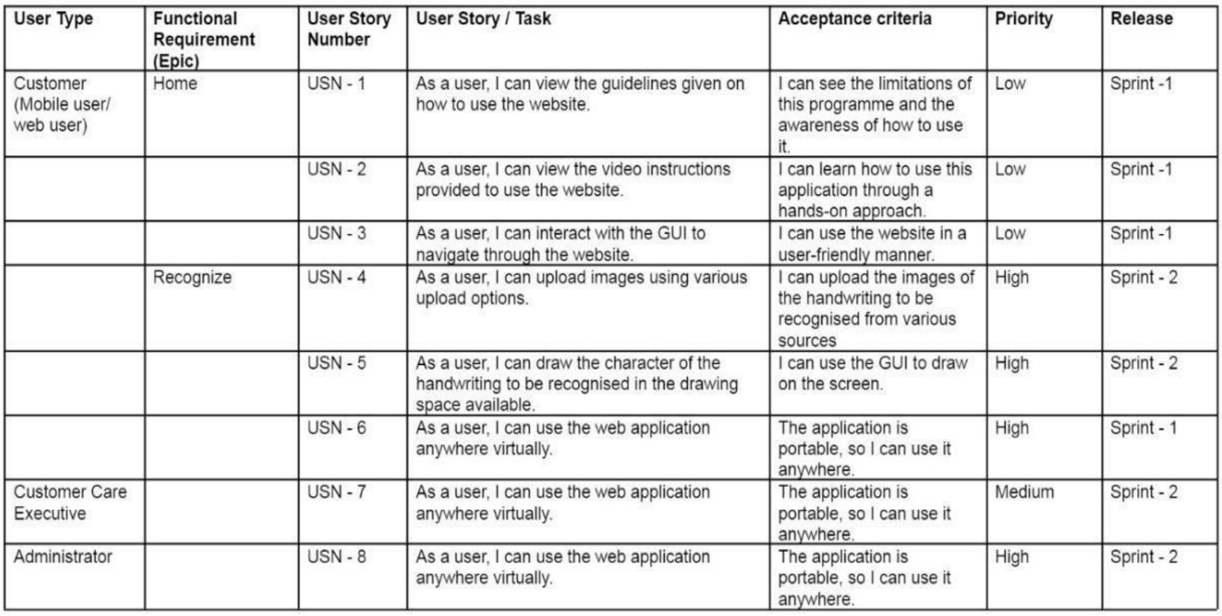
9.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Security Implementations	Various security / access controls implementations have been done for protecting user data	SHA-256, Encryptions, IAM Controls, OWASP etc.
2.	Scalable Architecture	The application can be easily scaled up to handle increased traffic. The model deployed can also be scaled to multiple languages based on increased training on differing data	3-tier, Microservices, Python
3.	Availability	The application will be highly available because of the light weight processing on the frontend. The cloud deployment of the model will help in making quick predictions on the user data	IBM Cloud

4.	Performance	Standard CNN implementations (CNN VGG-16) on MNIST data can easily give 98-99 % accuracy on the training data, hence, almost classifies every single handwritten digit correctly.	Python, Performance metrics

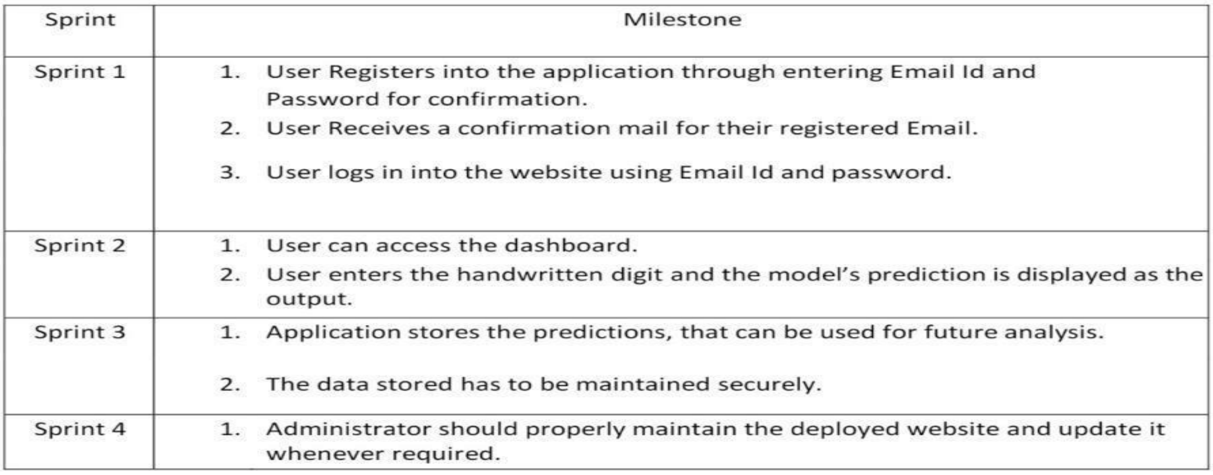
5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user/ web user)	Home	USN - 1	As a user, I can view the guidelines given on how to use the website.	I can see the limitations of this programme and the awareness of how to use it.	Low	Sprint -1
		USN - 2	As a user, I can view the video instructions provided to use the website.	I can learn how to use this application through a hands-on approach.	Low	Sprint -1
		USN - 3	As a user, I can interact with the GUI to navigate through the website.	I can use the website in a user-friendly manner.	Low	Sprint -1
	Recognize	USN - 4	As a user, I can upload images using various upload options.	I can upload the images of the handwriting to be recognised from various sources	High	Sprint - 2
		USN - 5	As a user, I can draw the character of the handwriting to be recognised in the drawing space available.	I can use the GUI to draw on the screen.	High	Sprint - 2
		USN - 6	As a user, I can use the web application anywhere virtually.	The application is portable, so I can use it anywhere.	High	Sprint - 1
Customer Care Executive		USN - 7	As a user, I can use the web application anywhere virtually.	The application is portable, so I can use it anywhere.	Medium	Sprint - 2
Administrator		USN - 8	As a user, I can use the web application anywhere virtually.	The application is portable, so I can use it anywhere.	High	Sprint - 2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



Sprint	Milestone
Sprint 1	1. User Registers into the application through entering Email Id and Password for confirmation. 2. User Receives a confirmation mail for their registered Email. 3. User logs in into the website using Email Id and password.
Sprint 2	1. User can access the dashboard. 2. User enters the handwritten digit and the model's prediction is displayed as the output.
Sprint 3	1. Application stores the predictions, that can be used for future analysis. 2. The data stored has to be maintained securely.
Sprint 4	1. Administrator should properly maintain the deployed website and update it whenever required.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Arthiya Atherai
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password.	5	High	Bujitha Dharani
Sprint-1	Data Collection & Pre-processing	USN-3	As a user, I can upload any kind of image with the pre-processing step involved in it.	10	High	Arthiya Bujitha
Sprint-2	Upload input Image	USN-4	As a user, I am able to input the images of digital documents, handwritten documents or images to the application.	5	Medium	Atherai Dharani

Sprint-2	Building the ML model	USN-5	As a user, I will getan application within ML model which provides high accuracy of recognized handwritten digits.	20	High	Atherai Bujitha
Sprint-2		USN-6	As a user, I can pass the handwritten digitimage for recognizing the digit.	5	Medium	Arthiya Dharani
Sprint-3	Building the UI App	USN-7	As a user, I will upload the handwritten digit image to the application by clicking an upload button.	3	Medium	Arthiya Atherai
Sprint-3		USN-8	As a user, I can know the details of the fundamental usage of the application	3	Low	Bujitha Dharani
Sprint-3		USN-9	As a user, I can see the predicted / recognized digits in the application.	4	High	Arthiya Atherai Bujitha Dharani

Sprint-4	Train and deploy the model in IBM Cloud	USN-10	As a user, I can access the web application and make use of the product from anywhere.	10	High	Arthiya Atherai Bujitha Dharani
Sprint-4	Recognize the digit	USN-11	As a user I am able to get the recognised digit as output from the images of digital documents or images.	10	Medium	Arthiya Atherai Bujitha Dharani

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022

Sprint-4	20	6 Days	14Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

- Using CNN Model in our Project : CNN is basically a model known to be Convolutional Neural Network and in recent times it has gained a lot of popularity because of its usefulness. CNN usesmultilayer perceptrons to do computational work
- CNN uses relatively little pre-processing compared to other image classification algorithms. This means the network learns through filters that in traditional algorithms were hand engineered. So, for the image processing tasks CNNs are the best-suited option.

7.2 Feature 2

- Using Flask application in our Project : Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

8. TESTING

8.1 Test Cases

Expected Result
Predict page should open
Home page should open
Predict page should open properly
Choosen File should be uploaded
A preview of the image should be viewed in the predict page
It should move from the predicted page to the result page.
It should validate the file uploaded
Check whether the file is reshaped and stored as variable in main.py

Check whether the file loads the dataset and predict the digit
Check whether the result page is displayed.
Check whether the result page displays the correct answer
Check whether the result page contains the upload again button
Test Scenario
Verify whether the user is able to use the recognize button
Verify whether the user is able to use the home button
Verify whether the predict page is opening
Verify if we are able to upload the file to be recognized
Verify whether user is able to view the preview the image which is being uploaded
Verify whether user is able use the recognize button
Verify whether it validates the file
Verify whether it accepts the file uploaded
Verify whether it loads the dataset and predict the digit
Verify whether the result is displayed
Verify whether the result displayed is correct

8.2 User Acceptance Testing

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	4	2	3	9
Duplicate	0	0	3	0	3
External	0	0	0	1	1

Fixed	0	4	5	4	13
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	1	1
Won't Fix	0	0	0	1	1
Totals	0	8	10	10	26

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

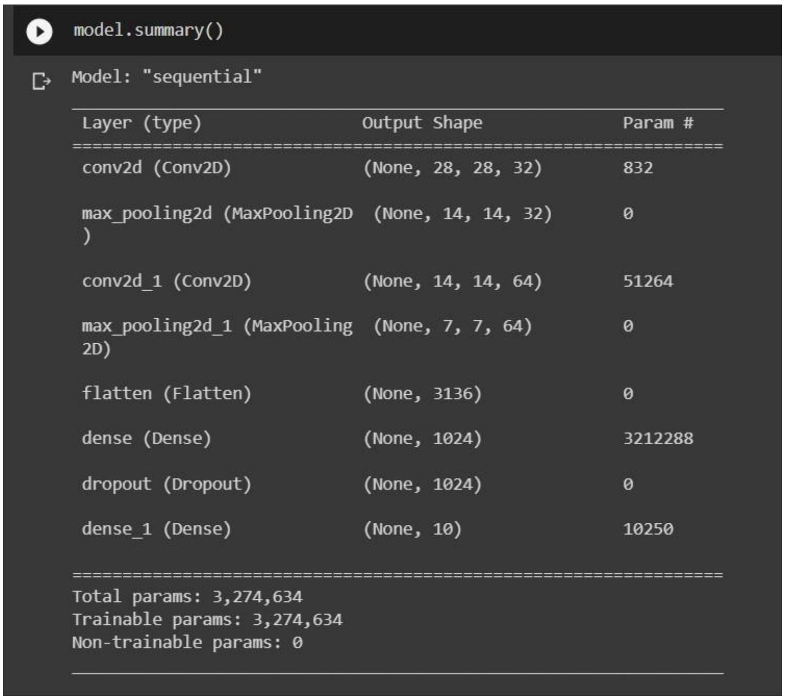
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	10	0	0	10
Security	1	0	0	1
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metric Model

Performance Testing:

S.No.	Parameter	Values	Screenshot

<table>
<tr>
<td>1.</td>
<td>Model Summary</td>
<td>-</td>
<td>


<pre>
model.summary()

Model: "sequential"
____
 Layer (type)   Output Shape   Param #
===
 conv2d (Conv2D)   (None, 28, 28, 32)   832

 max_pooling2d (MaxPooling2D (None, 14, 14, 32)   0
)

 conv2d_1 (Conv2D)   (None, 14, 14, 64)   51264

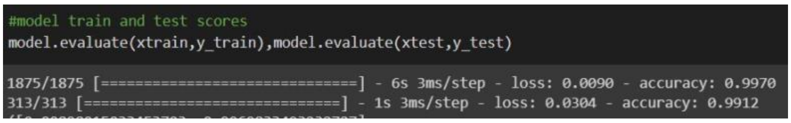
 max_pooling2d_1 (MaxPooling (None, 7, 7, 64)   0
 2D)

 flatten (Flatten)   (None, 3136)   0

 dense (Dense)   (None, 1024)   3212288

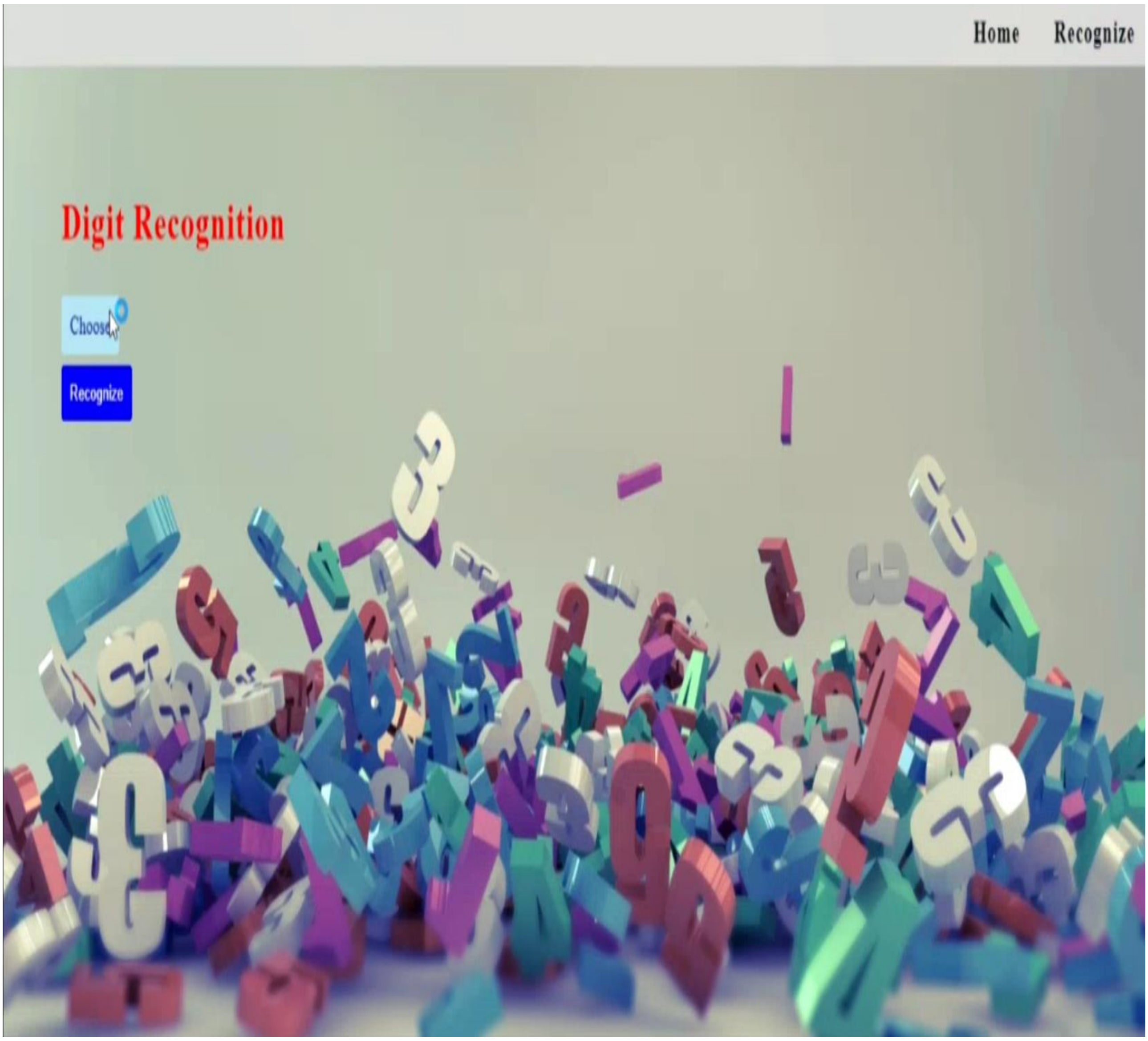
 dropout (Dropout)   (None, 1024)   0

 dense_1 (Dense)   (None, 10)   10250

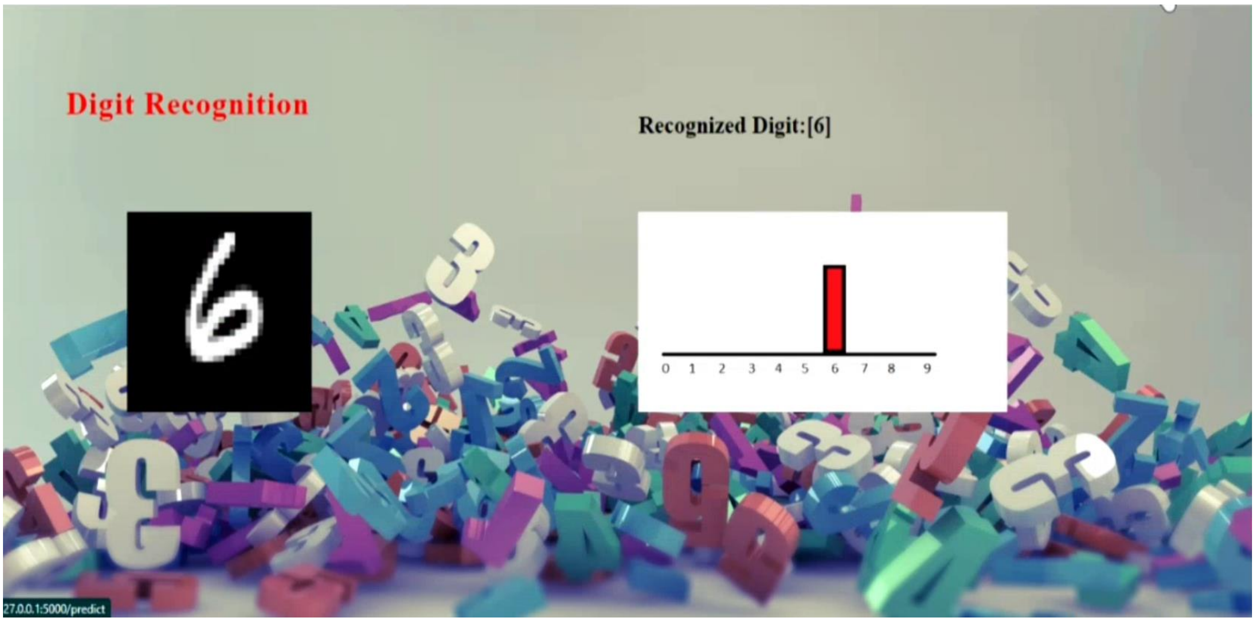
===
Total params: 3,274,634
Trainable params: 3,274,634
Non-trainable params: 0
____
</pre>
</td>
</tr>
<tr>
<td>2.</td>
<td>Accuracy</td>
<td>Training Accuracy -99.7%
Validation Accuracy – 99.12%</td>
<td>


<pre>
#model train and test scores
model.evaluate(xtrain,y_train),model.evaluate(xtest,y_test)

1875/1875 [==============================] - 6s 3ms/step - loss: 0.0090 - accuracy: 0.9970
313/313 [==============================] - 1s 3ms/step - loss: 0.0304 - accuracy: 0.9912
</pre>
</td>
</tr>
</table>



Home
Recognize
Digit Recognition
Choose
Recognize



Digit Recognition

Recognized Digit:[6]

10. ADVANTAGES & DISADVANTAGES

Handwriting Recognition has many advantages that made it grow rapidly in the technology world now. There are many different kinds of technologies that enable others to take advantage of handwriting recognition. The way this work was when people write letters a different way and they let the computer know what the intended letter was and change in to a text document

Certain cell phones have the handwriting recognition system in it. The advantage of this is that it allows people to write on their cell phones using a stylus and then the phone software translates the written words to the phone in text.

The disadvantage of handwriting recognition technologies is that not everyone's handwriting is the same, everyone writes differently. This starts the problem in the handwriting recognition technology when it needs to translate a person's handwriting into type and because of this problem many companies failed to perform well because many couldn't effectively use the program well enough.

11. CONCLUSION

The Handwritten Digit Recognition using Deep learning methods has been implemented. CNN has been trained and tested on the same data in order to acquire the comparison between the classifiers. Utilizing these deep learning techniques, a high amount of accuracy can be obtained. Compared to other research methods, this method focuses on which classifier works better by improving the accuracy of classification models by more than 99%. Using Keras as backend and Tensorflow as the software, a CNN model is able to give accuracy of about 99%

12. FUTURE SCOPE

The proposed system takes 28x28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build a Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output.

13. APPENDIX

main.py

```
import os import urllib import uuid from flask import Blueprint,
redirect, render_template, url_for, request from flask_login import
current_user, login_required, logout_user from
tensorflow.keras.preprocessing.image import load_img , img_to_array from
tensorflow.keras.models import load_model from PIL import Image import
numpy as np import cv2


# Blueprint Configuration
main_bp = Blueprint(
   "main_bp",___name__, template_folder="templates", static_folder="static"
)


BASE_DIR = os.path.dirname(os.path.abspath(_file_)) model =
load_model(os.path.join(BASE_DIR , 'CNN_MNIST_v1.h5'))



ALLOWED_EXT = set(['jpg' , 'jpeg' , 'png' , 'jfif']) def
allowed_file(filename):
    return '.' in filename and \ filename.rsplit('.',
           1)[1] in ALLOWED_EXT


classes = ['0','1', '2' , '3' , '4' ,'5' ,'6', '7' ,'8' ,'9']
def predict(filename , model):
    print(filename) img = Image.open(filename).convert('L') img_array
    = cv2.imread(filename) new_array = cv2.cvtColor(img_array,
    cv2.COLOR_BGR2GRAY) new_array
    = cv2.resize(new_array, (28,28)) img=new_array.reshape(1,28,28,1)
    result = model.predict(img) final_pred = classes[np.argmax(result)]
    classes_text=["Zero","One","Two","Three","Four","Five","Six","Seven","Eight","Nine"
    ] return_value=str(final_pred)+" ["+classes_text[int(final_pred)]+"]" print(type(final_pred))



    return return_value


@main_bp.route("/", methods=["GET"])
@login_required def
dashboard():
    """Logged-in User Dashboard.""" return
    render_template(
    "dashboard.jinja2", title="Dashboard",
    template="dashboard-template",
    current_user=current_user, body="Handwritten
    Digit Recognition
    - Prediction Page"
    )


@main_bp.route('/success' , methods = ['GET' , 'POST'])
@login_required def success(): error = '' target_img =
os.path.join(os.getcwd() , 'user_login\\static\\images') if request.method
== 'POST':
        if(request.form):
            link = request.form.get('link') try
            :

                resource = urllib.request.urlopen(link)
                unique_filename = str(uuid.uuid4()) filename =
                unique_filename+".jpg" img_path =
        os.path.join(target_img , filename) output =
        open(img_path , "wb")
        output.write(resource.read()) output.close()
        img = filename



        result = predict(img_path,model)


    except Exception as e :
        print(str(e)) error = 'This image from this site is not accesible or inappropriate
        input' if(len(error) == 0):


        return render_template("success.jinja2",title="Prediction Page",
                        template="dashboard-template", current_user=current_user,
                        body="Handwritten Digit
                        Recognition - Prediction Page", img  = img , predictions
                        = result)
    else: return


redirect("/") elif (request.files):


    file = request.files['file'] if file and allowed_file(file.filename):
    print(target_img) print(file.filename)
    file.save(os.path.join(target_img , file.filename)) img_path =
    os.path.join(target_img , file.filename) img = file.filename


        result = predict(img_path,model)
        else: error = "Please upload images of jpg , jpeg and png extension
          only"

        if(len(error) == 0):
          return render_template('success.jinja2' , title="Prediction Page", img = img , predictions =
result) else: return redirect("/")


@main_bp.route("/logout"
) @login_required def
logout():
    """User log-out logic.""" logout_user()
    return redirect(url_for("auth_bp.login"))
```

auth.py:

```
from flask import Blueprint, flash, redirect, render_template, request, url_for,session from
flask_login import current_user, login_user

from . import login_manager from .forms
import LoginForm, SignupForm from
.models import User, db from flask import
request


# Blueprint Configuration
auth_bp = Blueprint(
    "auth_bp",__name__, template_folder="templates", static_folder="static"
)


@auth_bp.route("/signup", methods=["GET", "POST"])
```

```
def signup():
    """
    User sign-up page.

    GET requests serve sign-up page.
    POST requests validate form & user creation.
    """ form =
    SignupForm()
    # session['messages'] = "" if form.validate_on_submit():
    existing_user = User.query.filter_by(email=form.email.data).first() if
    existing_user is None:
            user = User( name=form.name.data, email=form.email.data,
                website=form.website.data
            )
            user.set_password(form.password.data)
    db.session.add(user) db.session.commit() # Create new
    user login_user(user) # Log in as newly created user
    return redirect(url_for("main_bp.dashboard")) flash("A
    user already exists with that email address.") return
    render_template( "signup.jinja2", title="Create an
    Account.", form=form, template="signup-page",
    body="Sign up for a user account.",
    )
@auth_bp.route("/login", methods=["GET", "POST"])
def login():
    # session['messages'] = ""
    """
    Log-in page for registered users.
    GET requests serve Log-in page.

    POST requests validate and redirect user to dashboard.

    """

    # Bypass if user is logged in if current_user.is_authenticated:

    return redirect(url_for("main_bp.dashboard"))


    form = LoginForm() # Validate

    login attempt if

    form.validate_on_submit():

        user = User.query.filter_by(email=form.email.data).first() if user

        and user.check_password(password=form.password.data):

            login_user(user) next_page = request.args.get("next") return

    redirect(next_page or url_for("main_bp.dashboard"))

    flash("Invalid username/password combination") return

    redirect(url_for("auth_bp.login")) return render_template(

    "login.jinja2", form=form, title="Log in.",

    template="loginpage", body="Log in with your User account.",

    )



@login_manager.user_loader def

load_user(user_id):

    """Check if user is logged-in upon page load.""" if

    user_id is not None:

        return User.query.get(user_id) return

    None
```

```
@login_manager.unauthorized_handler def
unauthorized():
    """Redirect unauthorized users to Login page."""
    flash("You must be logged in to view that page.") return
    redirect(url_for("auth_bp.login"))
```

dashboard.jinja2:

```
{% extends "layout.jinja2" %}
{% block pagestyles %}
  <link href="{{ url_for('static', filename='dist/css/dashboard.css') }}" rel="stylesheet"
  type="text/css"> <link rel="stylesheet" href="https://use.typekit.net/sus7rlu.css">
{% endblock %}
{% block content %}
  {% if current_user.is_authenticated %}
  <div style="display: flex;justify-content:right;">
    <p>Hi {{ current_user.name }}!</p>
    <a href="{{ url_for('main_bp.logout') }}">Log out</a>
  </div>
  <h1 id="topic">{{ body }}</h1>
  <div class="upload-section">

            <div class="upload-file">
                <form class = "file-form" action="/success" method="post", enctype="multipart/formdata">
                    <input class="file-form-input" type="file" , name = "file"/>
                    <button class="btn btn-success btn-lg">Upload</button>
                </form>
            </div>

            <div class="uploadOR">
                <span>OR</span>
            </div>
            <div class="upload-link">
                <form class = "link-form" action="/success" method="post", enctype="multipart/formdata">
                    <input class="link-form-input" type="text" ,maxlength="1000" , name = "link" placeholder="Paste the image URL(not the site address)"/>
                    <button class="btn btn-success btn-lg">Proceed</button>
                </form>
            </div>
            <div class = "header-content-sub">
                    <p style = "text-align: center; margin-top: 15px;color: red" class="header-content-info">{{error}}</p>
            </div>
            <span>{{error}}</span>
        </div>{% endif %}{% endblock %}
```

GitHub & Project link: https://github.com/IBM-EPBL/IBM-Project-42676-1660704018