AI - A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

KNOWLEDGE INSTITUTE OF TECHNOLOGY

PNT2022TMID30180

MUHILAN P (611219104061)

PANNEER SELVAM B (611219104066)

RANJITH M (611219104074)

SAKTHIDHARI B (611219104080)

DOMAIN

ARTIFICIAL INTELLIGENCE

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

TABLE OF CONTENTS

CHAPTER – 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten digit recognition is a computer vision task of recognizing the digits from a given hand written digit with the help of advanced neural networks and deep learning mechanism which is designed and developed for the users to make their work easier considering as a user friendly tool where anyone can use without any prior knowledge. The goal of this project is to recognize handwritten digits from an image. This project will be using the MNIST dataset, which is a set of 60,000 28x28 grayscale images of handwritten digits from 0-9.The input for this project will be an image and we will convert it to grayscale. We then use a convolutional neural network to classify the digit that was drawn in the image. The goal of this project is to extract handwritten digits from a scanned image and recognize them by using convolutional neural network.

1.2 PURPOSE

The goal of this project is to build a model that predicts hand written digit recognition for different users so that they can just upload their hand written digit and get their output within no time also considering the privacy we don't store any user data and any other information. We aim to provide a new user experience for our user by providing various support to get more accuracy for their hand written digit. The purpose of this project is to create a system that can recognize hand written digits, which will be used in various fields such as banking, education, and others and to identify handwritten digits from a given set of images. The user is able to input the desired number of digits, and the program will then show a set of images for each digit. The user must then select which digit in the image matches what they have written on their paper. Handwritten digit recognition has many purposes, such as handwriting recognition for computing, and for recovering the original text from an image of a scanned document.

CHAPTER – 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Handwritten digit recognition finds its application in various fields such as post mail sorting system where scanned images of mail envelopes are made into queue and extract the section describing postcode to be delivered. With the help of digit recognizer, sorting of mails can be done based on these postcodes according to their region. Another application that utilizes this technique is form processing, digits are extracted from certain columns of a form and users put certain filters to get the desired results they want. But there is no interface for a user to get their images scanned and recognized which makes the task complicated to use for a normal user.

2.2 SURVEY REPORT

2.2.1 A NOVEL APPROCH FOR HANDWRITTEN DIGIT RECOGNITION USING MULTILAYER PERCEPTION NEUTRAL NETWORK [Toufik Datsi, Khalid Aznag, Ahmed El Oirrak, 2022]

This paper is based on Artificial Neural Networks which are proved their effectiveness in the areas of image processing. It is about minimize the number of pixels by using as input the data extracted and calculated from the initial image. The approach consists of transforming the image of the digit in the binary format then encode each column by value. The architecture of Artificial Neural Network used in this research is based on a multilayer perceptron neural network in order to recognize and predict the handwritten digit from 0 to 9. For better training and testing dataset, we have used the backpropagation as a learning algorithm. A dataset of 6000 samples was obtained from the MNIST database. For better training and testing dataset, we have used the backpropagation as a learning algorithm.

2.2.2 A NOVEL HANDWRITTEN DIGIT CLASSIFICATION SYSTEM BASED ON CONVOLUTIONAL NEUTRAL NETWORK APPROACH [Ali Abdullah Yahya, Jieqing Tan, et al, 2021]

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: 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. 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. To decrease the errors of training and validation, and avoid the limitation of datasets, data augmentation has been proposed to simulate the real-world problems.

2.2.3 EFFECTIVE HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTION NEURAL NETWORK [Yellapragada SS Bharadwaj, Rajaram P, et al, 2020]

This paper proposed a simple neural network approach towards handwritten digit recognition using convolution. With machine learning algorithms like KNN, SVM/SOM, recognizing digits is considered as one of the unsolvable tasks due to its distinctiveness in the style of writing. In this paper, Convolution Neural Networks are implemented with an MNIST dataset of 70000 digits with 250 distinct forms of writings. The proposed method achieved 98.51% accuracy for real-world handwritten digit prediction with less than 0.1 % loss on training with 60000 digits while 10000 under validation. Advancements in the field of computer vision using deep neural networks attract attention; thus, many A.I. practitioners are moving towards it. One of the influencing projects that opted for deep learning is OCR. Handwritten digit recognition (HDR) is a snippet of OCR where instead of taking the whole digital data, HDR detects digits. Comparing to OCR, HDR is light and faster. In fields like medical, banking, student management, and taxation process, HDR possesses great flexibility.

2.2.4 IMPROVED HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTUIONAL NEURAL NETWORK [Savita Ahlawat, Amit Choudhary, et al, 2020]

Traditional systems of handwriting recognition have relied on handcrafted features and a large amount of prior knowledge. Training an OCR system based on these prerequisites is a challenging task. Research in the handwriting recognition field is focused around deep learning techniques and has achieved breakthrough performance in the last few years. Still, the rapid growth in the amount of handwritten data and the availability of massive processing power demands improvement in recognition accuracy and deserves further investigation. Convolutional neural networks (CNNs) are very effective in perceiving the structure of handwritten digits in ways that help in automatic extraction of distinct features and make CNN the most suitable approach for solving handwriting recognition problems. Our aim in the proposed work is to explore the various design options like number of layers, stride size, receptive field, kernel size, padding and dilution for CNN-based handwritten digit recognition. In addition, we aim to evaluate various SGD optimization algorithms in improving the performance of handwritten digit recognition.

2.2.5 HANDWRITTEN DIGIT RECOGNITION USING ENSEMBLE LEARNING [Kuppa Venkata Padmanabha Nandan, Manoj Panda, S. Veni, 2020]

This paper is mainly focused on Ensemble learning. In pattern recognition, the recognition of handwritten digits has always been a very challenging and tedious task. In this work, a simple novel approach is proposed to recognize the handwritten digits. The primary goal of this work is recognition of the handwritten digits by using ensemble learning. Ensemble learning improves convergence by decreasing the complexity of the model to facilitate accurate and improved decision. This is also helpful to know about distribution of data in the random split and class-wise split. It’s about analysis of how the load is distributed among the base learners and how it impacts the model accuracy and training time. The overall trends of the ensemble model have also been analysed in this paper.

2.2.6 HYBRID CNN – SVM CLASSIFIER FOR HANDWRITTEN DIGIT RECOGNITION [Savita Ahlawat, Amit Choudhary,2020]

The aim of this paper is to develop a hybrid model of a powerful Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) for recognition of handwritten digit from MNIST dataset. The proposed hybrid model combines the key properties of both the classifiers. In the proposed hybrid model, CNN works as an automatic feature extractor and SVM works as a binary classifier. The MNIST dataset of handwritten digits is used for training and testing the algorithm adopted in the proposed model. The MNIST dataset consists of handwritten digits images which are diverse and highly distorted. The receptive field of CNN helps in automatically extracting the most distinguishable features from these handwritten digits. The experimental results demonstrate the effectiveness of the proposed framework by achieving a recognition accuracy of 99.28% over MNIST handwritten digits dataset. In the area of handwriting recognition, several methods have been proposed in the literature such as Artificial

Neural Network (ANN), Neuro-Fuzzy Systems (NFS), Support Vector Machine (SVM) and deep learning-based classifiers [1-9]. Although decent recognition accuracy has been reported by these classifiers; handwriting digit recognition is still an open research problem and demands for exploring new techniques and methodologies that would further improve the performance in terms of recognition accuracy, running time and computational complexity.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Customer	Use software	It takes long time	Some issue in software	Disappointed
PS-2	Semi blind/Old people	find the number	Not sure whether result is correct or not	Not Clear eye sight	Uncertain
PS-3	Bank Employee	Conversion to machine readable format	Not getting accuracy	Similarity in digits	Irritated
PS-4	Postal service clerk	Recognize zip code	Much time consumption	Struggling to read people handwriting	Depressed

CHAPTER – 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

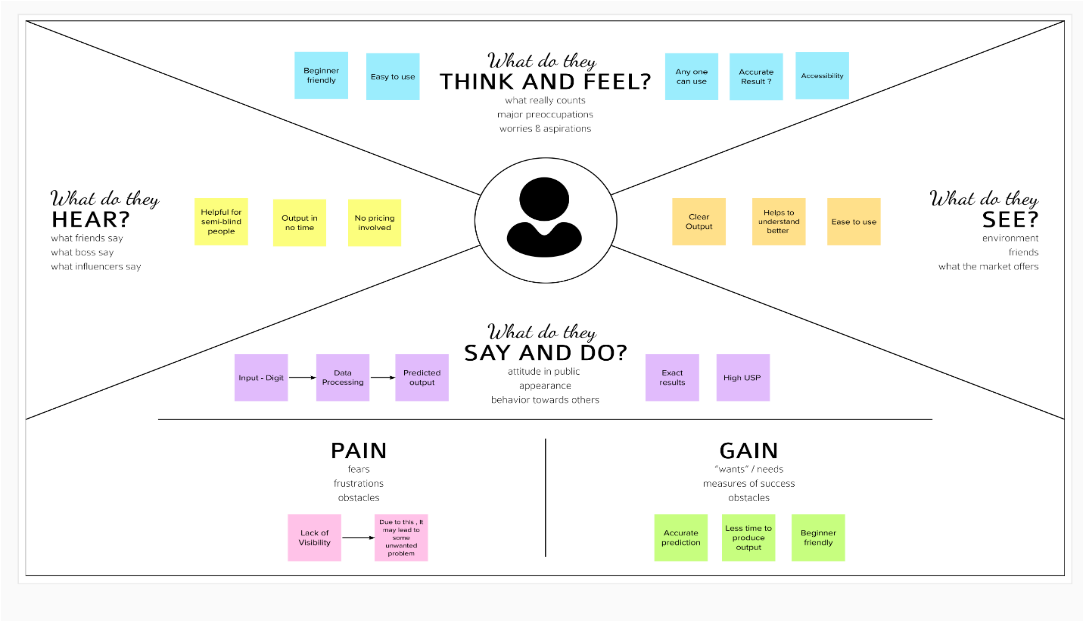


Fig 3.1.1 Empathy Map

3.2 IDEATION AND BRAINSTORMING

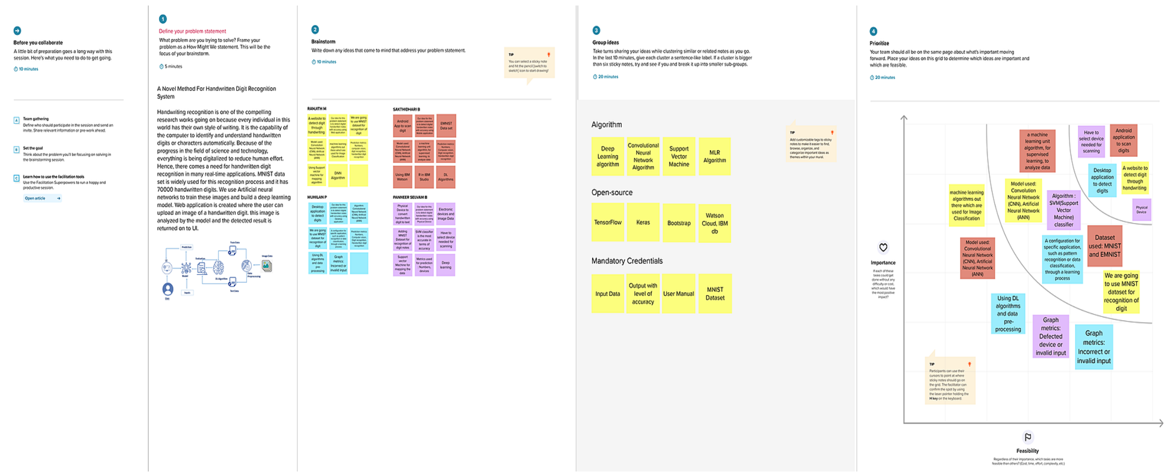


Fig 3.2.1 Brainstorming and Idea Prioritization

3.3 PROPOSED SOLUTION

Handwritten digit recognition is a technology that has been around for a while. It is the process of converting handwritten digits into machine-readable data. Handwritten digit recognition system software is used in many different applications such as document scanning, handwriting input, number identification and signature verification. Handwritten digit recognition is a difficult task for humans and for computer systems. The accuracy of human recognition is about 98%. This means that the error rate is about 2%. However, the error rate in machine-based recognition is much higher than that of humans.

The neural networks are a type of machine learning algorithm which has been proven to be able to solve this problem with much accuracy in the output. Developers have been working on improving the performance of these neural networks by optimizing algorithms and testing new architectures. The proposed solution is an AI-based system that can recognize handwritten digits using a convolutional neural network (CNN) and a graphical user interface (GUI). A neural network is a system of interconnected nodes that process data efficiently.

A Convolutional Neural Network is a type of neural network that has been used for handwriting recognition. In the past, the accuracy of handwritten digit recognition system was not high enough to be used in banking or other high-security applications. However, recent improvements in machine learning algorithms and hardware allow handwriting recognition systems to achieve higher accuracies than before.

The system is composed of a GUI, CNN, SVM classifier and the handwritten digit recognition module. The GUI is used for inputting data, CNN for processing and classifying the inputted data and finally the handwritten digit recognition module for recognizing the inputted data.

AI - A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Human brain can adopt to work or task performance after few practices and they can also analysis image with accuracy. So, prediction of handwritten digit is performed with different algorithm for converting handwritten digit to digital form and attain maximum accuracy in less time. Prediction can be done with help of many algorithms for better analysis. Some find it difficult to understand people handwriting so digitalization is done to reduce the difficulty. More hundreds of datasets are used for recognition of handwritten digit for better analysis.
2.	Idea / Solution description	Implementation of Handwritten digit recognition using various algorithms such as SVM, CNN, IBM Cloud and the IBM Cognos for analytic. Conversion of handwritten digit to computerized format using DL algorithm. Recognition of digit with less time and high accuracy.
3.	Novelty / Uniqueness	The product will have high quality outcome within in no time. Also predict digits with high accuracy.
4.	Social Impact / Customer Satisfaction	Usually, semi-blinded people face the problem of identifying the exact digit

		wherein if they are using this software, they will be able to use it wisely.
5.	Business Model (Revenue Model)	Banking sector for recognition of account number from cheque. Can create a profitable business with research and expertise. Postal zip codes recognition in postal sector. Data entry in form through recognition of digit.
6.	Scalability of the Solution	This software can help in all the banking sectors such as private banks, Government banks, and Urban banks in which they will be able to figure out the exact digit in order to prevent fraudulent transactions. It not only solves problems in banks but it also helps many individuals.

3.4 PROBLEM SOLUTION FIT

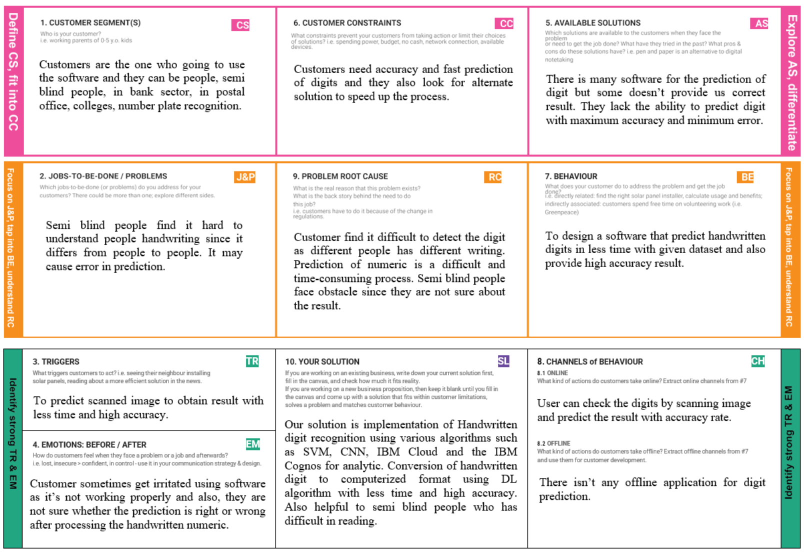


Fig 3.4.1 Problem Solution Fit

CHAPTER – 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	User Input	Handwritten digit
FR-2	Model	Model building, Adding CNN layers, Training the model on IBM.
FR-3	Analysis	With the help of pre-trained model, Analysing the current hand written digit with the help of DL and CNN algorithm with pre-trained model.
FR-4	Prediction	With the power of pre-trained data and model, The output becomes more accurate.

4.2 NON - FUNCTIONAL REQUIREMENT

Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is user friendly and ease to use which doesn't need any technical knowledge.
NFR-2	**Security**	This software doesn't collect and store user personal details which ensures that there is no possibilities of data leakage and other security concerns.
NFR-3	**Reliability**	It can work on any browser without considering the operating system configuration.
NFR-4	**Performance**	It operates and produces high and quality output within in less time.
NFR-5	**Availability**	It can work 24/7 and it doesn't require any much maintenance.
NFR-6	**Scalability**	Hand-written digit recognition can serve thousands and thousands of users to help them understand the digits better

CHAPTER – 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

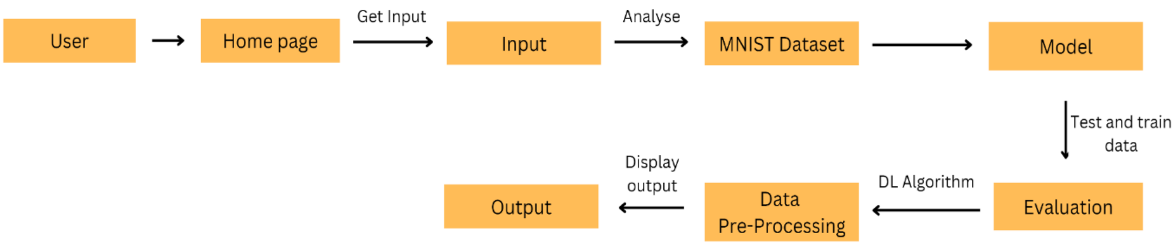


Fig 5.1.1 Data Flow Diagram

5. 2 SOLUTION AND TECHNICAL ARCHITECTURE

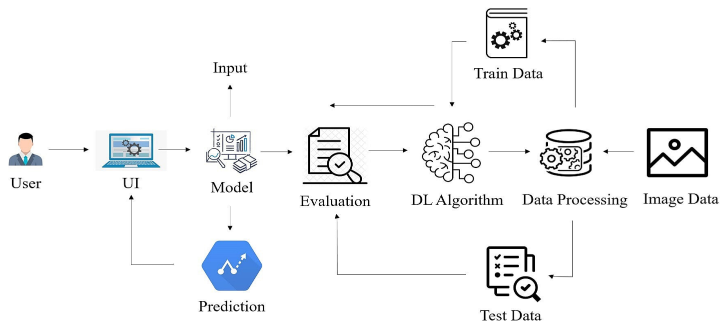


Fig 5.2.1 Solution Architecture

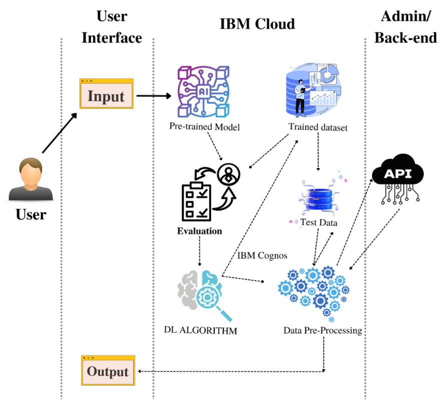


Fig 5.2.2 Technical Architecture

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home page	USN-1	As a user, I can view user guidelines and it's functionality.	I can view the guidelines	Low	Sprint-1
		USN-2	As a user, I can read user manual to understand the process and workflow	I can read the user manual	Low	Sprint-2
		USN-3	As a user, I can watch a video that describe about whole interface work	I can watch the video to understand the process of how to use the platform	Low	Sprint-1
	Input	USN-4	As a user, I can write the digits for prediction.	I can write digit that need to be predicted.	High	Sprint-1
	Recognition	USN-6	As a user, I will able to get the exact and accurate output	I can choose handwritten image from system and predict the output	High	Sprint-2

	Predict	USN-7	As a user, I am allowed to upload handwritten image to predict output.	I can choose the image from their own system	Medium	Sprint-3
		USN-8	As a user, I will get the output with the help of MNIST data pre-processing to ensure maximum accurate result.	MNIST dataset provides input of handwritten digit to provide more insights.	High	Sprint-4
		USN-9	As a user, I can view the accuracy rate of the digit predicted.	I can view accuracy rate	Medium	Sprint-3
Customer (Web user)	Access	USN-10	As a user, I can get to use software virtually and it is user friendly.	I can view awareness of this application and its accessibility.	Low	Sprint-1

CHAPTER – 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMTION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Home Page UI	USN-1	As a user, I can view the home page and I can understand how it works	2	Low	Panneer Selvam B
Sprint-1			Creating the home page		High	
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Muhilan P
Sprint-2	Input of Hand Written digit	USN-3	As a user, I can input any hand written digits.	5	High	Panneer Selvam B
Sprint-2	Processing	USN-4	As a user, Once I upload my image, it will start processing	3	Medium	Ranjith M
Sprint-2			Input image by the user will be trained		High	
Sprint-2			Input image by the user will be tested		High	

Sprint-3			Input image will be evaluated from the trained model		Medium	
Sprint-3	API	USN-5	As a user, I will be able to see the progress of the processing bar	3	Medium	Panneer Selvam B
Sprint-2	IBM Cloudant DB		The input image will be stored in the database		High	
Sprint-3			The Input image will be evaluated with MNIST data which will be fetched from the DB		Medium	
Sprint-4	Output	USN-6	As a user, I will be able to see the desired output	5	High	Ranjith M
Sprint-3	IBM Cloud		Deploy the trained model on the cloud		High	
Sprint-4	Success	USN-7	As a user, I will be able to see the success image once the output is generated	2	Low	Ranjith M
Sprint-1		USN-8	As a user, I will get to know how the process works	2	Low	Muhilan P
Sprint-4		USN-9	As a user, I will be able to watch the video of how to use the software	2	Low	Muhilan P

Sprint-4		USN-10	As a user, I will be able to contribute to this software as an open source	3	Medium	Sakthidhari B
Sprint-3		USN-11	As a user, the uploaded image will get processed in the backend	5	High	Sakthidhari B
Sprint-4	Git and GitHub		Updating details on GitHub		Low	
Sprint-2	Python Flask API		Collecting the data from the API		Medium	
Sprint-2	IBM Watson Studio		Data Pre-processing and optimising the data		High	

Sprint 1

This sprint started with defining the problem statement by going through user stories and use cases. After that, we brainstormed ideas for a home page layout that would best suit our needs. Our first idea was to create a step-by-step guide with screenshots on how to upload an image file and extract text from it in order to get back the numbers on an image file. This layout would not only contain text but also images of what you should do next in case you are stuck, how long will it take until you get your results, etc. - The goal of hand-written digit recognition is to produce a string of interpreted digits which can be used as input for other downstream processing such as statistical models. In this sprint we have developed a home page for users where they can understand how hand written digit recognition works. The page described what it is and how it works, with examples. It also covered different use cases for the software, including which industries the software would be most applicable for.

Sprint 2

In this sprint we have trained models on IBM cloud for our hand written digit recognition using CNN which helps for accuracy. - In this sprint, the deep learning models are stacked in an architecture called Convolutional Neural Network (CNN) to recognize and elucidate handwritten digits in images. The system integrates a classifier and a generator. The generator has to be able to produce data samples which is equivalent of input data for training the classifier. - The CNN architecture predicts the probability of every output pixel being one or zero (i.e., black or white, respectively). - We created a custom application where the users will enter the handwritten digits and it will return the detected digits after recognizing their handwriting with 98% accuracy.

Sprint 3

We have created a webserver using Flask to serve our content generation APIs. This is an implementation of an API for digit recognition which uses data from the TensorFlow Model that we trained in the previous sprint. We can now use this API to generate new text by passing it some input and getting a response back with generated text. - The goal of this sprint was to make a Flask API that can connect with a trained model, which will then be used in an application. This should allow for easy scalability and high efficiency for future projects.

Sprint 4

In this sprint, we have connected the model and home page which we have done in previous sprints. This is a significant change from the previous sprint where we connected the model to generate a feed for us. In this sprint, we have integrated our new model architecture into our existing site, built a new home page and deployed it to our local host. - Now that all the trained model and home page are working in a single UI Page that brings the output with maximum accuracy with the help of various neural networks like Convolutional Neural Networks and MNIST dataset which has 60,000 training images and 10,000 testing images - This concludes deploying a model and linking up our home pages to it in this project.

6.2 SPRINT DELIVERY SCHEDULE

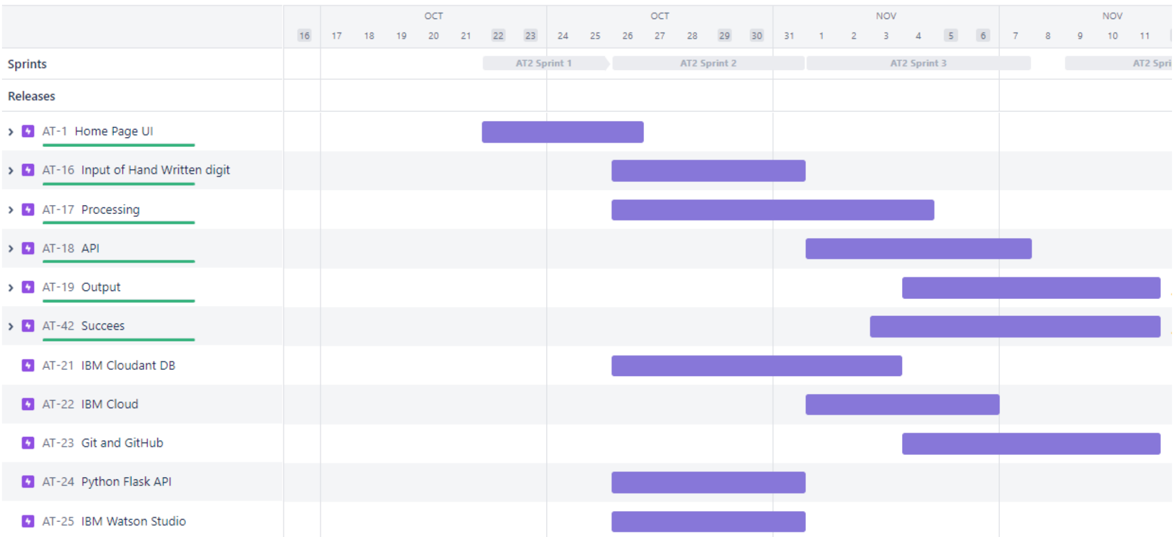
It shows the time taken for each sprint

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	8	4 Days	22 Oct 2022	25 Oct 2022	8	26 Oct 2022
Sprint-2	8	6 Days	26 Oct 2022	31 Oct 2022	8	01 Nov 2022
Sprint-3	8	6 Days	01 Nov 2022	06 Nov2022	8	07 Nov 2022
Sprint-4	11	8 Days	04 Nov 2022	11 Nov 2022	11	14 Nov 2022

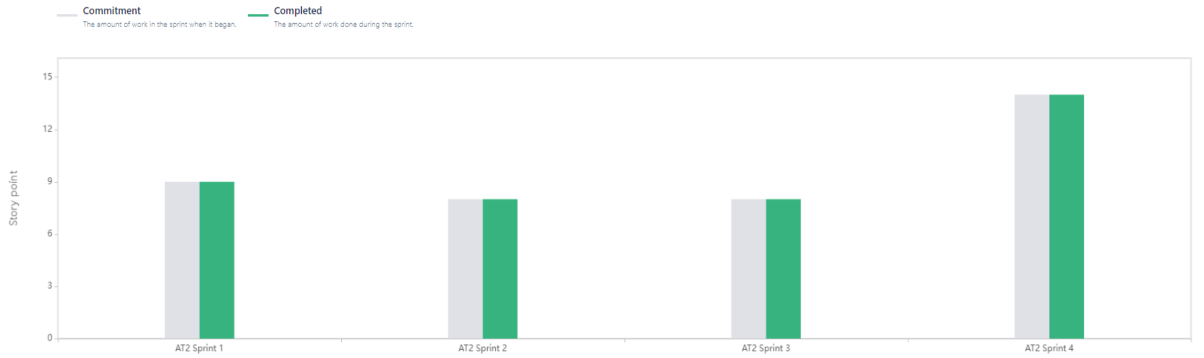
6.3 REPORT FROM JIRA

It shows time taken to completed status of all the issues in the sprint.

Roadmap



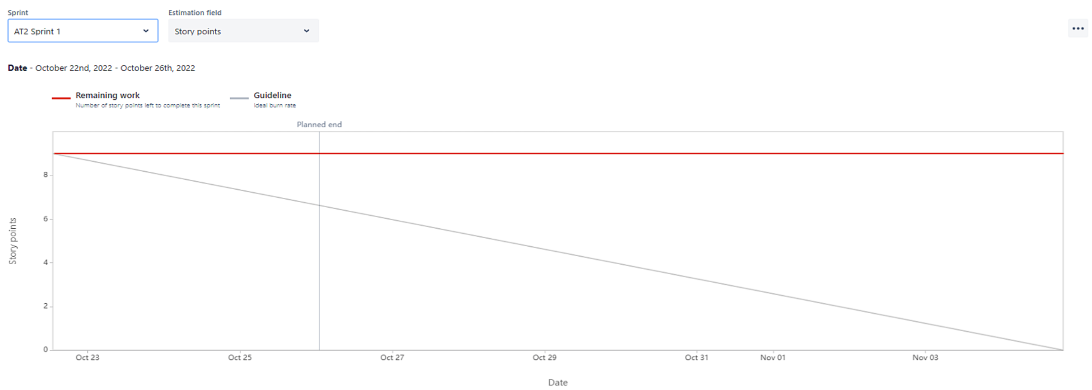
Velocity Report



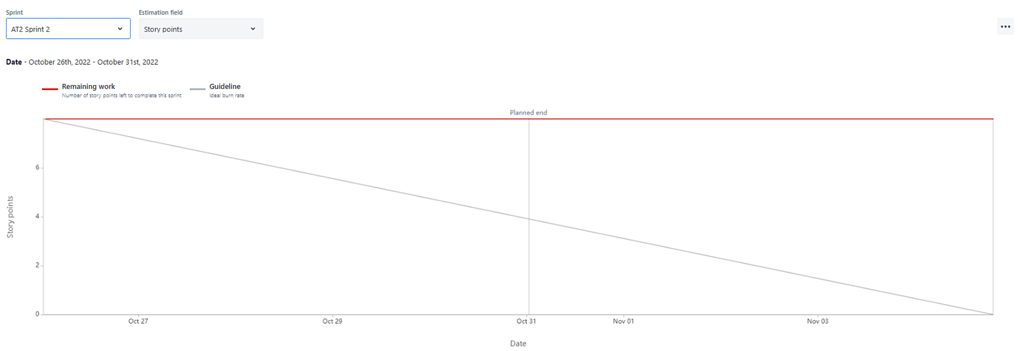
Sprint	Commitment	Completed
AT2 Sprint 1	9	9
AT2 Sprint 2	8	8
AT2 Sprint 3	8	8
AT2 Sprint 4	14	14

Burndown Chart

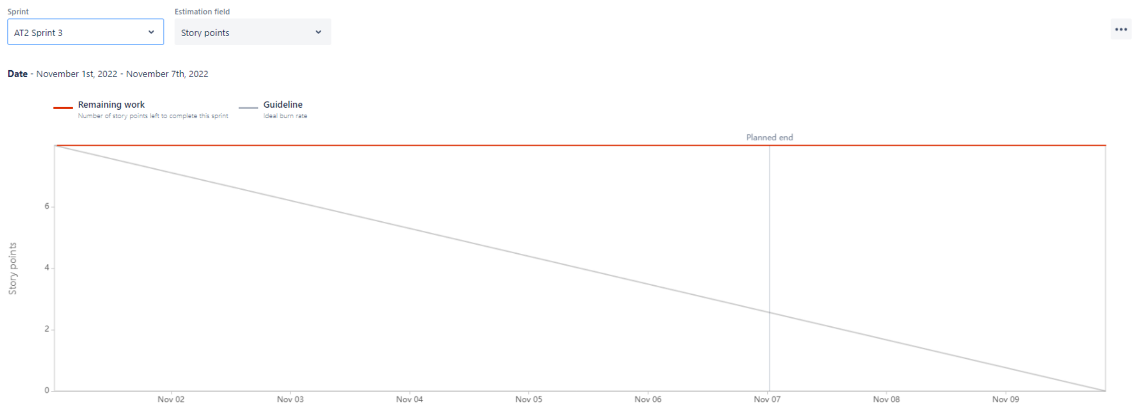
Sprint 1



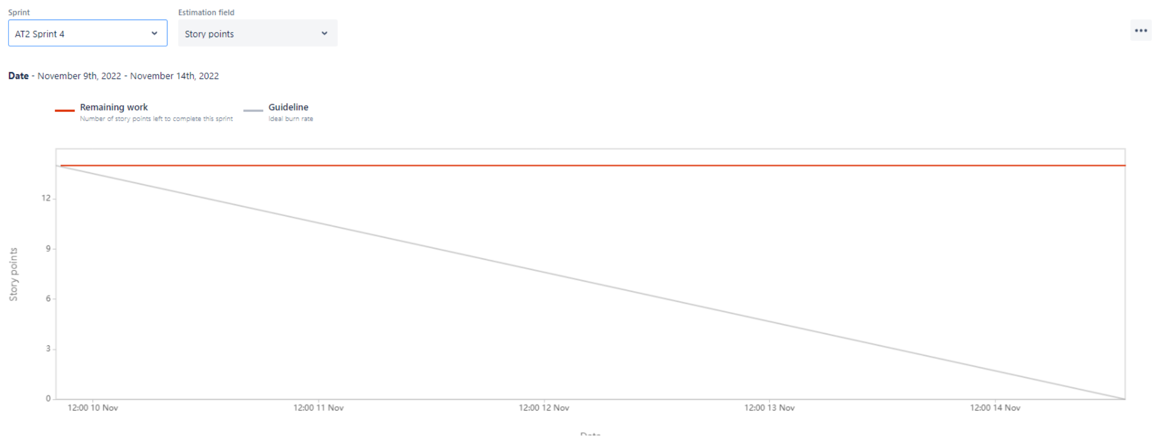
Sprint 2



Sprint 3



Sprint 4



CHAPTER – 7

CODING & SOLUTIONING

7.1 FEATURE 1

Handwritten digit recognition is a computer-based system for extracting and analysing the handwritten digits that have been entered into a text field. It has become easier to recognize handwritten digits because of the advancement in machine learning and pattern recognition algorithms. These algorithms are responsible for recognizing different numbers and letters by using traditional coding for each letter or number. The most common hand written digit recognition systems are optical character recognition (OCR) systems, which use software to interpret what was inputted into the computer, while some use physical sensors to interpret handwriting. Moreover, there are those who use "error-tolerant" methods which can understand even when symbols are not drawn correctly or skipped over altogether.

```
from flask import Flask,render_template,request,redirect
from flask import send_file
import requests
import json
import os
from views.make_prediction import make_prediction

app = Flask(__name__)
app.config["DEBUG"] = True



@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == "GET":
        return render_template('temp.html')
    else:
        file_data = request.files.get('excel_file')
        pred = make_prediction(file_data)
        print(pred)
        return render_template('temp.html', result = pred['response'], keys = pred['keys'], values = pred['values'])

if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

Fig 7.1.1 Coding & Solutioning

7.2 FEATURE

Handwritten digit recognition is the process of converting a hand-drawn number or digit image into a digital format. This conversion is accomplished by identifying and matching the written number against an internally stored collection of handwritten numbers. There are different ways to recognize handwriting, some methods include linear recognizers which identify each stroke as they come, while others use more sophisticated methods that can recognize words and sometimes even sentences. Handwritten digit recognition system is a computer application that can recognize handwritten digits from the images. Handwritten digits can be recognized by a variety of approaches, such as Using hand-drawn patterns and Identifying through images.

```python
from tensorflow import keras
from PIL import Image, ImageOps
from io import BytesIO
import numpy as np
import json

model = keras.models.load_model('static/model_1.h5')
def make_prediction(file):

    image = Image.open(BytesIO(file.read()))
    im2 = ImageOps.grayscale(image)
    img_28 = np.array(im2.resize((28, 28)))
    numpydata = np.asarray(img_28)
    #t = np.reshape(numpydata, [-1, 784])
    t  = numpydata.reshape((1,28,28,1))
    prediction = model.predict(t)
    if  prediction[0][np.argmax(prediction)] > 0.1:
        result = np.argmax(prediction)
        result = str(result)
    else:
        return "NA"
    json_dictionary = {ind:i for ind, i in enumerate(prediction[0].tolist())}
    pred_list = [i*100 for i in prediction[0].tolist()]
    return {"response" : result,"keys":list(json_dictionary.keys()), "values" :pred_list }
```

Fig 7.2.1 Coding & Solutioning

CHAPTER – 8

SYSTEM TESTING

8.1 TEST CASE

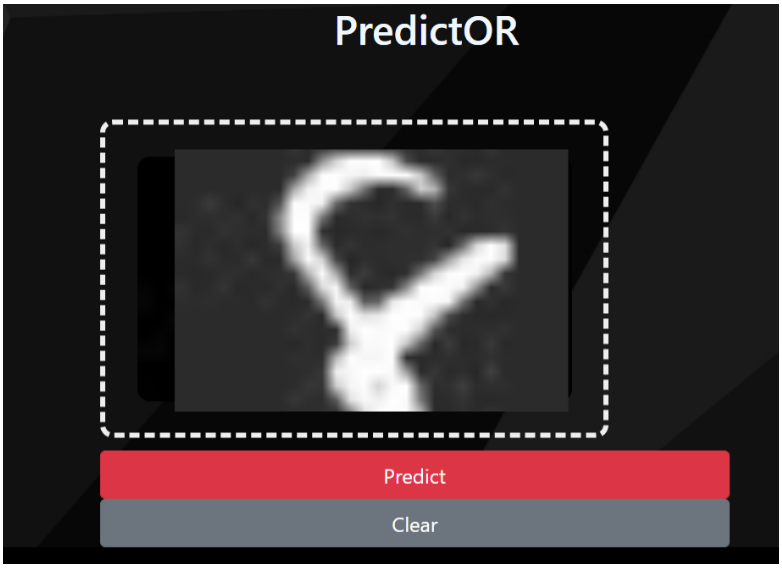


Fig 8.1.1 Test Case - Upload

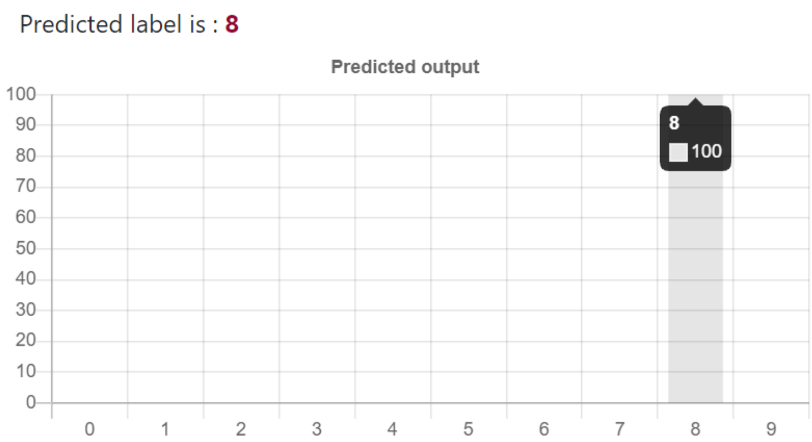


Fig 8.1.2 Test Case - Output

8.2 USER ACCEPTANCE TESTING

8.2.1 Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issuesof the Hand written digit recognition project at the time of the release to User Acceptance Testing (UAT).

8.2.2 Defect Analysis

This reportshows the numberof resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	2	2	3	8
Duplicate	1	0	1	0	2
External	2	3	0	1	6
Fixed	6	2	4	20	32
Not Reproduced	0	1	0	3	4
Skipped	0	1	1	0	2
Won't Fix	0	2	3	2	7
Totals	10	10	11	30	61

8.2.3 Test Case Analysis

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

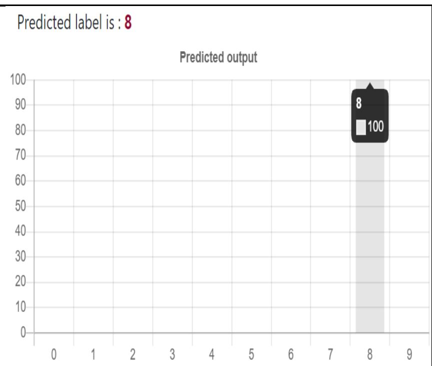
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	28	0	0	28
Security	6	0	0	6
Outsource Shipping	5	0	0	5
Exception Reporting	10	0	0	10
Final ReportOutput	12	0	0	12
Version Control	8	0	0	8

CHAPTER – 9

RESULTS

9.1 PERFORMANCE METRICS

Project team shall fill the following information in model performance testing.

S.No.	Parameter	Values	Screenshot
1.	Model Summary		Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 26, 26, 32) 320 max_pooling2d (MaxPooling2D) (None, 13, 13, 32) 0 conv2d_1 (Conv2D) (None, 11, 11, 64) 18496 max_pooling2d_1 (MaxPooling 2D) (None, 5, 5, 64) 0 flatten (Flatten) (None, 1600) 0 dropout (Dropout) (None, 1600) 0 dense (Dense) (None, 10) 16010 Total params: 34,826 Trainable params: 34,826 Non-trainable params: 0
2.	Accuracy	Training Accuracy - 0.9886 Validation Accuracy - 0.993	

CHAPTER – 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

The proposed system was able to obtain high accuracy on the MNIST 10,000 test dataset. They have considered not only the accuracy, but also the training time, recognition time and memory requirements for entire process. Further, they have identified the digits which were misclassified by the algorithm. The 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. The generative models can perform recognition driven segmentation and the method involves a relatively small number of parameters and hence training is relatively easy and fast.

DISADVANTAGES

The main difficulty in the handwritten digit recognition is different handwritten style which is a very personal behavior where there are a lot of models for numbers based on the angles, length of the segments, stress on some parts of numbers, etc. The disadvantage is that it is not done in real time as a person writes and therefore not appropriate for immediate text input. Applications of offline handwriting recognition are numerous: reading postal addresses, bank check amounts, and forms.

CHAPTER – 11

CONCLUSION

We were able to successfully design and execute a hand written digit recognition software that is useful for recognizing the hand written digit. We used a Convolutional Neural Network to work and were able to achieve an accuracy of 99.9% for the task of recognizing hand written digit. The system will be implemented in Python and use the Matplotlib, NumPy, Pillow library for image processing. It will be done by implementing a system that can recognize handwritten digits. It is an important part of the digitization process. The system has many applications, including in banking and security, where handwritten signatures are required to authenticate transactions.

CHAPTER - 12

FUTURE SCOPE

The task of handwritten digit recognition, using a classifier, has great importance and use such as – online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example - tax forms) and so on. Handwritten digit recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc. This handwritten recognition has high scope in character or pattern recognition which will also help in recognizing various ancient written words.

CHAPTER – 13

APPENDIX

13.1 SOURCE CODE

JAVASCRIPT

```
const accordion = document.getElementsByClassName('contentBx');
    for ( i = 0; i<accordion.length; i++ ){
        accordion[i].addEventListener('click',function(){
            this.classList.toggle('active')
        })
    }
document.querySelectorAll(".drop-zone__input").forEach((inputElement) => {
    const dropZoneElement = inputElement.closest(".drop-zone");
    dropZoneElement.addEventListener("click", (e) => {
        inputElement.click();
    });
    inputElement.addEventListener("change", (e) => {
        if (inputElement.files.length) {
            updateThumbnail(dropZoneElement, inputElement.files[0]);
        }
 dropZoneElement.addEventListener("dragover", (e) => {
        e.preventDefault();
        dropZoneElement.classList.add("drop-zone--over");
    });
    ["dragleave", "dragend"].forEach((type) => {
        dropZoneElement.addEventListener(type, (e) => {
            dropZoneElement.classList.remove("drop-zone--over");
        });
    });
    dropZoneElement.addEventListener("drop", (e) => {
        e.preventDefault();

        if (e.dataTransfer.files.length) {
```

```
            inputElement.files = e.dataTransfer.files;
            updateThumbnail(dropZoneElement, e.dataTransfer.files[0]);
        }
        dropZoneElement.classList.remove("drop-zone--over");
    });
});

/**
 * Updates the thumbnail on a drop zone element.
 *
 * @param {HTMLElement} dropZoneElement
 * @param {File} file
 */
function updateThumbnail(dropZoneElement, file) {let thumbnailElement = dropZoneElement.querySelector(".drop-zone__thumb");
```

PYTHON FILE

```
from flask import Flask,render_template,request,redirect
from flask import send_file
import requests
import json
import os
from views.make_prediction import make_prediction
app = Flask(__name__)
app.config["DEBUG"] = True
@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == "GET":
        return render_template('temp.html')
    else:
        file_data = request.files.get('excel_file')
        pred = make_prediction(file_data)
        print(pred)
        return render_template('temp.html', result = pred['response'], keys = pred['keys'], values = pred['values'])
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0')


from tensorflow import keras
from PIL import Image, ImageOps
from io import BytesIO
import numpy as np
import json

model = keras.models.load_model('static/model_1.h5')
def make_prediction(file):

    image = Image.open(BytesIO(file.read()))
    im2 = ImageOps.grayscale(image)
```

13.2 GITHIB AND PROJECT DEMO LINK

GitHub Link : **https://github.com/IBM-EPBL/IBM-Project-13836-1659532949**

Demo Link : **https://youtu.be/wblNinMoa0w**

CHAPTER - 14

REFERENCES

[1] Akhlaghi, M., Ghods, V.: Farsi handwritten phone number recognition using deep learning. SN Appl. Sci. 2(3), 1 (2020). [CrossRef]

[2] Saabni, R.: Recognizing handwritten single digits and digit strings using deep architecture of neural networks . In: 2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR), pp.1–6. IEEE, September (2016).

[3] Han Xiaofeng and Li Yan, "The Application of Convolution Neural Networks in Handwritten Numeral Recognition", International Journal of Database Theory and Application, vol. 8, no. 3, pp. 367-376, (2015).

[4] Niu n Xiao-Xiao and Y. Suen Ching,"A novel hybrid CNN-SVM classifier for recognizing handwritten digits " , Elsevier, vol. 45, no. 4, pp. 1318-1325, April (2012).

[5] Goltsev A., Gritsenko V. Investigation of the efficient features for image recognition by Neural Networks, 28, pp. 15-23, (2012).

[6] Shuai Tan and Zhi Tan, "Improved LeNet-5 Model Based On Handwritten Numeral Recognition [C]", The 31st Chinese Control and Decision Conference, pp. 6474-6477, (2019).

[7] R. Shruti Kulkarni and R Bipin,"Spiking neural networks for handwritten digit recognition – Supervised learning and network optimization", Neural Networks, vol. 103, pp. 118-127, (2018).