

IBM PROJECT

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

TEAM ID: PNT2022TMID21532

**S. B. Dhanya Prabha
N. Ramswetha
P. Harshavarthini
S. Nandhini**

CONTENTS

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2. LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

5. PROJECT DESIGN

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

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

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

8. TESTING

- 8.1 Test Cases
- 8.2 User Acceptance Testing

9. RESULTS

- 9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten digit recognition is one of the important problems. The handwritten digit recognition system has to identify the handwritten digits which are of different styles. This is the major challenge of the handwritten digits recognition system. The applications of digit recognition include 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 handwritten digits and which is submitted by users by way of a scanner, tablet, and other digital devices.

1.2 PURPOSE

The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digits. The system proposed should be very efficient and easy to use. The handwritten digits recognition system should be able to recognize different handwriting styles with greater accuracy.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

The handwritten digits recognition is a very essential system that has application in various fields. But the existing handwritten digits recognition system is not so accurate and the predicted results are not so reliable. The handwritten digits are of different styles. Each individual would have a unique handwriting and it is difficult to recognize those different styles accurately. The existing systems have several drawbacks and those should be overcome in the proposed system. With the help of the deep learning techniques we can develop a reliable system that predicts the given input correctly.

2.2 REFERENCES

i. JOURNALS:

S.NO	PUBLISHED IN	YEAR OF PUBLISHING	TITLE	AUTHORS	ABSTRACT
1.	IEEE	2019	A NOVEL METHOD FOR HAND WRITTEN DIGIT RECOGNITION USING DEEP LEARNING	Rohini.M1,Dr. D.Surendran2 1,Assistant Professor,Sri Krishna College of Engineering and Technology, 2,Professor, Sri Krishna College of Engineering and Technology	Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, the results of some of the most widely used Machine Learning Algorithms like CNN-convolution neural networks and Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensorflow are used. MNIST is a dataset which is widely used for

					<p>handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images. The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field. For example Convolution Neural networks with back propagation for image processing. The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.</p>
2.	IJCSIT	2011	A novel method for Handwritten Digit Recognition with Neural Networks	<p>MALOTHU NAGU,1, N .VIJAY SHANKAR, 2,K.ANNAPUR NA,3 1,Department of ECE, V.K.R &V.N.B.Engg College,Gudivada. Krishna (Dist), A. P, S INDIA. 2,Department of EIE ,S R T I S T, Ramananda Nagar, Nalgonda (Dist).A.P, S.INDIA.</p>	<p>It plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. This is a system widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed. Bayesian Decision theory,</p>

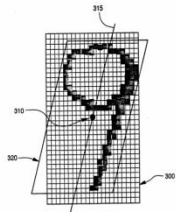
				3,School of Electronics, Vignan University, Guntur (Dist).A.P, S.INDIA.	Nearest Neighbor rule, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition uses Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%.This accuracy rate is very high.
--	--	--	--	---	---

ii. **CONFERENCE:**

S.NO	TITLE	AUTHORS	ABSTRACT	CONFERENCE
1	Handwritten Digit Recognition Using Machine Learning: A Review	Anchit Shrivastav Isha Jaggi Sheifali Gupta Deepali Gupta Chitkara University Institute of Engineerin g and	The task for handwritten digit recognition has been troublesome due to various variations in writing styles. Therefore, the authors have tried to create a base for future researches in the area so that the researchers can overcome the existing problems. The existing methods and techniques for handwritten digit recognition were reviewed and understood to analyze the most suitable and best method for digit recognition. A number of 60,000 images were used as training sets of images with pixel size of 28×28. The images/training sets were matched with original image. It was found out after complete analysis and	2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)

		Technology, Chitkara University, Punjab,	review that classifier ensemble system has the least error rate of just 0.32%. In this paper, review of different methods handwritten digit recognition were observed and analyzed	
2	A Comparative Study on Handwriting Digit Recognition Using Neural Networks	Mahmoud M.Abu Gosh Ashraf Y. Maghari Faculty of Information Technology, Islamic University of Gaza, Palestine	The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). In this paper, the three NN approaches are compared and evaluated in terms of many factors such as accuracy and performance. Recognition accuracy rate and performance, however, is not the only criterion in the evaluation process, but there are interesting criteria such as execution time. Random and standard dataset of handwritten digit have been used for conducting the experiments. The results show that among the three NN approaches, DNN is the most accurate algorithm; it has 98.08% accuracy rate. However, the execution time of DNN is comparable with the other two algorithms. On the other hand, each algorithm has an error rate of 1-2% because of the similarity in digit shapes, specially, with the digits (1,7), (3,5), (3,8), (8,5)	2017 International Conference on Promising Electronic Technologies (ICPET)

iii. PATENTS:

S.N O	PATENT NUMBER/ FILE	INVENTOR	APPLICATIONS	DIAGRAM
1.	US532544 7A https://patentimages.storage.googleapis.com/b0/ab/b3/e4f251e6deb81/US5325447.pdf	Robert C. Vogt, III	A handwritten character image normalization technique provides predetermined pixel dimensions and a normalized skew. The skew slope of the input image is calculated. This skew slope is employed to determine the bounds of the smallest parallelogram which completely encloses all of the stroke pixels of the input image. This parallelogram has a first pair of opposed horizontal sides and a second pair of opposed sides having the skew slope. The stroke pixels of this parallelogram are then mapped into the standard size horizontal row and vertical column pixel dimensions using horizontal and vertical scaling factors determined from the parallelogram dimensions and the standard dimensions. This mapping employs a subpixel grid of the normalized pixels. Candidate stroke pixels are identified which correspond to any part of a stroke pixel of the input image. A candidate stroke pixel is set to a stroke pixel if and only if the number of such subpixels mapped into a stroke pixel of the input image exceeds a predetermined number.	 <p>FIG - 3</p>

2.	EP0555227A4 https://patents.google.com/patent/EP0555227A4/en	David L Mccubbrey	<p>There are many instances where it would be useful or desirable to provide a computer readable form of a document not available in a compatible computer readable form. Normally it is the case that the document is not available in machine readable form because the document was handwritten or typewritten and thus no computer readable form exists, or because the computer readable form is not available. In some instances there is a "foreign11 document, i.e. an existing computer readable form but the document was produced on an incompatible computer system. In some instances, such as facsimile transmission, a simple optical scan of the document can produce the required form. In most instances the form most useful for later use and decision making is a separate indication of each character of the document.</p> <p>The field of optical character recognition deals with the problem of separating and indicating printed or written characters. In optical character recognition, the document is scanned in some fashion to produce a electrical image of the marks of the document. This image of the marks is analyzed by computer to produce an indication of each character of the document. It is within the current state of the art to produce relatively error free indication of many typewritten and printed documents. The best systems of the prior art are capable of properly distinguishing a number of differing type fonts.</p>	
----	--	-------------------	--	--

2.3 PROBLEM STATEMENT DEFINITION

Handwritten digit recognition is very important as it will be very helpful to reduce human effort. As each individual has different handwritings for representing digits, the system should have a capability to identify every handwriting with maximum accuracy. Such a system will be useful to reduce human interventions in identification, as everything is being digitized. The main

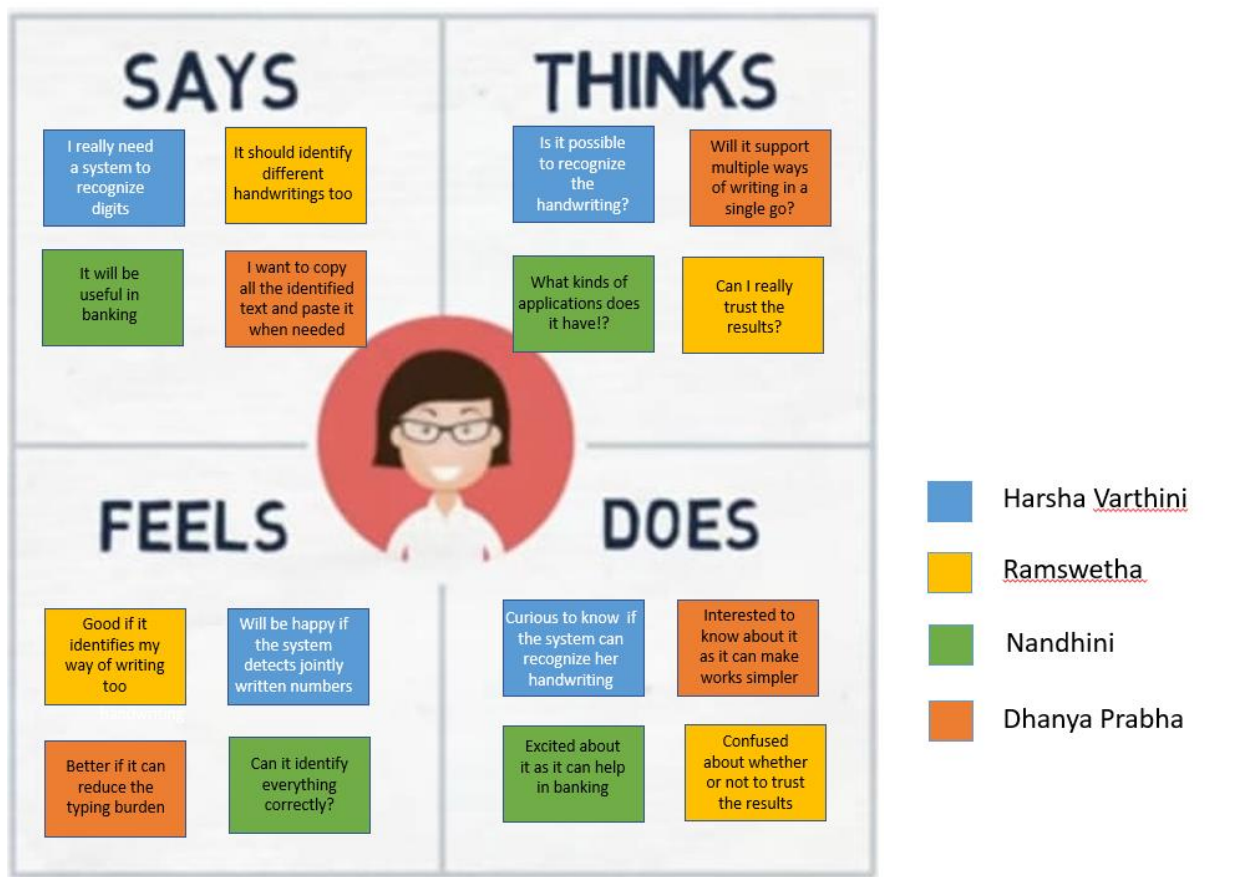
objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free. Handwriting recognition has gained a lot of attention in the field of pattern recognition and machine learning due to its application in various fields. Various techniques have been proposed to for digit recognition in handwriting recognition system.

The user has to just handwrite using the fingers on the smart screen of mobile of PC and the software of the device must be able to identify the handwriting. There should also be the feature of producing text out of the recognized contents and the user must be able to copy the contents and paste it to the desired locations.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 IDEATION & BRAINSTORMING

Brainstorming:

The following are the ideas generated by our team members

Dhanya Prabha S B: Let's choose a topic related to AI domain and let's choose a project for handwritten

digit recognition system. Let's do it for digits from 0 to 9.

Nandhini S: We can make it for the English alphabets, so that it will be easy for the users and it will

eliminate the need of manpower.

Ramswetha N: Let's go for the digit recognition as it can be improved with newer technologies whereas there are lots of existing methodologies to identify alphabets.

Harshavarthini P: And in addition to that, we have to develop the digit recognition system with much

reduced latency.

Idea Listing and Grouping:

- o Handwritten digit recognition system
- o Recognition system for English alphabets
- o Improvising the digit recognition with newer technologies
- o The digits should be recognised with zero latency

Finally, we decided to implement the handwritten digit recognition system using the Artificial Intelligence. The system must recognize with zero error and zero latency. The existing methodologies for the digit recognition should be improved to produce better results.

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A novel method for handwritten digit recognition system. Handwritten digit recognition is very important as it will be very helpful to reduce human effort. As each individual has different handwritings for representing digits, the system should have a capability to identify every handwriting with maximum accuracy. Such a system will be useful to reduce human interventions in identification, as everything is being digitized. The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free. Handwriting recognition has gained a lot of attention in the field of pattern recognition and machine learning due to its application in various fields. Various techniques have been proposed to for digit recognition in handwriting recognition system.

2.	Idea / Solution description	The idea is to recognize the handwritten digits in any smart device. The solution provided will be a software that can recognize the digits written on the screen and convert the recognized contents in the form of text or speech. The user is free to copy or store the recognized contents in the desired locations.
3.	Novelty / Uniqueness	To recognize the handwritten digits using newer technologies. The system must recognize with zero latency and zero error.

4.	Social Impact / Customer Satisfaction	Due to this solution, the common android users are satisfied as the digits written by them can be easily recognized. The customers such as the normal mobile mobile users, research scholars and the archaeologists are so much benefitted by this project.
----	---------------------------------------	---

5.	Business Model (Revenue Model)	A revenue model is a framework for generating financial income. It identifies which revenue source to pursue, what value to offer, how to price the value, and who pays for the value. The system for handwritten digit recognition system is designed based on the revenue model
6.	Scalability of the Solution	The solution can be scaled for many different languages. Other than digit recognition, the system can be used to recognize the alphabets of multiple languages.

3.4 PROBLEM SOLUTION FIT

Project Title: A Novel Method for Handwritten Digit Recognition System
Team ID: PNT2022TMID21352

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) Who is your customer? <ul style="list-style-type: none"> Normal android users and laptop users Research scholars Archaeologists 	6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking action or limit their choices of solutions? <ul style="list-style-type: none"> Must require a mobile device Should support the recognition system software The system RAM should be fast enough to support the recognition with zero latency 	5. AVAILABLE SOLUTIONS 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? A system to recognize the letters written on the smart screen using fingers or the supported styllet A keyboard with digit recognition feature so as to translate the handwritten form	Explore AS, different BE
	2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. Many users are not fluent in typing and the typing is error some. It takes a lot of time to type and the process is tedious. It is difficult for the research scholars to type the entire research paper from handwritten notes The archaeologists could not recognize the exact text from the engravings at times.	9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job? Many people find it difficult to type in keyboards For languages such as Tamil, due to the availability of lots of letters, it's difficult to type using a single keyboard As researchers and archaeologists find it hard, it will be very helpful for them to identify the handwritten digits.	7. BEHAVIOUR What does your customer do to address the problem and get the job done? They will undertake the tedious process of typing and will get frustrated in no time. For different languages like Tamil and Chinese, the users have to use the limited keyboard which is a very tedious process. They have to undertake the complex process of typing the special characters if they have to do so.	

3. TRIGGERS What triggers customers to act? The tedious process of typing triggers them to switch over to digit recognition so that they can just write with their fingers on the Smart Screen or on the keyboards.	10. YOUR SOLUTION A novel method for handwritten digit recognition system. Handwritten digit recognition is very important as it will be very helpful to reduce human effort. As each individual has different handwritings for representing digits, the system should have a capability to identify every handwriting with maximum accuracy.	8. CHANNELS of BEHAVIOUR 8.1 ONLINE It will be useful for browsing online and can be used in many online applications 8.2 OFFLINE
--	--	---

4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job and afterwards? Before: Frustrated, irritated, boring, mundane After: Relaxed, saves time, easy, happy and satisfied	Such a system will be useful to reduce human interventions in identification, as everything is being digitized. The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free. Handwriting recognition has gained a lot of attention in the field of pattern recognition and machine learning due to its application in various fields. Various techniques have been proposed to for digit recognition in handwriting recognition system.	The same application can be built for offline purposes in many softwares and can be implemented in the keyboard that is to be used in the smart phones
--	--	--

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	Smart Keyboard	Enable button to allow handwriting through fingers or stylet Select button to switch between languages
FR-2	Smart scanner	Select button to scan the handwriting Copy button to allow copy the recognized text
FR-3	Smart Document editor	Enable button to allow handwriting, converting it to text in the document
FR-4	Audio player for handwritten characters/words/digits	Enable Audio output reading the handwritten characters/words/digits
FR-5	A touch sensitive surface, which may be integrated with, or adjacent to, an output display.	Precise recognition even of the unclear handwritten digits/characters

4.2 NON-FUNCTIONAL REQUIREMENT

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	A decent scanner is required so that the composed specimens can be examined and given as info the framework. On the off chance that scanner is not accessible then just composed characters or digits are perused by framework with mouse or advanced pen.
NFR-2	Security	In registering, security is the strategies for guaranteeing that information put away in the app can't be perused or bargained by any people without approval. Most PC efforts to establish safety include information encryption and passwords. Information encryption is the interpretation of information into a structure that is indiscernible without a disentangling system. A watchword is a mystery word or expression that gives a client access to a specific project or framework.
NFR-3	Reliability	Unwavering quality is a property of any App related part (programming, or equipment, or a system, for

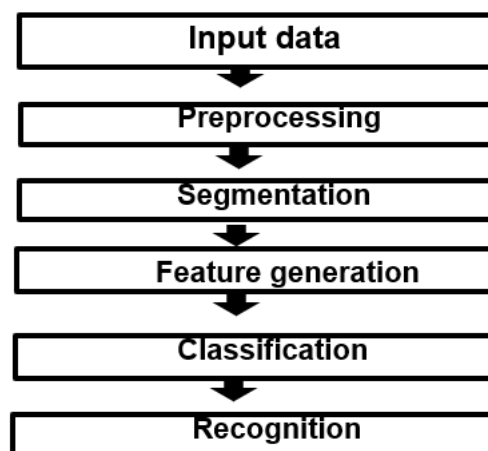
		instance) that reliably performs as indicated by its determinations. It has for sometime been viewed as one of three related qualities that should be considered when making, purchasing, or utilizing an app item or part
NFR-4	Performance	Execution relying upon the connection, high App execution might include one or a greater amount of the accompanying: Short reaction time for a given bit of work. Execution is described by the measure of valuable work achieved by a PC framework or App contrasted with the time and assets utilized
NFR-5	Availability	In App frameworks and systems administration, accessibility is a general term that is utilized to depict the measure of time over a one-year period that the framework assets is accessible in the wake of part disappointments in the system. A framework constantly accessible with its all assets are viewed as fruitful.
NFR-6	Scalability	Most of the characters in several languages such as Hebrew, Sanskrit, braille ,etc...can be recognized with reduced latency inclusive of many features such as touch sensitive keyboard and audio output.

5. PROJECT DESIGN

The design phase is a stage where software developers define the technical details of the product. In the design phase, one or more designs are developed, with which the project result can apparently be achieved. Depending on the subject of the project, the products of the design phase can include dioramas, sketches, flow charts, site trees, HTML screen designs, prototypes, photo impressions and UML schemas.

5.1 DATA FLOW DIAGRAMS

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.



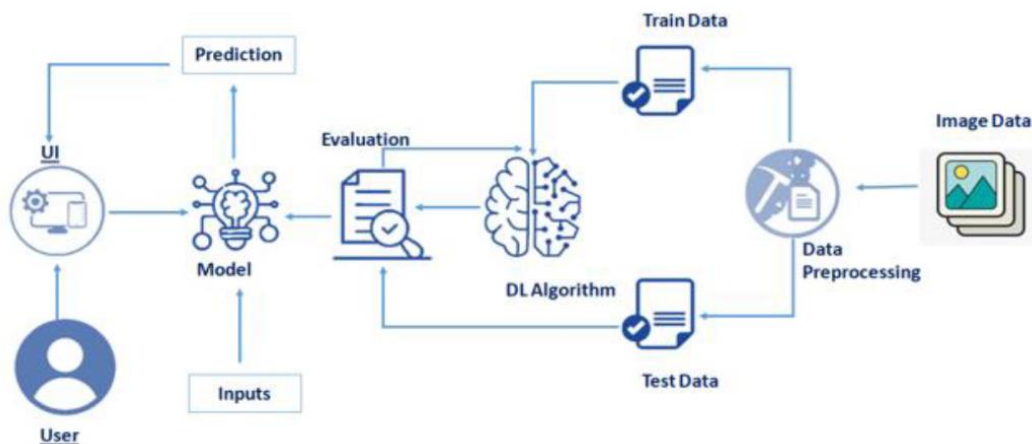
5.1.1 Data flow diagram of the system

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

The solution architecture helps ensure that a new system will fit the existing enterprise environment. To perform this task, a solution architect has to understand how all parts of the business model work together including processes, operating systems, and application

architectures. A solution architecture (SA) is an architectural description of a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

Technical architecture



5.2.1 technical architecture flow chart

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The communication protocol being used in the proposed solution might act as an interface the way like WiFi, Bluetooth and ZigBee	MIT APP
2.	Application Logic	Numerous apps have been developed to help you convert your scanned documents into digital text. All those apps use a technology called Optical Character Recognition, or OCR. Upload a photo of your handwritten notes or use a scanner and transfer it to your computer, tablet, iPhone, or android device.	Python, IBM Watson STT service
3.	Database	Data to be segregated and secured in the form of relational DBMS.	MySQL
4.	Cloud Database	Database Service on Cloud	IBM Cloudant
5.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
6.	External API-1	The MNIST database is a large database of handwritten digits	training and testing in the field of machine learning.
7.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration	Cloud Foundry

Table-2: Application Characteristics:

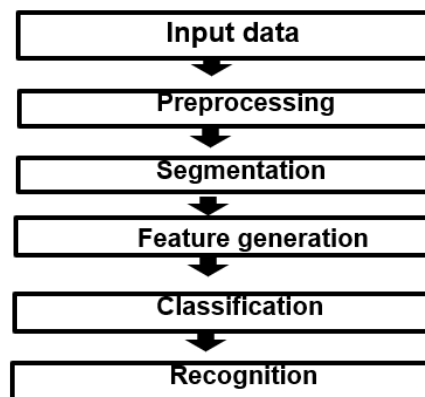
S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	an open-source Python library developed by the Google Brain labs for deep learning research, you will take hand-drawn images of the numbers 0-9 and build and train a neural network to recognize and predict the correct label for the digit displayed	TensorFlow technology
2.	Security Implementations	The developed application should be accessible in the	Encryptions, IAM Control
S.No	Characteristics	Description	Technology
		way it can only respond to the comments of the relevant users	
3.	Scalable Architecture	The app format comes the way easier to handle and operate.	Not yet determined
4.	Availability	The developed solution tends to be available in the market at any time	Not yet determined
5.	Performance	Highly proper and betterment functionalities are to be ensured in the designed solution	Not yet determined

SOLUTION ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

SOLUTION ARCHITECTURE DIAGRAM:



5.2.2 solution architecture flow chart

5.3 USER STORIES

A user story is a tool in Agile software development used to capture a description of a software feature from a user's perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Handwritten)	Recognition	USN-1	As a user, I can use the system for efficient identification of the handwritten digits. The predicted output is accurate and the numerical	I can get good prediction results.	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
digit recognition system user)			data identified can be processed and used with convenience.			
	Image processing	USN-2	As a user, I can use this system to identify the handwritten digits in the images by importing it into the interface.	I can identify handwritten digits from images.	High	Sprint-1
	Scanner	USN-3	As a user, I can use this handwritten digit recognition system to scan the handwritten digits using the camera and get outstanding predicted results.	I can use the scanner.	High	Sprint-1
	Training	USN-4	As a user, I can identify different styles of handwriting from different sources of data. The predicted numerical data is accurate enough and reliable.	I can recognize different styles through the system.	High	Sprint-1
	Scalability	USN-5	As a user, I can use the system for various languages and recognize the digits in various languages with higher accuracy and speed.	I can use the system for different languages.	Low	Sprint-4
	Storage	USN-6	As a user, I can use the system to identify the numerical data and process it to get information and store it for future reference.	I can use the system for storing the identified data	Medium	Sprint-2
	Performance	USN-7	As a user, I can achieve higher performance with this system in handwritten digits recognition.	I can get good performance.	High	Sprint-1
	Speed	USN-8	As a user, I can achieve good speed in identifying the numerical data from various sources.	I can achieve high speed.	Medium	Sprint-3

6. PROJECT PLANNING AND SCHEDULING

The process of planning primarily deals with selecting the appropriate policies and procedures in order to achieve the objectives of the project. Scheduling converts the project action plans for scope, time cost and quality into an operating timetable.

6.1 Sprint Planning & Estimation

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

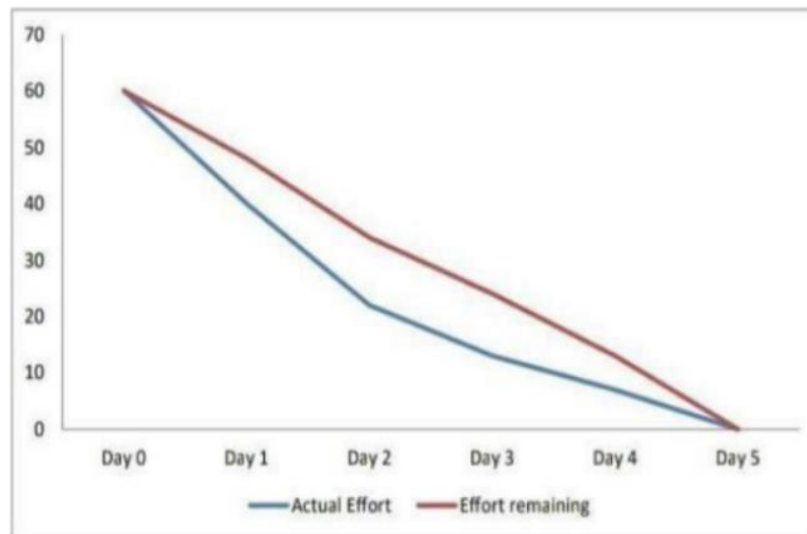
6.2 Sprint Delivery Schedule

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)

$$\begin{aligned}AV &= \text{Sprint duration} / \text{Velocity} \\ &= 20 / 6 \\ &= 3.33\end{aligned}$$

Burndown Chart:

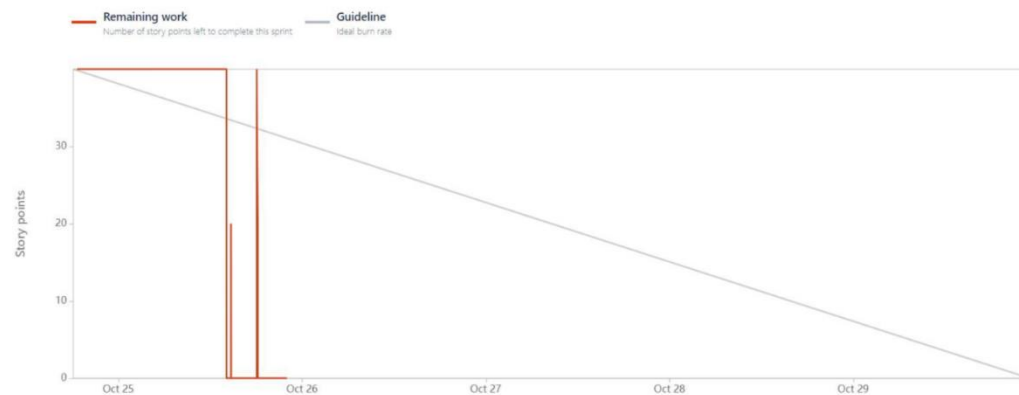
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.2.1 Burndown chart

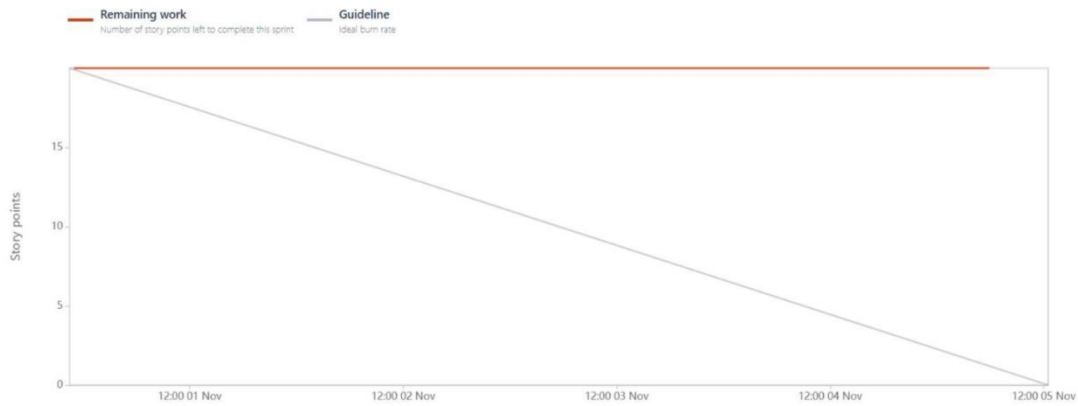
SPRINT 1:

Date - October 24th, 2022 - October 29th, 2022



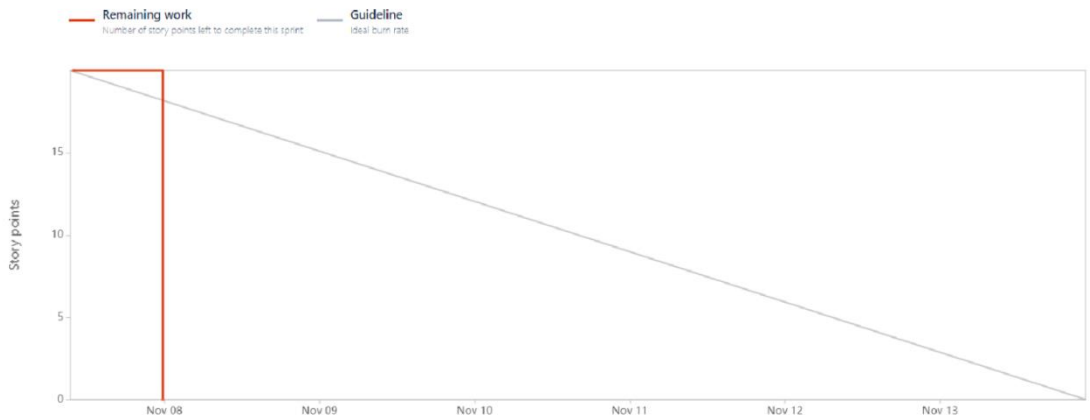
SPRINT 2 :

Date - October 31st, 2022 - November 5th, 2022



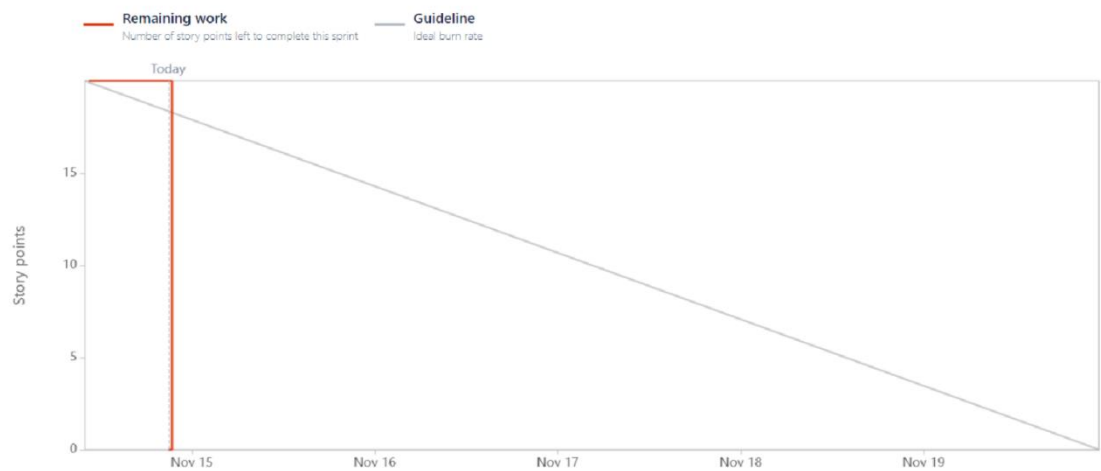
SPRINT 3:

Date - November 7th, 2022 - November 13th, 2022



SPRINT 4:

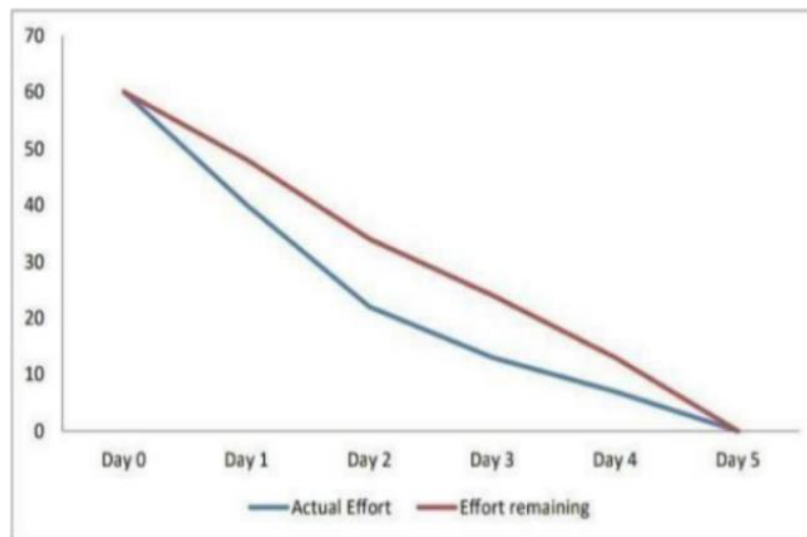
Date - November 14th, 2022 - November 19th, 2022



6.2 REPORTS FROM JIRA

The reports from JIRA deliver value to customers faster with real-time insights at your fingertips. Jira Software enables teams to make data-driven decisions with agile reports, dashboards, and more.

BURNDOWN CHART:



7. CODING & SOLUTIONING

It's called problem solving, or critical thinking. The idea behind a term like solutioning, though, is that it seeks to structure the problem solving process so that it is consistent, repeatable, and generates optimal results.

SPRINT 1:

FEATURES: Understanding the data

ML depends heavily on data, without data, it is impossible for a machine to learn. It's the most crucial aspect that makes algorithm training dataset. Machine learning project, we need a training data set. It's the actual dataset used to train the model for performing various actions. Tensorflow already has MNIST dataset. So, there is no need to explicitly download or create dataset.

MNIST dataset contains 10 classes; from 0 to 9. Each digit is taken as a class.

The following steps are followed:

- Importing the required libraries
- Loading the data
- Analysing the data
- Reshaping the data
- Applying one hot encoding

UNDERSTANDING THE DATA

```
import numpy #for numerical analysis
```

```
import tensorflow#open source ml tool by google
```

```
from tensorflow.keras.datasets import mnist
```

```
# mnist dataset
```

```
from tensorflow.keras.models import Sequential # stack for layers
```

```
from tensorflow.keras import layers#input,middle and output layers forcnn structure
```

```
from tensorflow.keras.layers import Dense,Flatten #dense and flatten layers
```

```
from tensorflow.keras.layers import Conv2D #convolutional layers
```

```
from tensorflow import keras #library for building neural networks built on tensorflow
from tensorflow.keras.optimizers import Adam #optimizers
from keras.utils import np_utils
```

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

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

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

```
(10000,)
```

```
array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0, 124, 253, 255, 63,  0,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0, 96, 244, 251, 253, 62,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0, 127, 251, 251, 253, 62,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0, 68, 236, 251, 211, 31,  8,  0,  0,  0,  0,
         0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0,  0, 60, 228, 251, 251, 94,  0,  0,  0,  0,  0,
         0,  0],
```

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 155, 253, 253, 189, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 20, 253, 251, 235, 66, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
32, 205, 253, 251, 126, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
104, 251, 253, 184, 15, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 80,
240, 251, 193, 23, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 253,
253, 253, 159, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 151, 251,
251, 251, 39, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 48, 221, 251,
251, 172, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 234, 251, 251,
196, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 253, 251, 251,
89, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 159, 255, 253, 253,
31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 48, 228, 253, 247, 140,
8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 64, 251, 253, 220, 0,

```

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 64, 251, 253, 220, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 193, 253, 220, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]], dtype=uint8)

```

LOADING THE DATA

```
(x_train,y_train),(x_test,y_test)=mnist.load_data()
```

ANALYZING THE DATA

```

print(x_train.shape)print(x_test.shape)
print(x_test.shape)print(y_test.shape)
x_train[3]y_train[0]import matplotlib.pyplot as pltplt.imshow(x_train[36])

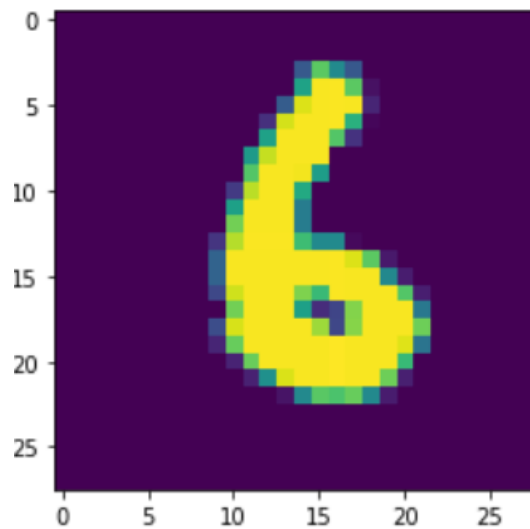
```

RESHAPING THE DATA

```

#(batch,height,width,channel)
x_train=x_train.reshape(60000,28,28,1).astype('float32')
x_test=x_test.reshape(10000,28,28,1).astype('float32')

```



APPLYING ONE-HOT ENCODING

#(batch,height,width,channel)

x_train=x_train.reshape(60000,28,28,1).astype('float32')

x_test=x_test.reshape(10000,28,28,1).astype('float32')

no_of_classes=10

y_train=np_utils.to_categorical

(y_train,no_of_classes)y_test=np_utils.to_categorical(y_test,no_of_classes)

y_test[3]

array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32)

SPRINT2:

MODEL BUILDING

The following features are included:

- Add CNN Layers
- Compiling the model
- Train the model
- Observing the metrics
- Test the model
- Save the model
- Test with saved model

ADDING CNN LAYERS

```
model=Sequential ()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
model.add(Flatten())
model.add(Dense(number_of_classes,activation = 'softmax'))
```

COMPILING THE MODEL

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

TRAINING THE MODEL

```
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

OBSERVING THE METRICS

```
metrics = model.evaluate(x_test, y_test, verbose = 0)
print("Metrics(Test loss and test accuracy): ")
print(metrics)
```

TESTING THE MODEL

```
prediction=model.predict(x_test[6000:6001])
print(prediction)
import numpy as np
print(np.argmax(prediction, axis=1))
np.argmax(y_test[5000:5001])
```

OBSERVING THE METRICS

```
metrics = model.evaluate(x_test, y_test, verbose = 0)
print("Metrics(Test loss and test accuracy): ")
print(metrics)
```

TESTING THE MODEL

```
prediction=model.predict(x_test[6000:6001])
print(prediction)
import numpy as np
print(np.argmax(prediction, axis=1))
np.argmax(y_test[5000:5001])
```

SAVING THE MODEL

```
model.save('sprint2/project.h5')
```

TESTING WITH THE SAVED MODEL

Done by testing the file with another jupyter code

After training the model with 5 epochs, we get a model that could be saved in our directory.

SPRINT 3

HTML CODE [INDEX]

FEATURES:

We designed the frontend using html that creates a webpage and it accepts an uploaded image from the user. After prediction using the backend python file, we receive the predicted result.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Image Classification</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
  <h1 align="center">
    <font face="Lato" color="#017bf5" size="7">
      Upload an <br> image to <br> Convert!!! <br>
    </font>
  </h1>
  <br><br>
  <form class="form-horizontal" action="/submit" method="post" enctype="multipart/form-data">
```

```

<div class="form-group">
  <label class="control-label col-sm-2" for="pwd">Upload Your Image :</label>
  <div class="col-sm-10">
    <input type="file" class="form-control" placeholder="Hours Studied" name="my_image" id="pwd">
  </div>
</div>

<div class="form-group">
  <div class="col-sm-offset-2 col-sm-10">
    <button type="submit" class="btn btn-success">Recognize</button>
  </div>
</div>
</form>

  {% if prediction %}

  <h2> Your Prediction : <i> {{prediction}} </i></h2>

  {% endif %}

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

```

PYTHON CODE:

FEATURES:

We use flask which is a web application framework written in python. It runs in the backend process and it runs the cnn model to predict/ classify the images uploaded by the user.

```

from flask import Flask, render_template, request
from keras.models import load_model
from keras.preprocessing import image

app = Flask(__name__)

```



```

dic = {0 : '0', 1 : '1', 2: '2', 3 : '3', 4: '4', 5: '5', 6: '6', 7: '7', 8: '8', 9: '9'}

model = load_model('model.h5')

model.make_predict_function()

def predict_label(img_path):
    i = image.load_img(img_path, target_size=(100,100))
    i = image.img_to_array(i)/255.0
    i = i.reshape(1, 100,100,3)
    p = model.predict_classes(i)
    return dic[p[0]]

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

@app.route("/about")
def about_page():
    return "Please subscribe Artificial Intelligence Hub..!!!"

@app.route("/submit", methods = ['GET', 'POST'])
def get_output():
    if request.method == 'POST':
        img = request.files['my_image']

        img_path = "static/" + img.filename
        img.save(img_path)

        p = predict_label(img_path)

        return render_template("index.html", prediction = p, img_path = img_path)

if __name__ == '__main__':

```

```
#app.debug = True
```

```
app.run(debug = True)7.3 Database Schema (if Applicable)
```

8. TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

8.1 Test Cases

1. Testing the model:

The below code is used to test the created python deep learning model

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

[[5.50544734e-15  7.41999492e-20  5.00876077e-12  1.26642463e-09
  3.52252804e-21  1.54133163e-17  3.15550259e-21  1.00000000e+00
  1.32678888e-13  6.44072333e-14]
 [1.51885260e-08  8.02883537e-09  1.00000000e+00  6.44802788e-13
  6.37117113e-16  3.40490114e-15  2.15804121e-08  2.18907611e-19
  3.38496564e-10  2.07915498e-20]
 [3.14093924e-08  9.99941349e-01  2.01593957e-06  1.45100779e-10
  5.25237965e-06  1.59223120e-07  3.15299786e-08  1.53995302e-07
  5.09846941e-05  1.14552066e-07]
 [1.00000000e+00  1.35018288e-14  2.28308122e-10  1.79766094e-16
  1.28767550e-14  7.12401882e-12  2.92727509e-11  3.52439052e-13
  2.56207252e-12  2.32345068e-12]]
```

```
import numpy as np
print(np.argmax(prediction,axis=1)) #printing our labels from first 4 images
print(y_test[:4]) #printing the actual labels

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

8.2 User Acceptance Testing

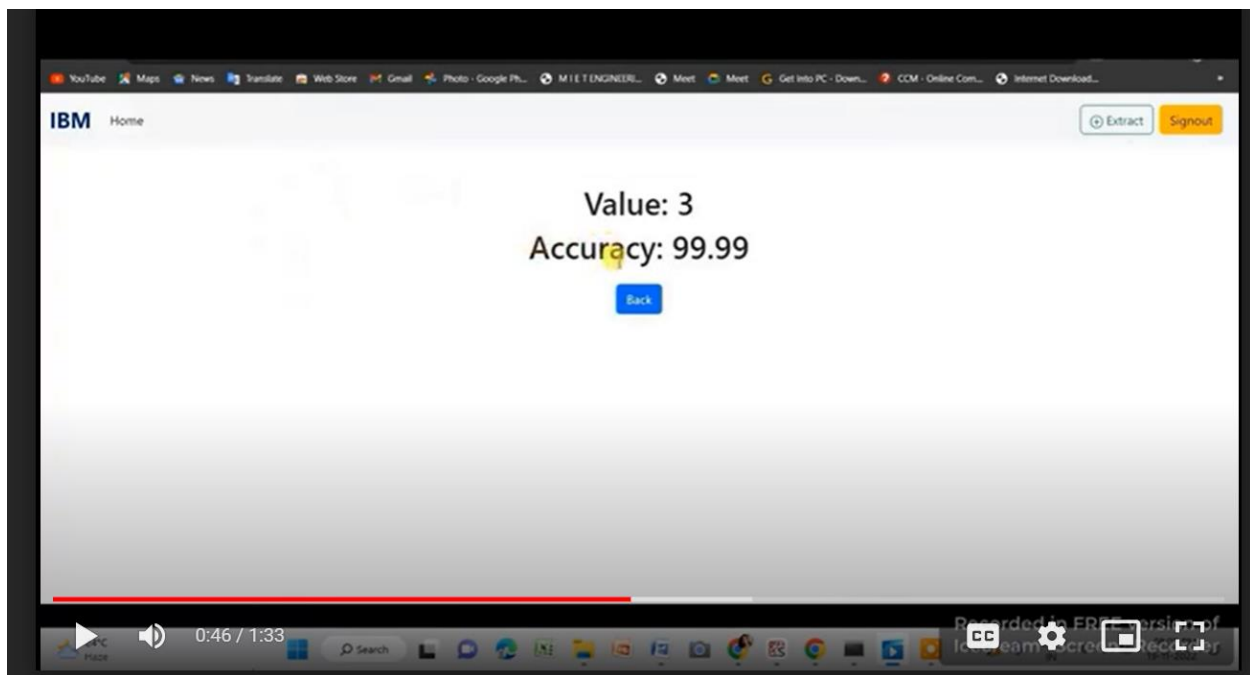
These are the test cases given by the user in the web application and get the predicted results as shown in the succeeding images.

Test case 1:

INPUT:



OUTPUT:

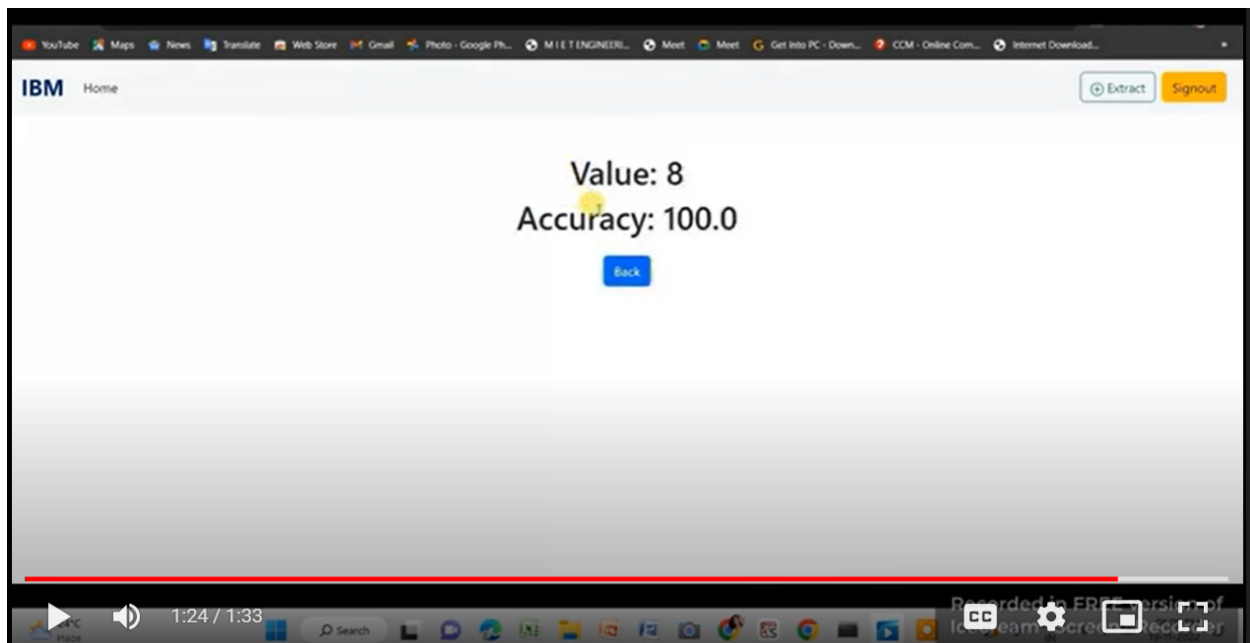


Test case 2:

INPUT:



OUTPUT:



9. RESULTS

Performance Metrics

As with any work or project taken up in the field of machine learning and image processing we are not considering our results to be perfect. Machine learning is a constantly evolving field and there is always room for improvement in your methodology; there is always going to be another new approach that gives better results for the same problem. The application has been tested using three models: Multi-Layer Perceptron (MLP), Convolution Neural Network (CNN). With each model we get a different accuracy of the classifier which shows which one is better

The results of training the network is stored in .npz format so that whenever a user tries to recognize the digit, the application does not go into the training loop again. For classification, we have used logistic classifier, softmax function, one hot encoding, cross entropy and loss minimization using mini batch gradient descent. These are some of the basics of Neural Network which are required to process the output from the network and display in the form the user can understand.

INPUT:



OUTPUT:

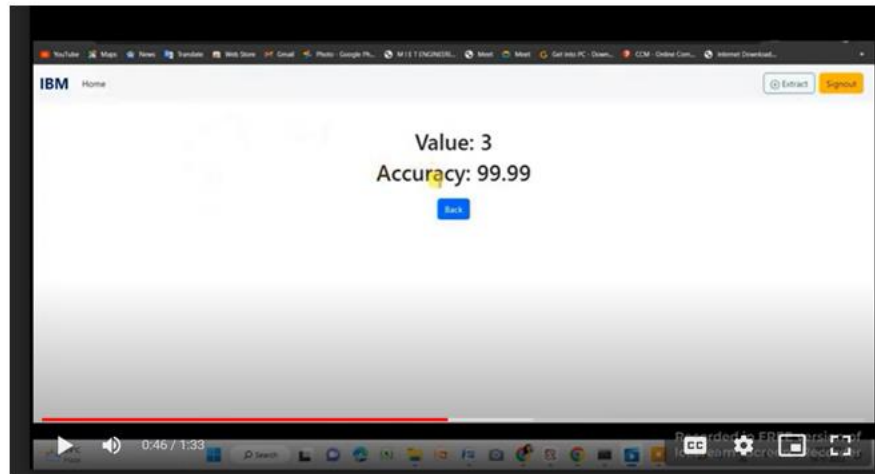


Fig.: Output of digit recognition of numerical '6'

A. DATASET USED The dataset used is the MNIST database of handwritten digits. It consists of a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image.

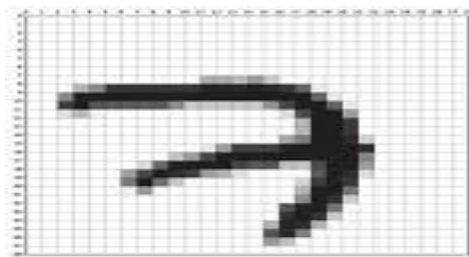


Fig MNIST sample belonging to the digit '7'



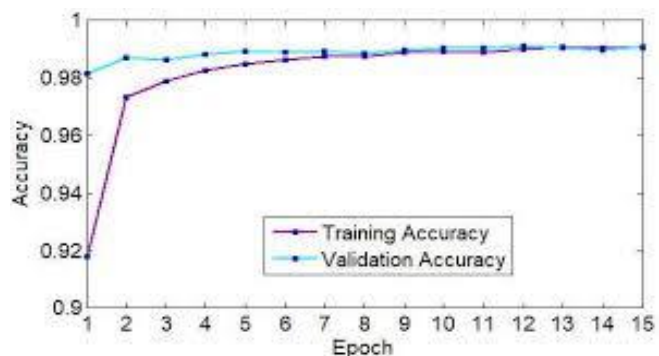
Fig 100 samples from the MNIST training set

The images are of size 28*28 pixels. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

B. ANALYSIS OF THE RESULTS:

In business, System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. System analysis and design relates to shaping organizations, improving performance and achieving objectives for profitability and growth. The emphasis is on systems in action, the relationships among subsystems and their contribution to meeting a common goal. Looking at a system and determining how adequately it functions, the changes to be made and the quality of the output are parts of system analysis. Organizations are complex systems that consist of interrelated and interlocking subsystems. Changes in one part of the system have both anticipated and unanticipated consequences in other parts of the system. The systems approval is a way of thinking about the analysis and design of computer based applications. It provides a framework for visualizing the organizational and environmental factors that operate on a system. Proposed Application Module: The proposed application has been implemented using Python on terminal. The user is given two options in the home image: Simple Upload, Model Form Upload. Simple Upload will allow the user to upload the image and predict it then and there. After navigating away from that page, the link to the uploaded image is lost. The Model Form Upload will allow the user to upload the image with description. With this link, the user will be able to store the image and see its link on the home page itself. By clicking on the link, the user will be able to get the result from the CNN classifier.

```
Epoch 1/5
- 14s - loss: 0.4236 - acc: 0.8208
Epoch 2/5
- 12s - loss: 0.2800 - acc: 0.8854
Epoch 3/5
- 11s - loss: 0.2147 - acc: 0.9115
Epoch 4/5
- 11s - loss: 0.1714 - acc: 0.9329
Epoch 5/5
- 11s - loss: 0.1344 - acc: 0.9464
```



OBSERVING THE METRICS

```
metrics = model.evaluate(x_test, y_test, verbose = 0)
print("Metrics(Test loss and test accuracy): ")
print(metrics)
```

```
Metrics(Test loss and test accuracy):
[0.09733106195926666, 0.9772999882698059]
```

Fig.Data training and Accuracy

10. ADVANTAGES

In our proposed model, the data augmentation technique serves as the essential build module. Our model takes the advantage of applying data augmentation to modify the original limited dataset (MNIST dataset) with a view towards possessing big data characteristics. In this sense, extracting extra information from the original dataset enables us to obtain training data with improved quality and size, which helps in preventing overfitting in our neural network. Inspired by the fact that a large number of epochs may result in overfitting, and a small number of epochs usually leads to underfitting, our model adopts the early stopping technique to determine the optimal number of training epochs. In order to correctly initialize the dataset to be fit for utilizing in our proposed CNN model, data preparation is conducted as an essential first step of our proposed model. By applying the data preparation process, we can determine whether there are any redundant variables or irrelevant variables to the target variable. To simulate the real-world natural influences that can affect image quality, which in turn influences the classification accuracy of the image, additive white Gaussian noise with 0.5 was added to the MNIST dataset. Then, the noisy MNIST dataset is used to evaluate the classification performance of our proposed algorithm.

11. CONCLUSION

An implementation of Handwritten Digit Recognition using Deep Learning has been implemented in this paper. Additionally, some of the most widely used Machine Learning algorithms i.e. CNN using Tensorflow have been trained and tested on the same data to draw a comparison as to why we require deep learning methods in critical applications like Handwritten Digit Recognition. In this paper, I have shown that that using Deep Learning techniques, a very high amount of accuracy can be achieved. Using the Convolutional Neural Network with Keras and Theano as backend, I am able to get an accuracy of 95.72%. Every tool has its own complexity and accuracy. Although, we see that the complexity of the code and the process is bit more as compared to normal Machine Learning algorithms but looking at the accuracy achieved, it can be said that it is worth it. Also, the current implementation is done only using the CPU. Thus we settled on classifying a given handwritten digit image as the required digit using three different algorithms and consequently testing its accuracy. In future we are planning to further explore the topic to recognize people's handwriting.

12. FUTURE SCOPE

We believe that our proposed model can further be applied to other datasets. In contrast, as a future work, we find that it is worth taking further actions to improve our model performance in terms of how to perfectly learn and extract the local features in the hidden layers, and how to enhance the recognition ability in the fully connected layers to avoid mislabeling problems.

13. APPENDIX

GitHub Link

<https://github.com/IBM-EPBL/IBM-Project-25504-1659966356>

Project Demo Link

https://drive.google.com/file/d/1n46zx0zQTPQl8S0xAYiF8yGC50wsaMNU/view?usp=share_link

Reference:

<https://www.atlassian.com/agile/project-management>

<https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software>

<https://www.atlassian.com/agile/tutorials/epics>

<https://www.atlassian.com/agile/tutorials/sprints>