A NOVEL METHOD FOR HAND WRITTEN DIGIT RECOGNITION SYSTEM

HX 8001-
Professional Readiness For Innovation, Employability and Entrepreneurship

IBM-Project-15157-1659594516

TEAM ID: PNT2022TMID08334

Submitted by

Neelam Gayitri	(810419104070)
Pakanati Sravya	(810419104074)
Paritala Hanna	(810419104076)
Peram Naga sudha	(810419104079)

in partial fulfilment for the award of the degree
of
BATCHELOR OF ENGINNERING
IN
COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
PERAMBALUR-621212

TABLE OF CONTENTS

ABSTRACT

Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defences etc, and classify. them into 10 predefined classes (0-9). This has been a Content of bottomless- exploration in the field of deep literacy. Number recognition has numerous operations like number plate recognition, postal correspondence sorting, bank check processing, etc. In Handwritten number recognition, we face numerous challenges. because of different styles of jotting of different peoples as it. is not an Optic character recognition. This exploration provides a comprehensive comparison between different machine literacy and deep literacy algorithms for the purpose of handwritten number recognition. For this, we've used Support. Vector Machine, Multilayer Perceptron, and Convolutional. Neural Network. The comparison between these algorithms is carried out on the base of their delicacy, crimes, and testing-training time corroborated by plots and maps that have been constructed using matplotlib for visualization. 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 PyTorch library over the MNIST dataset to recognize handwritten digits. Handwritten character recognition is an extensive exploration area that formerly contains detailed ways of perpetration which include major literacy datasets, popular algorithms, features scaling and point birth styles. MNIST dataset (Modified National Institute of Norms and Technology database) is the subset of the NIST dataset which is a combination of two of NIST's databases Special.

1. INTRODUCTION

In the current age of digitization, handwriting recognition plays an important role in information processing. A lot of information is available on paper, and processing of digital files is cheaper than processing traditional paper files. The aim of a handwriting recognition system is to convert handwritten characters into machine readable formats. The main applications are vehicle license-plate recognition, postal lettersorting 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. It has been suggested that deep neural architectures are more advantageous than shallow neural architectures [1–6]. The key differences are described in Table 1. The deep learning field is ever evolving, and some of its variants are autoencoders, CNNs, recurrent neural networks (RNNs), recursive neural networks, deep belief networks and deep Boltzmann machines. Here, we introduce a convolutional neural network, which is a specific type of deep neural network having wide applications in image classification, object recognition, recommendation systems, signal processing, natural language processing, computer vision, and face recognition. The ability to automatically detect the important features of an object (here an object can be an image, a handwritten character, a face, etc.) without any human supervision or intervention makes them (CNNs) more efficient than their predecessors (Multi layer perceptron (MLP), etc.). The high capability of hierarchical feature learning results in a highly efficient CNN.

1.1 Project Overview

An enormous number of CNN classification algorithms have been proposed in the literature. Nevertheless, in these algorithms, appropriate filter size selection, data preparation, limitations in datasets, and noise have not been taken into consideration. As a consequence, most of the algorithms have failed to make a noticeable improvement in classification accuracy. To address the shortcomings of these algorithms, our paper presents the following contributions: Firstly, after taking the domain knowledge into consideration, the size of the effective receptive field (ERF) is calculated. Calculating the size of the ERF helps us to select a typical filter size which leads to enhancing the classification accuracy of our CNN. Secondly, unnecessary data leads to misleading results and this, in turn, negatively affects classification accuracy. To guarantee the dataset is free from any redundant or irrelevant variables to the target variable, data preparation is applied before implementing the data classification mission. Thirdly, to decrease the errors of training and validation, and avoid the limitation of datasets, data augmentation has been proposed. Fourthly, to simulate the real-world natural influences that can affect image quality, we propose to add an additive white Gaussian noise with s = 0.5 to the MNIST dataset. As a result, our CNN algorithm achieves state-of-the-art results in handwritten digit recognition, with a recognition accuracy of 99.98%, and 99.40% with 50% noise.

1.2 Purpose

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

2. LITERATURE SURVEY

AUTHOR / YEAR / PUBLICATION	TITLE	CONCEPT	ISSUES

Aliya Fathima, S. Geethanjali, M. Janani, Dr .R. Geetha /2007	A Survey on Handwritten Text Recognition Using Deep Learning	The various pre Processing Techniques involved in the text recognition with a variety of pictures ranging from simple written form based documents and documents containing coloured and sophisticated background are dealt in this paper.	Lastly, it has been concluded that using a single method for pre-processing, the image cannot be processed completely.
Chirag Dodiya, DR. Gayatri S Pandi / 2013	Handwritten Recognition	In this paper, the offline handwritten recognition will be done using a Convolutional neural network and TensorFlow.	There has been plenty 0f research done in the field of HCR but still, it is an open problem as we are still lacking in getting the best accuracy.
Yash Pandey ,Bhanu Pratap, Sangras Bhargav, J. Shiva Nandhini /2014	Optical Character Recognition	Handwriting recognition has two basic type existing one is online and other is offline. In this project, by using Linear Support Vector we will present the hand writing recognition system in a very simple and feasible way.	Less Accuracy and takes more time.
T. Wakabayashi and F. Kimura /2007	Handwritten Numeric Recognition	Digit recognition is used in post offices, in banks for reading cheques, for license plate recognition. The digit recognition can be divided into two groups, printed digit recognition and handwritten digit	On the other hand, there are numerous handwriting styles for the same digit; hence more effort is required to find the accurate handwritten digit.

		recognition. Recognition of printed digits is easier compared to the handwritten digit recognition.	
J. Pradeep, E. Srinivasan and S. Himavathi / 2011	Diagonal based feature extraction for handwritten alphabets recognition system using neural network	An off-line handwritten alphabetical character recognition system using multilayer feed forward neural network is described in the paper. Diagonal based feature extraction is introduced for extracting the features of the handwritten alphabets.570 different handwritten alphabetical characters are used for testing	Extraction process is complicated.

2.1 Existing problem

The quality of Hand written digit recognition system depends on the quality of input image that is provided to it. This means that if there are any imperfections in an image, this will have a harder time extracting text from it. This technology has to analyse each image and convert it into text, which can take some time. For example, this might take several seconds to convert a single page of text. This can be a problem if you need to convert a large document into text. Additionally, optical character recognition can be expensive, and it may not be available for all document types. This technology is not 100% accurate, and it can sometimes make mistakes when converting images to text. For example, this might mistake a lowercase "l" for a "1", or a "b" for an "8".

This can cause problems if the text is used for critical purposes, such as in a legal document. One of the main disadvantages of Hand written digit recognition is that sometimes the formatting of the output documents are lost during the process. This can result in text that is difficult to read or difficult to understand. It can introduce errors that can mislead the value of the document.

This recognition can introduce errors, such as incorrectly recognizing a character as a word or line break. One of the problems associated with Hand written digit recognition is a lack of information on some, such as punctuation. There are many punctuation marks that cannot be read by this software because they are too small or non-contiguous, or because they're upside down and backwards.

One of the disadvantages of Hand written digit recognition system is that it may not recognize properly right to left languages. The hand written digit recognition function does not recognize the following language numbers: Japanese, Chinese, Korean, Arabic, and Hebrew. It can only understand the digits. This recognition system cannot be able to recognize the roman numbers

2.2 References

1. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp.
886–893.

2. Lowe, D.G. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 2004, 60, 2. [CrossRef]

3. Xiao, J.; Xuehong, Z.; Chuangxia, H.; Xiaoguang, Y.; Fenghua, W.; Zhong, M. A new approach for stock price analysis and prediction based on SSA and SVM. Int. J. Inf. Technol. Decis. Making 2019, 18, 287–310. [CrossRef]

4. Wang, D.; Lihong, H.; Longkun, T. Dissipativity and synchronization of generalized BAM neural networks with multivariate discontinuous activations. IEEE Trans. Neural Netw. Learn. Syst. 2017, 29, 3815–3827. [PubMed]

5. Kuang, F.; Siyang, Z.; Zhong, J.; Weihong, X. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. Soft Comput. 2015, 19, 1187–1199. [CrossRef]

6. Choudhary, A.; Ahlawat, S.; Rishi, R. A binarization feature extraction approach to OCR: MLP vs. RBF. In Proceedings of the International Conference on Distributed Computing and Technology ICDCIT, Bhubaneswar, India, 6–9 February 2014; Springer: Cham, Switzerland, 2014; pp. 341–346.

7. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 1980,
36, 193–202. [CrossRef]

8. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; LeCun, Y. What is the best multi-stage architecture for object recognition. In Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japam, 29 September–2 October
2009.

9. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Highperformance neural networks for visual object classification. arXiv 2011, arXiv:1102.0183v1.

10. Ciresan, D.C.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. arXiv 2012, arXiv:1202.2745.

11. Niu, X.X.; Suen, C.Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. Pattern Recognit. 2012, 45, 1318–1325. [CrossRef]

12. Qu, X.; Wang, W.; Lu, K.; Zhou, J. Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network. Pattern Recognit. Lett. 2018, 111, 9–15. [CrossRef]
13. Alvear-Sandoval, R.; Figueiras-Vidal, A. On building ensembles of stacked denoising auto-encoding classifiers and their further improvement. Inf. Fusion 2018,
39, 41–52, [CrossRef]

14. Demir, C.; Alpaydin, E. Cost-conscious classifier ensembles. Pattern Recognit. Lett.

2005, 26, 2206–2214. [CrossRef]

15. Choudhary, A.; Ahlawat, S.; Rishi, R. A neural approach to cursive handwritten character recognition using features extracted from binarization technique. Complex Syst. Model. Control Intell. Soft Comput. 2015, 319, 745–771.

16. Choudhary, A.; Rishi, R.; Ahlawat, S. Handwritten numeral recognition using modified BP ANN structure. In Proceedings of the Communication in Computer and Information Sciences (CCIS-133), Advanced Computing, CCSIT 2011, Royal Orchid Central, Bangalore, India, 2–4 January 2011; Springer: Berling/Heildelberg, Germany,
2011; pp. 56–65.

17. Cai, Z.W.; Li-Hong, H. Finite-time synchronization by switching state-feedback control for discontinuous Cohen–Grossberg neural networks with mixed delays. Int. J. Mach. Learn. Cybern. 2018, 9, 1683–1695. [CrossRef]

18. Zeng, D.; Dai, Y.; Li, F.; Sherratt, R.S.; Wang, J. Adversarial learning for distant supervised relation extraction. Comput. Mater. Contin. 2018, 55, 121–136.

19. Long, M.; Yan, Z. Detecting iris liveness with batch normalized convolutional neural network. Comput. Mater. Contin. 2019, 58, 493–504. [CrossRef]

20. Chuangxia, H.; Liu, B. New studies on dynamic analysis of inertial neural networks involving non-reduced order method. Neurocomputing 2019, 325, 283–287.

2.3 Problem Statement Definition

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find

the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

PROBLEM STATEMENT - 1

Handwritten digit recognition system used in Banking sectors

This system can be used to overcome the following issues:

PROBLEM STATEMENT – 2

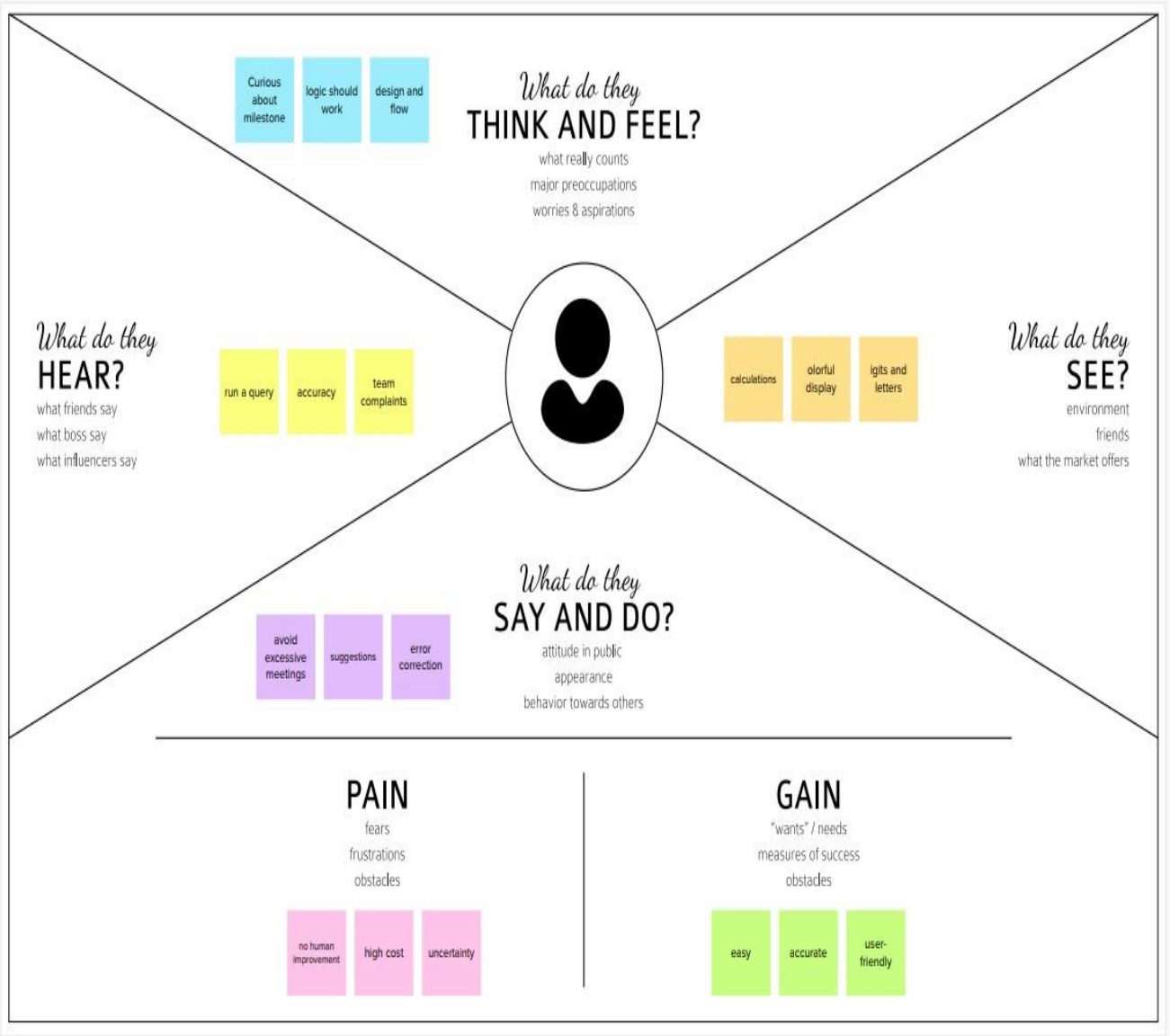
Handwritten digit recognition system used in Data Entry jobs

This system can be used to overcome the following issues:

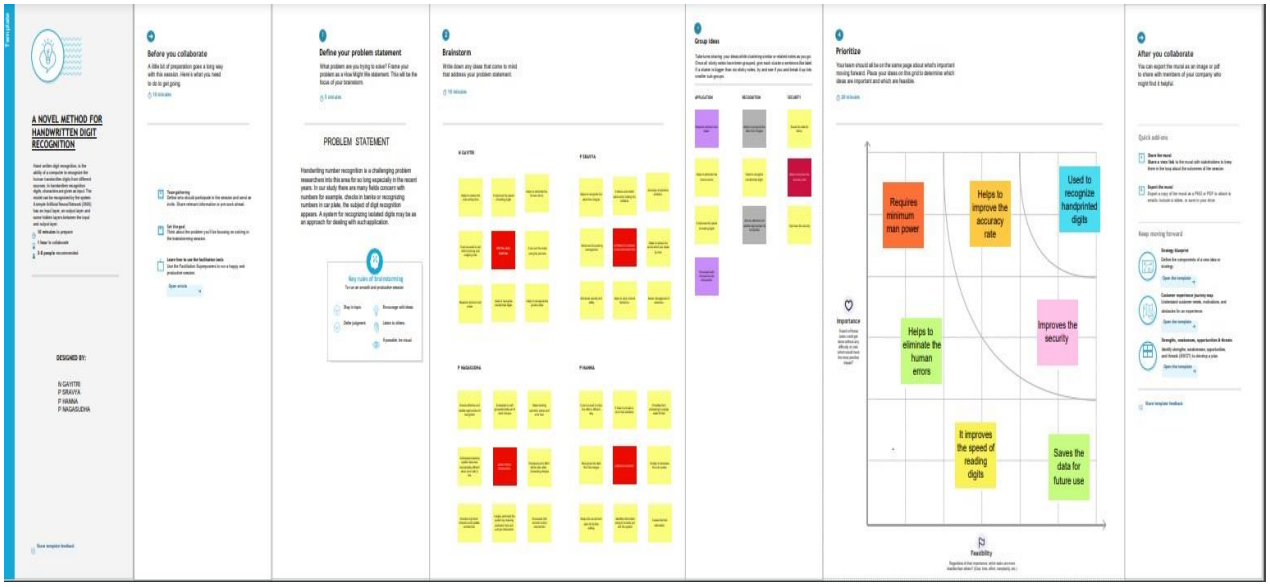
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Build empathy and keep your focus on the user by putting yourself in their shoes.


Curious about milestone
logic should work
design and flow
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
run a query
accuracy
team complaints
calculations
olorful display
igits and letters
What do they
SEE?
environment
friends
what the market offers
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
avoid excessive meetings
suggestions
error correction
PAIN
fears
frustrations
obstacles
no human improvement
high cost
uncertainty
GAIN
"wants" / needs
measures of success
obstacles
easy
accurate
user-friendly

3.2 Ideation & Brainstorming



3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A Novel Method for Handwritten Digit Recognition System
2.	Idea / Solution description	The proposed solution is to classify the digits which is in handwritten format by using CNN based model and this model can be trained by using MNIST database which contains 60,000 training samples and 10,000 test samples.
3.	Novelty / Uniqueness	To classify the image datasets by using CNN, which provides efficient solution compare to other methods. Here ANN algorithm is used for voice recognition which helps blind people.
4.	Social Impact / Customer Satisfaction	Users no need to use external dependencies or devices to recognize the digits, this process can be done through our mobile phones.
5.	Business Model (Revenue Model)	• Input module • Image processing module Segmentation module • Feature extraction module • Data set training module • Classification module
6.	Scalability of the Solution	The accuracy of the result for the training data set is **99.98%**, and 99.40% with 50% noise by using MNIST. Even we can improve this model to achieve the better results by training different types of datasets.

3.4 Problem Solution fit

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	The product essentially converts handwritten digits to digital form.	The user is first asked to draw a number on the canvas, and the model that is built is then utilised to compare the data and provide an output in digitalized form.
FR-2	Recognizing the handwritten digit and displaying.	Recognizing the handwritten digit and displaying.
FR-3	Import dataset file directly to the program from a command that will download the dataset from its website. Save the dataset file in the same directory as the program	Installing packages and applications.
FR-4	Build a Neural Network with a number of nodes in the input layer equal to the number of pixels in the arrays	Nil
FR-5	Activating the Neural Network	Packages – tensor flow

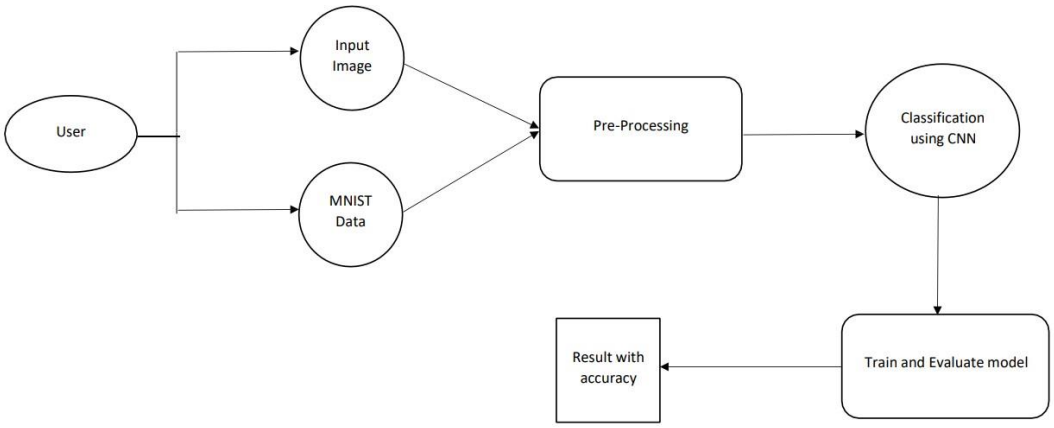
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	System design should be easily understood and user friendly to users. Furthermore, users of all skill levels of users should be able to navigate it without problems.
NFR-2	**Security**	The system should automatically be able to authenticate all users with their unique username and password
NFR-3	**Performance**	Should reduce the delay in information when hundreds of requests are given.
NFR-4	**Availability**	Information is restricted to each users limited access
NFR-5	**Scalability**	the system should be able to handle 10000 users accessing the site at the same time

5. PROJECT DESIGN

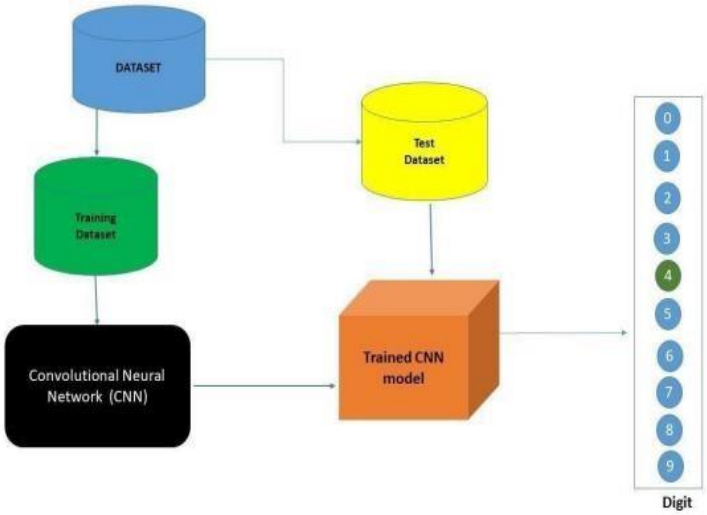
5.1 Data Flow Diagrams

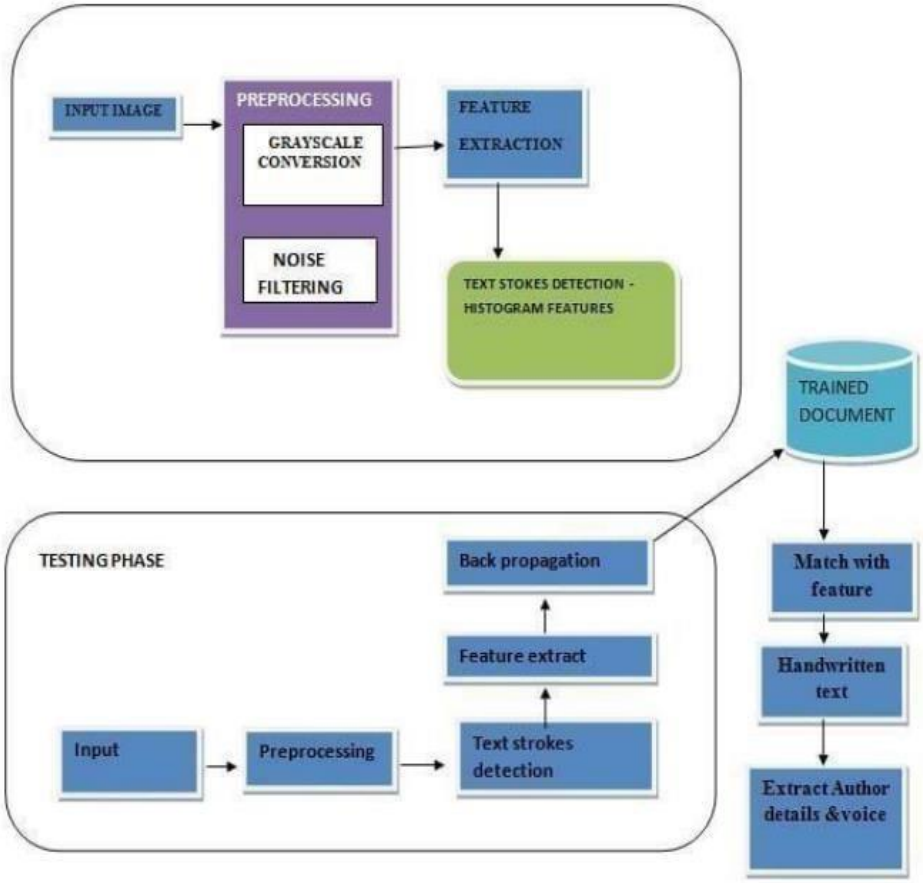
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

The architectural diagram of the model is as below ,





BLOCK DIAGRAM

5.3 User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Application	USN-1	As a user, I can application by opening it easily.	I can download the application	High	Sprint-1
		USN-2	As a user, I will be given access to the canvas board to draw or write the number	I can access the canvas	High	Sprint-1
		USN-3	As a user, I can change the colour of the pen ink.	I can use the canvas pen	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Home	USN-1	As a user, I can view the guide and awareness to use this application.	1	Medium	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D
Sprint-1		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	3	High	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D
Sprint-1		USN-3	As a user, I can read the instructions to use this application.	2	Low	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D
Sprint-2	Recognize	USN-4	As a user, In this recognition page I get to choose the image.	4	High	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Predict	5	As a user, I'm Allowed to upload and choose the image to be uploaded	3	Low	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D
Sprint-3		6	As a user, I will train and test the input to get the maximum accuracy of output.	4	High	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D

Sprint-3		7	As a user, I can access the MNIST data set	2	Medium	Karthick B, Nithiyanandam K, Praveen Kumar S, Amarnath D

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	31Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-3	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022
Sprint-4	20	6 Days	21Nov 2022	26 Nov 2022	20	26 Nov 2022

6.3 Reports from JIRA

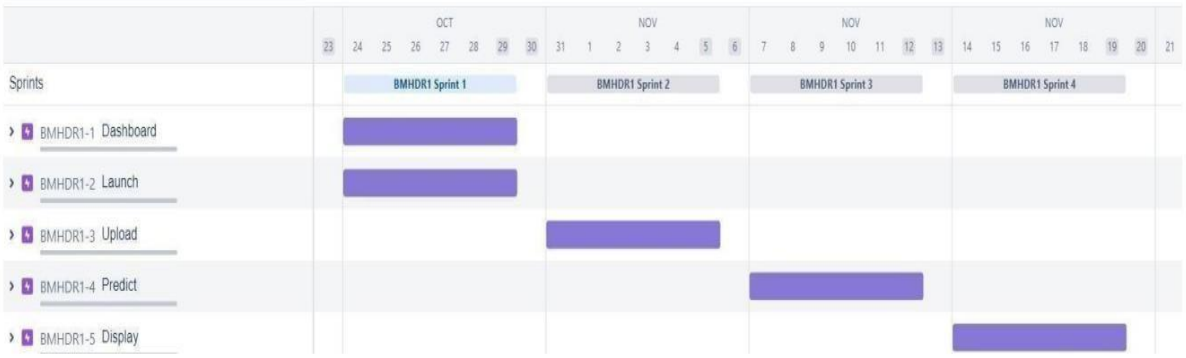
Velocity:

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)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

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



7. CODING & SOLUTIONING

7.1 PREDICTION PHASE

```
<!DOCTYPE html>

<html >


<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title> HAND WRITTEN PREDUCTION</title>

  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">

    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>

    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>

    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
```

```
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<link href="{{url_for('static',filename='css/final.css')}}" rel="stylesheet">
<style>
.header {
                        top:0;
                        margin:0px;
                left: 0px;
        right: 0px;
                        position: fixed;
        background-color: #3d24a3;
                        color: white;
                        box-shadow: 0px 8px 4px grey;
                        overflow: hidden;
        padding-left:20px;
                        font-family: 'Josefin Sans';
                        font-size: 2vw;
                        width: 100%;
        height:8%;
                        text-align: center;
                }
                .topnav {
  overflow: hidden;   background-
color: #333;
}

.topnav-right a {
  float: left;
```

```
 color: #f2f2f2;  text-
align: center;  padding:
14px 16px;  text-
decoration: none;  font-
size: 18px;
}

.topnav-right  a:hover  {      background-
color: #ddd;  color: black;
}

.topnav-right a.active {  background-
color: #565961;
 color: white;
}

.topnav-right {
 float: right;
 padding-right:100px;
}

.login{
margin-top:-70px;
}
body {

 background-color:#ffffff;  background-
repeat: no-repeat;  background-size:cover;
background-position: 0px 0px;
```

```
 }
.login{
        margin-top:100px;
}

.container {   margin-
top:40px; padding: 16px;
}
select {
        width: 100%;            margin-
bottom: 10px;          background:
rgba(255,255,255,255);
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #000000;
        text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
        border: 1px solid rgba(0,0,0,0.3);
        border-radius: 4px;
        box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
        -webkit-transition: box-shadow .5s ease;
        -moz-transition: box-shadow .5s ease;
        -o-transition: box-shadow .5s ease;   -ms-
transition: box-shadow .5s ease;       transition: box-
shadow .5s ease;
}
```

```
</style>
</head>

<body style="font-family:Montserrat;overflow:scroll; background-image: 2.gif;">

<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
paddingtop:1%"><i>Handwritten Prediction</i></div>
  <div class="topnav-right" style="padding-top:0.5%;">



 </div>
</div>
<div class="container">
      <div id="content" style="margin-top:2em">
                  <div class="container">
                   <div class="row">
                         <div class="col-sm-6 bd" >


                          <br>
                               <img src="{{url_for('static',filename='images/13.gif')}}"
style="height:450px;width:550px"class="img-rounded" alt="Gesture">
                         </div>
                         <div class="col-sm-6">
                               <div>
                                    <h4>Drop in the image to get the prediction  </h4>
                         <form action = "" id="upload-file" method="post"
enctype="multipart/form-data">
```

```
<label for="imageUpload" class="upload-label" style="background: #d5d4da;">
                choose
                        </label>
                        <input type="file" name="image" id="imageUpload"
accept=".png, .jpg, .jpeg">
<div class="showcase"><input type="text" name="showcase"></div>
                </form>


                <div class="image-section" style="display:none;">
                        <div class="img-preview">
                                <div id="imagePreview">
                                </div>
                        </div>
                        <div>
                                <button type="button" class="btn btn-info btn-lg "
id="btn-predict" style="background: #08f447;">Predict!</button>
                        </div>
                </div>


                <div class="loader" style="display:none;"></div>


                <h3>
                        <span id="result" style="font-size:17px; "> </span>
                </h3>
        </div>
                </div>
        </div>
        </div>
```

```
            </div>
    </div>
</body>
<footer>
    <script src="{{url_for('static',filename='js/main.js')}}" type="text/javascript"></script>
</footer>
</html>
```

8. TESTING

8.1 Test Cases

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

S.NO	Scenario	Input	Expected Output	Actual Output
1	Inserting 9 image as an input		Nine	Nine [9]
2	Inserting 8 image as an input		Eight	Eight [8]
3	Inserting 7 image as an input		Seven	Seven [7]
4	Inserting 6 image as an input		Six	Six [6]
5	Inserting 5 image as an input		Five	Five [5]
6	Inserting 4 image as an input		Four	Four [4]
7	Inserting 3 image as an input		Three	Three [3]
8	Inserting 2 image as an input		Two	Two [2]
9	Inserting 1 image as an input		One	One [1]

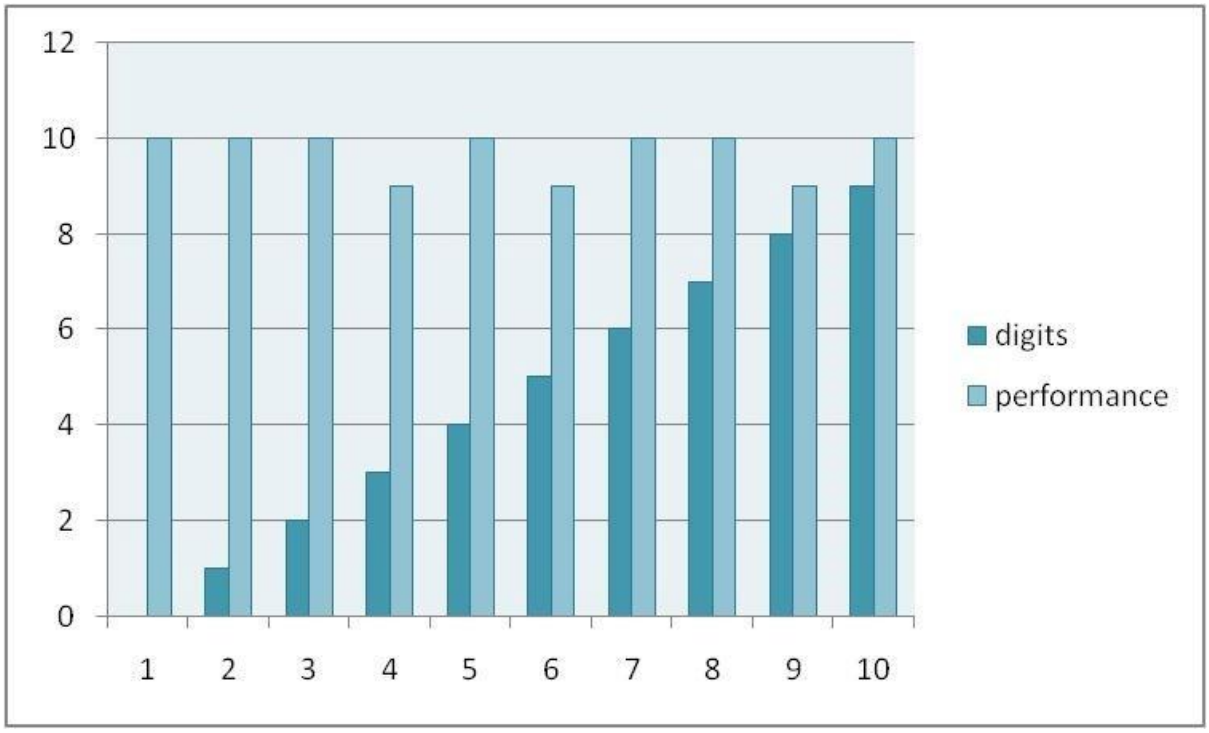
10	Inserting 0 image as an input		Zero	Zero [0]

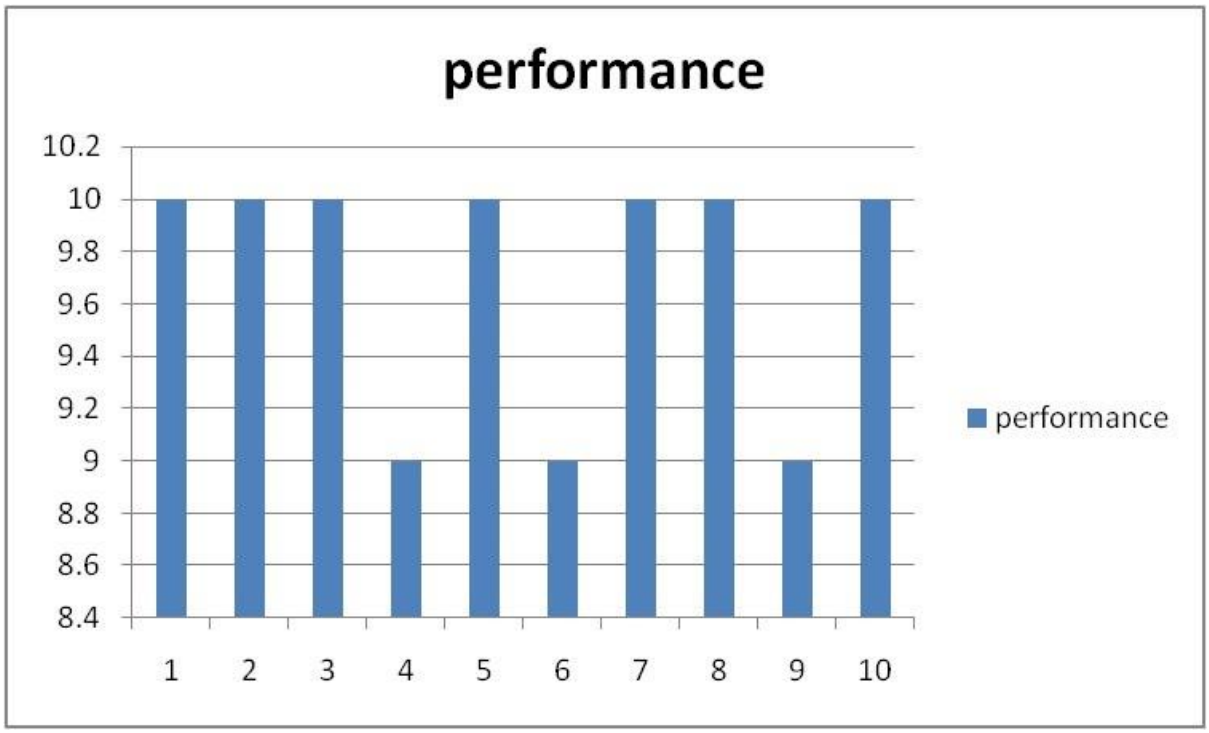
8.2 User Acceptance Testing

User Acceptance Testing is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done. This sort of testing is carried out by clients, or other authorized bodies to identify the requirements and operational procedures of an applications or piece of software. The most crucial stage of testing is acceptance testing since it determine whether or not the customer will accept the application or programmer.

9. RESULTS

9.1 Performance Metrics





10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. This system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style.
2. The generative models can perform recognition driven segmentation.
3. the method involves a relatively small number of parameters and hence training is relatively easy and fast.
4. This method helps to read and understand the digits properly by system.
5. Unexpected digit data can be predicted
6. It is able to adapt any kind of scripting styles
7. We can import any type of data set from any open source platform

DISADVANTAGES:

1. Quality is not always high
2. Time consuming and expensive
3. Sometimes inaccurate
4. Losing documents formatting
5. Error prone
6. Lack of information on some characters

7. Inability to recognize some languages
8. May not recognize right to left languages
9. Inaccuracy with damaged texts

11. CONCLUSION

Handwritten digit recognition has immense applications in the field of medical, banking, student management, and taxation process etc. Many classifiers like KNN, SVM, CNN are used to identify the digit from the handwritten image. as per the review, CNN is providing better performance than others. Stages of HDR using CNN classifier is discussed in this paper. MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers. HDR consists of three different stages. First is pre processing where dataset is converted into binary form and image processing has been applied on it. Second stage is segmentation where the image is converted into multiple segments. Third stage is feature extraction where features of image are identified. Last stage is classification where classifiers like KNN, SVM, CNN are used. Results of HDR is improved a lot by using CNN classifier but it can be improved further in terms of complexity, duration of execution and accuracy of results by making combination of classifiers or using some additional algorithm with it.

The variations of accuracies for handwritten digit were observed for 15 epochs by varying the hidden layers. The accuracy curves were generated for the six cases for the different parameter using CNN MNIST digit dataset.

The six cases perform differently because of the various combinations of hidden layers. The layers were taken randomly in a periodic sequence so that each case behaves differently during the experiment. The maximum and minimum accuracies were observed for different hidden layers variation with a batch size of 100. Among all the observation, the maximum accuracy in the performance was found 99.21% for 15 epochs in case 2 (Conv1, pool1, Conv2, pool2 with 2 dropouts). In digit recognition, this type of higher accuracy will cooperate to speed up the performance of the machine more adequately. However, the minimum accuracy among all observation in the performance was found 97.07% in case 6

(Conv1, pool1, Conv2, pool2 with 1 dropout). Moreover, among all the cases, the total highest test loss is approximately 0.049449 found in case 3 without dropout and the total lowest test loss is approximately 0.026303 found in case 2 with dropout. This low loss will provide CNN better performance to attain better image resolution and noise processing. In the future, we plan to observe the variation in the overall classification accuracy by varying the number of hidden layers and batch size.

12. FUTURE SCOPE

a) Electronic form filling

One of the applications of online handwriting recognition is electronic form filling. Internationally, the expenditure for entry of data from handwritten forms, notes and records is trillions of dollars. If we look at 2010 census of our country, more than fifty thousand enumerators were employed to collect data using handwritten forms, where they took six months to do this job. Further, it took two more years to feed this data into servers. So, in such applications, an immediate and direct conversion of handwritten data to typed data will result in reducing the huge cost and it will also increase the productivity. In this way, all government application forms can be completed and filled using handwriting recognition and the data will be directly entered to structured databases. It will be only possible, when the handwriting recognition is standardized, perfected and it is available to all computing devices. Further the writer dependent and independent systems could be developed as per the requirement.

b) Writing electronic applications in one's own handwriting and native script

There are a number of native language/script speakers/writers who want to exchange information with the computer system. These writers know their native script only, but don’t know typing in their script. One of the problems of these writers is to write electronic applications. In such situations, it becomes difficult for native script writers to communicate with computing devices. One of the options for such writers is speech recognition. But speech recognition has certain limitations/problems. In such scenarios, writing electronic applications in one’s own handwriting and native script is the best solution. It can be supported using online handwriting recognition. c) Automated music symbol notation reader

One of the applications of handwriting recognition is the development of automated music symbol notation reader. In this way, a composer can write his composition using all the notations directly. Then it is converted to the standard format to display/print for a book or for his symphony group. Further, the Vedic Sanskrit is identical, with the symbols for udata, anudata, svaritha, deerghasvaritha and plutha, and the engine which is capable of recognizing hand written Grantha or Devanagari with all such symbols will be in great use Putting in the mathematical equations by simple handwriting. The handwriting recognition is assumed to achieve the goal of automatic conversion of online handwritten mathematical equations to typed form or student notes. So it will not be wrong to imagine the happiness that would be experienced by a chemist, mathematician or an academician, if they can place the mathematical equations by simple online handwriting on the digital surface and make his/her book, plenary talk and journal papers.

13.APPENDIX

Source Code

```
from flask import Flask,render_template,request,redirect,url_for
from PIL import Image import numpy as np
from tensorflow.keras.models import load_model
import requests
from tensorflow.python.keras.backend import set_session import
os
from tensorflow.keras.preprocessing import image
import pandas as pd import tensorflow as tf
from werkzeug.utils import secure_filename app =
Flask(__name__, instance_relative_config=True) model
= load_model("mnistCNN.h5")
@app.route('/') def
home():
    return render_template('sai.html')
@app.route('/predict.html') def
prediction():
    return render_template('predict.html')
@app.route('/predict',methods=['POST','GET'])
def predict():     if request.method
=='POST':         f = request.files['image']
basepath= os.path.dirname(__file__)

file_path=os.path.join(basepath,'data',secure_filename(f.filename))

        img = Image.open(file_path).convert("L")
img=img.resize((28,28))
im2arr=np.array(img)
im2arr=im2arr.reshape(1,28,28,1)
y_pred=np.argmax(model.predict(im2arr))
print(y_pred)
```

```
        df=pd.read_excel('digitpred.xlsx')
loader=print(df.iloc[y_pred]['digits'])
else:
        return None
    return df.iloc[y_pred]['digits']
  if
__name__=='__main__':
  app.run(host='0.0.0.0',port=5000,debug=False)
```

HOME PAGE.html

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" type="text/css"
href="{{url_for('static',filename='css/intro.css')}}">

    <link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link
href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&family=Poor+Story&display=swap" rel="stylesheet">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link
href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&family=Pacifico&family=Poor+Story&display=swap" rel="stylesheet">

    <title>HOME</title>

</head>

<body>

<div class="topnav">

  <span>Handwritten Digit Recognition System</span>

  <a href="predict.html">Predict</a>
```

```
 <a href="#Intro">Home</a>
</div>

<div class="title">
    <p class="intro">INTRODUCTION</p>
</div>

<div class="container">
<span class="content"> The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image. Then, we will be predicting the labels based on the CNN trained model weights of handwritten digit recognition.</span>
</div>
</body>
</html>
```

HOME PAGE.CSS

```
body{ margin: 0px; padding:
0px; background-size: cover;
background-image: url("2.gif");
}

.topnav { background-color:
transparent; overflow: hidden;
height: 9vh; text-align: center;
justify-content: center;
padding: 20px; font-family:
'Pacifico', cursive; color: black;
}
```

```
/* Style the links inside the navigation bar */
.topnav a {
  float: right;
color: black;   text-
align: center;
text-align: center;
justify-content:
center;   padding:
17px 20px;   text-
decoration: none;
font-size: 22px;
}

/* Add a color to the active/current link */
.topnav a.active {   background-
color:skyblue;
  color: black;
}
 span{

  float: left;
  font-size: 40px;
color: black;
 }

.title{
```

```
 text-align: center;   color: black;
font-family: 'Dancing Script', cursive;
font-size: 40px;   background-color:
transparent;
}

.intor{

 margin: 25px;
}
 .container{   background-color:
rgba(0, 0, 0, 0.4);   height: 170px;
text-align: center;   padding: 30px;
}

 .content{

 font-size: 25px;   font-family:
'Pacifico', cursive;   justify-
content: center;   text-align:
center;   color: black;
 }
```

PREDICT.html

```
<!DOCTYPE html>
<html >

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
 <title> HAND WRITTEN PREDUCTION</title>

 <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">

   <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>

   <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>

   <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<link href="{{url_for('static',filename='css/final.css')}}" rel="stylesheet">

<style>

.header {

                        top:0;

                        margin:0px;

                left: 0px;

     right: 0px;

  position: fixed;    background-color: #3d24a3;

                        color: white;

                        box-shadow: 0px 8px 4px grey;

                        overflow: hidden;

     padding-left:20px;

                        font-family: 'Josefin Sans';
```

```
                font-size: 2vw;
                width: 100%;
        height:8%;          text-
align: center;
        }
        .topnav {
 overflow: hidden;  background-
color: #333;
}

.topnav-right a {
 float: left;
 color: #f2f2f2;  text-
align: center;  padding:
14px 16px;  text-
decoration: none;  font-
size: 18px;
}

.topnav-right a:hover {  background-
color: #ddd;
 color: black;
}

.topnav-right a.active {  background-
color: #565961;
 color: white;
}
```

```
.topnav-right {
  float: right;
  padding-right:100px;
}

.login{
margin-top:-70px;
}
body {

  background-color:#ffffff;   background-
repeat: no-repeat;   background-size:cover;
background-position: 0px 0px;
  }
.login{
      margin-top:100px;
}

.container { margin-
top:40px;   padding: 16px;
}
select {
      width: 100%;              margin-
bottom: 10px;       background:
rgba(255,255,255,255);    border: none;
      outline: none;            padding:
10px;         font-size: 13px;          color:
#000000;
```

```
    text-shadow: 1px 1px 1px rgba(0,0,0,0.3); border:
    1px solid rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;   -ms-
transition: box-shadow .5s ease;            transition:
box-shadow .5s ease;
}



</style>
</head>

<body style="font-family:Montserrat;overflow:scroll; background-image: 3.gif;">
<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
paddingtop:1%"><i>Handwritten Prediction</i></div>
  <div class="topnav-right" style="padding-top:0.5%;">




  </div>
</div>
<div class="container">
      <div id="content" style="margin-top:2em">
              <div class="container">
                <div class="row">
```

```
                    <div class="col-sm-6 bd" >


                     <br>

                            <img src="{{url_for('static',filename='images/3.gif')}}"
style="height:450px;width:550px"class="img-rounded" alt="Gesture">

                    </div>

                    <div class="col-sm-6">

                            <div>

                                    <h4>Drop in the image to get the prediction  </h4>

                    <form action = "" id="upload-file" method="post"
enctype="multipart/form-data">


  <label for="imageUpload" class="upload-label" style="background: #d5d4da;">

                Choose

</label>

  <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">

                            <div class="showcase"><input type="text"
name="showcase"></div>

                    </form>



                    <div class="image-section" style="display:none;">

                            <div class="img-preview">

                                    <div id="imagePreview">

                                    </div>

                            </div>

                            <div>

                                    <button type="button" class="btn btn-info btn-lg "
id="btn-predict" style="background: #08f447;">Predict!</button>

                            </div>
```

```
                    </div>

                    <div class="loader" style="display:none;"></div>

                    <h3>
                        <span id="result" style="font-size:17px; "> </span>
                    </h3>

                </div>
                    </div>
                </div>
                </div>
                </div>
    </div>
</body>

<footer>
    <script src="{{url_for('static',filename='js/main.js')}}" type="text/javascript"></script>
</footer>
</html>
```

Final.css

```
.img-preview {     width: 256px;     height: 256px;
position: relative;     border: 5px solid #F8F8F8;
box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
margin-top: 1em;
    margin-bottom: 1em;
}
```

```
.img-preview>div {     width:
100%;     height: 100%;
background-size: 256px 256px;
background-repeat: no-repeat;
background-position: center;
}

input[type="file"] {
display: none;
}

.upload-label{
   display: inline-block;
padding: 12px 30px;
background: #28272c;
   color: #fff;
   font-size: 1em;
transition: all .4s;     cursor:
pointer;
}

.upload-label:hover{
background: #C2C5A8;     color:
#39D2B4;
}

.loader {     border: 8px solid #f3f3f3; /*
Light grey */     border-top: 8px solid
#28272c; /* Blue */     border-radius: 50%;
```

```
width: 50px;    height: 50px;    animation:
spin 1s linear infinite;
}

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

Js.Javascript

```
$(document).ready(function () {
  // Init
  $('.image-section').hide();
  $('.loader').hide();
  $('#result').hide();

  // Upload Preview
function readURL(input) {
    if (input.files && input.files[0]) {
var reader = new FileReader();
reader.onload = function (e) {
          $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
          $('#imagePreview').hide();
          $('#imagePreview').fadeIn(650);
        }
        reader.readAsDataURL(input.files[0]);
    }
  }
  $("#imageUpload").change(function () {
    $('.image-section').show();
```

```
        $('#btn-predict').show();
        $('#result').text('');
$('#result').hide();        readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {        var form_data
= new FormData($('#upload-file')[0]);
  // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
type: 'POST',
url: '/predict',
            data: form_data,
contentType: false,
cache: false,
processData: false,
async: true,            success:
function (data) {                //
Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
$('#result').text('Prediction: '+data);
console.log('Success!');
            },
        });
```

```
    });

});
```

GitHub & Project Demo Link:

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-15157-1659594516

YouTube Link:

https://youtu.be/IIXx1ynLNQY

Project Demo Link:

https://drive.google.com/file/d/1NQeYVSN4gqXZOvJCykx0boMt7kbTWJhv/view?usp=drivesdk