

# Guide's Approval:

The screenshot shows a Gmail interface on a desktop browser. The address bar displays the URL: `mail.google.com/mail/u/1/#sent/QgrcJHsHkwwGZZgnwcknQCQSVxkRtpzGFGwQ`. The Gmail logo and search bar are at the top. The left sidebar contains navigation links: Compose, Inbox (3,853), Starred, Snoozed, Sent (selected), Drafts, and More. Below these are sections for Meet (New meeting, My meetings) and Hangouts (E Nithyashri, No recent chats, Start a new one). The main inbox area shows two emails. The first email is from E Nithyashri (18BCS0070) to Vanitha, dated Apr 16, 2021, 8:32 PM. The subject is "Regarding Capstone project's review-2 document approval". The body text reads: "Good evening,I'm E.Nithyashri(18BCS0070), your mentee for the Capstone project.Our topic is "HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING". I hereby submit the document for verification and your approval. I request you to add your signature for the same. Thank you." Below the text is a document preview titled "Capstone Review-2..." with a thumbnail showing the document's header: "HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING" and "BACHELOR OF SCIENCE IN COMPUTER SCIENCE". The second email is from Vanitha M to E Nithyashri, dated Apr 17, 2021, 10:06 AM. The subject is "Review2 document verified on 17.4.2021." The body text reads: "Review2 document verified on 17.4.2021."

*A project report on*

# **HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING**

*Submitted in partial fulfillment for the award of the degree of*

## **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

*By*

**VASANTHAKUMAR.P(18BCS0022)**

**MANIBHARATHI.S(18BCS0023)**

**E.NITHYASHRI(18BCS0070)**



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF INFORMATION TECHNOLOGY &  
ENGINEERING (SITE)**

June,2021

# **HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING**

*Submitted in partial fulfillment for the award of the degree of*

**Bachelor of Science in Computer Science**

*by*

**VASANTHAKUMAR.P(18BCS0022)**

**MANIBHARATHI.S(18BCS0023)**

**E. NITHYASHRI (18BCS0070)**



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF INFORMATION TECHNOLOGY &  
ENGINEERING (SITE)**

June,2021

**DECLARATION**

I hereby declare that the thesis entitled “HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING” submitted by me, for the award of the degree of Bachelor of Science in Computer Science VIT is a record of bonafide work carried out by me under the supervision of Dr.Vanitha.M.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date:

**Signature of the Candidate**

## **CERTIFICATE**

This is to certify that the thesis entitled “HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING” submitted by E.NITHYASHRI (18BCS0070) School of Information Technology & Engineering (SITE) VIT, for the award of the degree of Bachelor of Science in Computer Science is a record of bonafide work carried out by her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.

**Signature of the Guide**

**Signature of the HoD**

**Internal Examiner**

**External Examiner**

## **ABSTRACT**

Handwritten Digit recognition is one of the emerging fields in the current technical world. One of the most abilities of humans is they will recognize any object or thing. So handwritten transcriptions can easily be identified and understood by humans, but computers cannot identify as easily as humans because different handwriting has different patterns to spot. Thus handwritten transcription cannot be identified by the machine and it is difficult to spot the script by the system.

This project is built based upon the most vital and integral concepts of Deep learning and Conventional Neural Networks, along with the essential libraries and tools like Tensorflow, Keras. The input image is processed and the features are extracted. Further, in this process of text recognition, classification schema along with training the system is done so that the system can acknowledge the input. As the system is trained, it has the capability to seek out the similarities and the differences among various handwritten samples. Finally the application takes the image of a handwritten input and converts it into a digital transcript and gives it as the output.

## **ACKNOWLEDGEMENT**

It is my pleasure to express with deep sense of gratitude to Dr.Vanitha.M, School of Information Technology and Engineering (SITE), Vellore Institute of Technology, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Data Science.

I would like to express my gratitude to Dr.G.Viswanathan , Mr. G.V.Selvam, Dr. Rambabu Kodali, Dr. S. Narayanan, and Dr. Balakrushna Tripathy, School of Information Technology and Engineering(SITE), for providing an environment to work in and for this inspiration during the tenure of the course.

In a jubilant mood I express ingeniously my whole-hearted thanks to Dr.Senthil Kumaran U. Project Coordinator, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date:

Name of the student

**E.Nithyashri**

# **CONTENTS**

<b>CONTENTS</b>	iv
-----------------	----

<b>LIST OF FIGURES</b>	ix
------------------------	----

<b>LIST OF TABLES</b>	xi
-----------------------	----

<b>LIST OF ACRONYMS</b>	xii
-------------------------	-----

## **CHAPTER 1**

### **INTRODUCTION**

1.1 AIM	1
1.2 OBJECTIVES	1
1.3 SCOPE OF THE PROJECT	1

## **CHAPTER 2**

### **BACKGROUND AND REQUIREMENTS**

2.1 LITERATURE SURVEY	2
2.2 HARDWARE AND SOFTWARE REQUIREMENTS	6
2.2.1 HARDWARE REQUIREMENTS	6
2.2.2 SOFTWARE REQUIREMENTS	6

## **CHAPTER 3**

### **DETAILED DESIGN**

3.1 SYSTEM ARCHITECTURE	8
-------------------------	---



3.2 ER DIAGRAM	9
3.3 DFD DIAGRAM	9
3.4 USE CASE DIAGRAM	10
3.5 UML DIAGRAM	11
3.6 MODULE DESCRIPTION	11
3.6.1 DATA SET	11
3.6.3 DATA VISUALIZATION	12
3.6.4 DATA PREPROCESSING	12
<b>CHAPTER 4</b>	
<b>IMPLEMENTATION</b>	
4.1 IMPORTING AND PLOTTING OF DATASET	13
4.2 CNN MODEL	15
4.2.1 SUMMARY OF THE MODEL	18
4.2.2 TESTING	19
4.3 SVM MODEL	20
4.3.1 SUMMARY OF THE MODEL	22
4.2.2 TESTING	22
<b>CHAPTER 5</b>	
<b>CONCLUSION &amp; FUTURE WORK</b>	<b>24</b>
<b>APPENDICES</b>	<b>25</b>
APPENDIX 1	
APPENDIX 2	
<b>REFERENCES</b>	<b>26</b>

::

*iii*

## **LIST OF FIGURES**

1.1 INFRA STRUCTURE BASED NETWORK	1
1.2 INFRA STRUCTURE LESS NETWORK	2
1.3 MOBILE AD-HOC NETWORK	4

## **LIST OF TABLES**

2.1 MULTIPATH ROUTING PROTOCOLS IN AMNETS 22

3.1 SIMULATION PARAMETERS 48

## **LIST OF ACRONYMS**

CNN Convolutional Neural Network

SVM Support Vector Machine

## **Chapter 1**

### **Introduction**

#### **1.1. AIM:**

This project is aimed to train a Machine learning model which can take an Image of a handwritten digit as an input and recognize it. The model is built on top of Convolutional Neural Network(CNN) with the use of MNIST data set for training and testing .

#### **1.2 OBJECTIVES :**

1. The main objective is to recognize the handwritten digits only.
2. Reduced man-power by recognizing the handwritten numbers and producing the digitized form of the digit. So that the old handwritten data in the banks, post offices etc., can be recovered and stored in the databases.
3. Proposed system serves as a guide and works in digit recognition areas.

#### **1.3 SCOPE OF THE PROJECT:**

1. System is designed in a way to ensure that offline handwritten recognition of numeric digits is done.
2. Old and epic handwritten digits are recognized and are restored in digital form.
3. Use of Neural Network for classification.
4. Large numbers of training data sets will improve the efficiency of the suggested approach as the model gets trained very well.

## Chapter 2

### Background and Requirements

#### 2.1 LITERATURE SURVEY:

[1] Comparison between the online and offline handwriting recognition using digitizer technology and various algorithms like shape recognition algorithms, preprocessing and post processing techniques, and experimental systems are done and discussed.

[2] Four different approaches are used to combine several individual classifiers like Bayesian, k-nearest-neighbor to obtain the recognition of totally unconstrained handwritten numerals with 95-98.9% recognition with 0-0.90% substitution and 0.2-5% rejection.

[3] Recognition of unconstrained handwritten text is very challenging as it is hard to segment the data and the data may contain long-range bidirectional inter-dependencies. Instead of using the most approached Markov models for speech and handwriting recognition, this paper relies on a novel type of recurrent neural network(RNN) which overcomes the challenging tasks and gives accuracy of 79.7% on online data and 74.1% on offline data.

[4] This survey describes the nature of both online and offline cases of handwritten language and its transduction into electronic data. Some basic concepts and algorithms of preprocessing, character and word recognition are discussed.

[5] Using Hidden Markov model(HMM), sub-character stroke types are modeled and are concatenated and embedded to form the stochastic language model. In spite of the fact that a stochastic language model for handwriting recognition using HMM is still in development, this survey has obtained an independent recognition rate of 94.5% on 3,823 unconstrained handwritten word samples from 18 writers covering a 32 word vocabulary.

[6] This paper is based on Lancaster-Oslo/Bergen (LOB) corpus that comprises 1 million word instances and a database with 1066 forms by 400 writers, which is a total of 82227 word instances out of 10841 words in the vocabulary. The forms are segmented into lines and words using image-processing procedures.

[7] Using a multi-state time delay neural network, a recognition engine named NPen++ was developed at the University of Karlsruhe and Carnegie Mellon University that gives recognition rates from 96% for a 5,000 word dictionary to 93.4% on a 20,000 word dictionary and 91.2% for a 50,000 word dictionary. This engine can be run with large dictionaries as tree

search and pruning techniques reduce the search space without attenuating the recognition performance.

[8] Recurrent neural networks (RNNs) with Long Short-Term memory cells gives best results in unconstrained handwriting recognition and dropout approach gives superior performance in convolutional networks. This paper has used both these techniques for the very first time in such a way that dropout doesn't affect the recurrent connections so that modeling sequences are preserved.

[9] It is an offline Roman cursive handwriting recognition system where the input is provided in form of a digit or a word or text. After a number of processing procedures, the ASCII format of the input is obtained as the output.

[10] This paper is a synopsis of the Markov-model that is widely used for offline handwriting recognition. Markov-chain or n-gram models and the crucial concepts about the hidden Markov-model(HMM) are discussed.

[11] Using a hidden Markov Model(HMM) continuous speech recognition system is applied to an online cursive handwriting recognition. A 1.1% word error rate is achieved for a 3050 word lexicon, 52 character, writer-dependent task and 3%-5% word error rates are obtained for six different writers in a 25,595 word lexicon, 86 character, writer-dependent task. This result was obtained by simply using the handwriting ingredients instead of speech.

[12] A database based on Lancaster-Oslo/Bergen(LOB) corpus, which consists of English sentences that includes 556 forms by 250 writers are used for off-line handwriting recognition. This paper focuses on using linguistic knowledge over the lexicon level. These forms undergo Preprocessing and text segmentation steps for obtaining the promising result.

[13] Least squares SVM (LS-SVM) which is easier to train as it requires only the solution to a convex linear problem, is based on the margin-maximization principle performing structural risk minimization. This paper discusses the model selection for the LS-SVM using an empirical error criterion. This classifier is used in handwritten character recognition and also greatly demonstrates how a model selection can Improve the performance.

[14] This paper discusses character termination using a finite state automaton where an unknown stroke is compared with the database of strokes to match and identify the stroke. This system is used in online recognition of handwritten Tamil characters as a handwritten character is constructed via sequence of strokes.

[15] This paper discusses the system architecture and novel components that are used on powerful machines for online handwriting recognition systems in the cloud as well as on mobile devices with a bit of changes in the settings of the system. Other components such as unified time and position-based input interpretation, trainable segmentation, minimum-error rate training for feature combination, and a cascade of pruning strategies are used in this system, which currently supports 22 scripts and 97 languages. This system focuses most on the fast, touch-enabled devices with high-accuracy text entry for mobile.

[16] On binary images, a set of robust language independent features are extracted and then, lower and upper baselines subsets are derived so that the word variability will be assessed. This approach of offline handwriting recognition system using 1D HMM learns the model without even undergoing the pre-segmentation process.

[17] This paper promises that their HMM-based system for offline handwriting recognition outperforms the current state-of-the-art approach for English and French writings using sturdy methods like image object recognition: moment-based image normalization, writer adaptation, discriminative feature extraction and training, and open-vocabulary recognition.

[18] This paper uses a novel technology for 3D handwriting recognition where accelerometers and gyroscopes are attached to the users' back of the hands. When the user writes in the air, these gestures are captured by the motion sensors for spotting and recognizing those gestures in further stages. A Support vector machine(SVM) is used to identify the data segments whereas the Hidden Markov models(HMM) are used to generate a text representation from the captured gestures. These individual characters are concatenated to word models.

[19] A two-pass N-best approach is taken on several UNIPEN data sets for a writer independent handwriting recognition system. This system is robust because of the combination of signal normalization preprocessing and other perpetual features. The accuracy is improved due to the melding of point oriented and stroke oriented features.

[20] This paper discusses comparing and understanding the eastern(Japanese) and western character recognition like the preprocessing, classification and post-processing stages. This lets us understand the similar and non-similar foundation, and to develop the compact modules for integrated systems that recognize multi-language writings.

[21] This paper compares the two strategies of SVM i.e., "one against one" and "one against all" to find out the difference and the significance. It appears that the "one against all" strategy is more significant for digit recognition whereas "one against one" is comparatively



easier to train and can be used for a very large number of classes. But when it comes to upper-case letters, there is a very less difference between these two strategies. Also when compared to MLP, SVM's strategies show a better estimation than MLP.

[22] A bidirectional recurrent neural network with long short-term memory blocks approach is used alongside an objective function called Connectionist Temporal Classification (CTC) for online handwriting recognition. This system achieves 74.0% of word recognition while a system built based on HMM gives 65.4% of word recognition.

[23] This system outperforms all the other systems for the recognition of French handwriting based on three different recognition technologies. These technologies include a framework of multi-word recognition based on weighted finite state transducers, an explicit word segmentation, and a combination of isolated word recognizes and a language model.

[24] This paper discusses the application of neuroevolution and all the way till CNN topologies and therefore developing solutions based on genetic algorithms and grammatical evolution. MNIST data set consists of handwritten digits that are difficult even for humans to recognize. In spite of this, the system has produced an impressive result without any kind of data augmentation or preprocessing methods.

[25] Feature selection-based combination and class dependent features are used for handwriting recognition in this paper. Based upon their class separation and recognition capabilities, features are evaluated. Also a new feature vector is constructed using multiple feature vectors. The results of the new feature vector helps in upgrading the recognition rate and also the selected features are effective in separating pattern classes.

[26] This paper aims at discussing the problems in handwritten symbols, new neural network models (NNM). Two subsystems of neural networks are used particularly for gender classification for which the biometric system for Automated Fingerprint Information Retrieval System (AFIRS) is also designed and developed.

[27] CASIA-HWDB/OLHWDB databases are used as the trained set for the following five tasks- classification on extracted features, online/offline isolated character recognition, online/offline handwritten text recognition and obtained 93.89%, 94.77%, 97.39%, 88.76%, and 95.03% from test results respectively .

[28] For a modified CNN-RNN hybrid architecture, training is done using efficient initialization of the network using synthetic data, image normalization for slant correction, and domain specific data transformation and distortion for learning the in-variants. This proposal is

for recognizing the handwriting for scanned offline document images.

[29] Writing styles can vary from self-combined to various other styles. This may cause problems and that is why Genetic algorithms are proposed. A pool of images of the characters are converted into images. Graph of each character is blended so that an intermediate style of the parent character is obtained. This method has achieved an accuracy of 98.44%.

[30] A deep neural network architecture that supports for 102 languages, has put back the segment-and-decode-based system by reducing the error rate by 20–40%. This system recognizes 10 times faster than the previous system as it combines the sequence recognition with a new input encoding using Bézier curves. This paper discloses that for both the open and closed data sets, new state-of-the-art results on IAM-OnDB.

## 2.2 HARDWARE AND SOFTWARE REQUIREMENTS:

### 2.2.1 HARDWARE REQUIREMENTS:

1. Windows 10 with Intel Core 3 Duo/Quad/hexa/Octa or higher end 64 bit processor PC or Laptop (Minimum operating frequency of 2.5GHz)

2. Hard Disk capacity of 500GB - 4TB with 4 GB RAM or above.

3. 10 Gigabit Ethernet or Bonded Gigabit Ethernet

### 2.2.2 SOFTWARE REQUIREMENTS:

#### Functional requirements:

1. Modified National Institute of Standards and Technology(MNIST) dataset is imported and is saved in the same directory as the program.

2. A model is created. This model takes the value of the color in pixels and puts them in a one-dimensional array. A Neural network is built and activated in such a way that the number of nodes in the input layer is equal to the number of pixels in the arrays.

3. From the testing set, accuracy is tested.

4. For recognizing the handwritten digit, the user has to input the value on an image of dimension 28 x 28 then, the value of the number written on the image is predicted in real-time.

Non-functional requirements:

1. Python
2. Numpy ( Library to handle complex Array operations )
3. Keras ( Machine learning Library )
4. Tensorflow ( Machine learning Library )
5. Sklearn ( Machine learning Library )

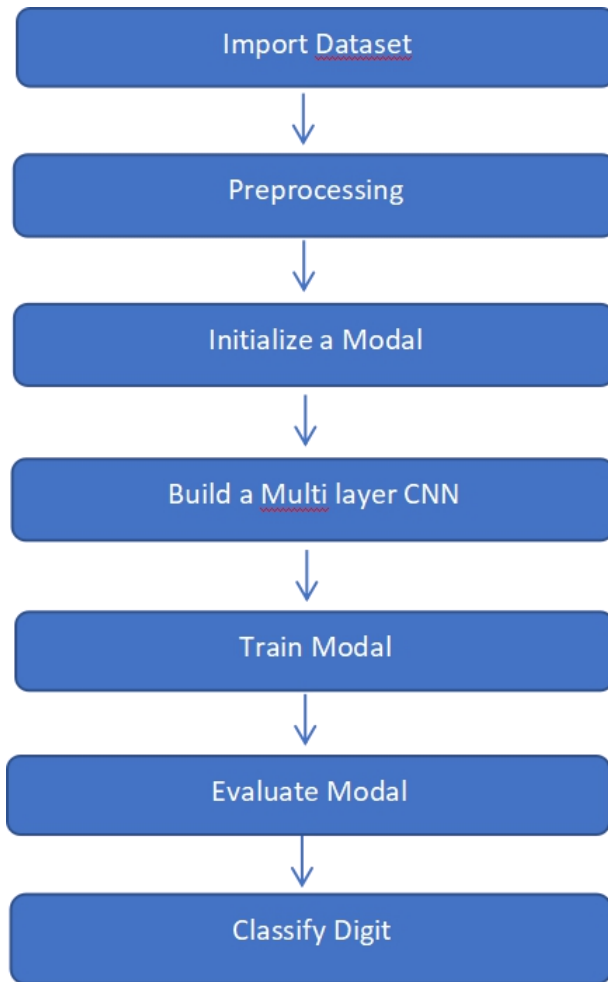
System constraints:

System takes no part in server error because the network provider can shut down the network for small maintenance work.

## Chapter 3

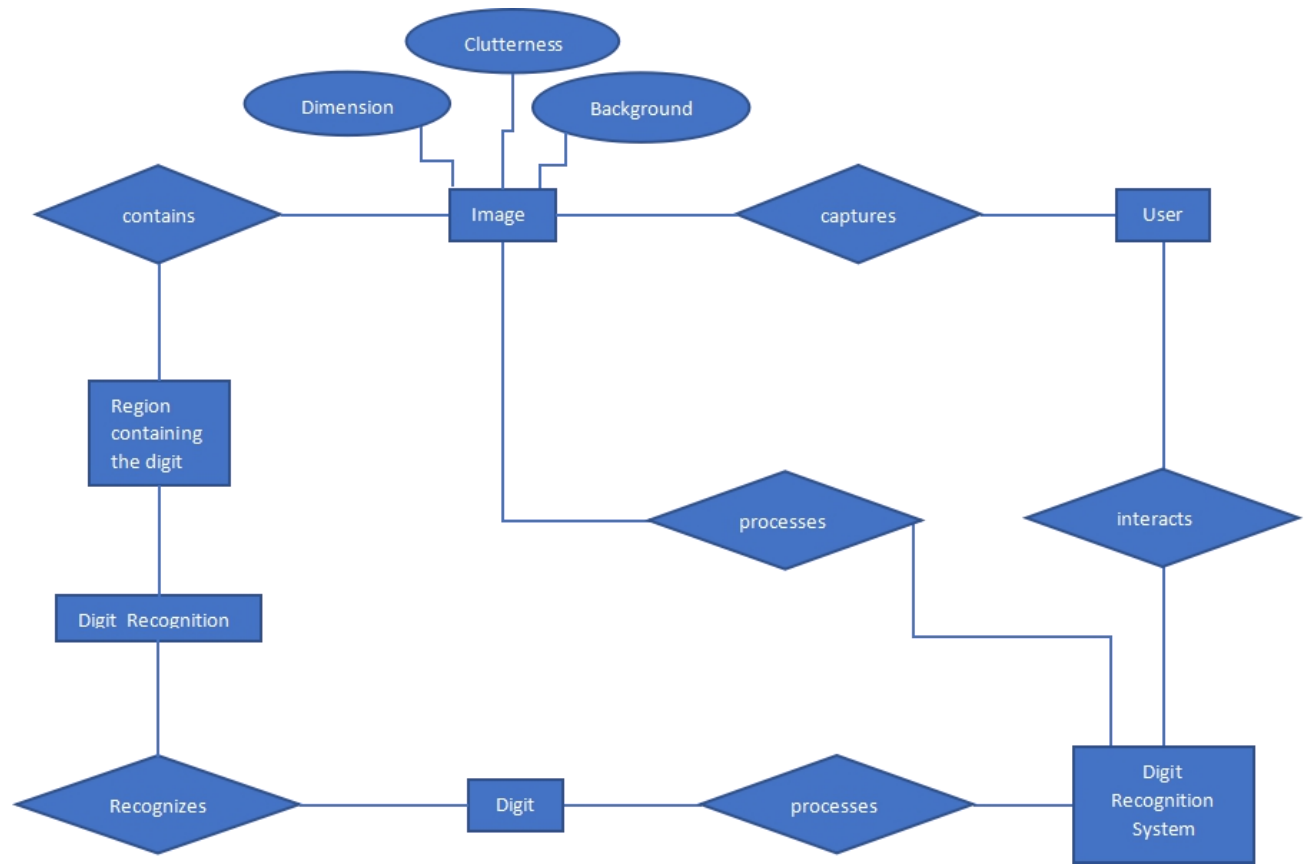
### Detailed Design

#### 3.1 SYSTEM ARCHITECTURE:



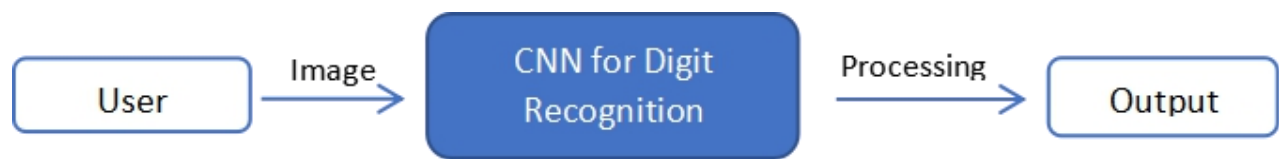
An MNIST dataset is imported and preprocessed. A model is built and initiated and for training and testing the imported dataset. A multi-layer CNN is built and then the dataset is trained in the model. The model is tested and then evaluated based upon the results obtained from the digit classification. Accuracy is predicted and the model is summarized.

### 3.2 ER DIAGRAM:

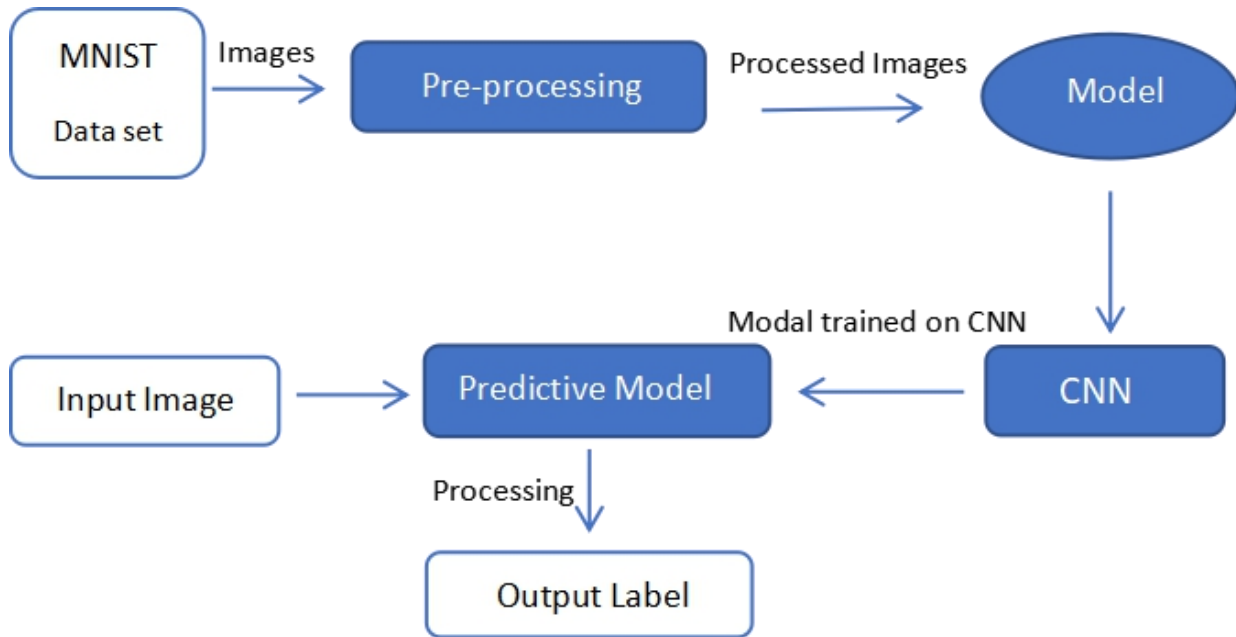


### 3.3 DATA FLOW DIAGRAM(DFD):

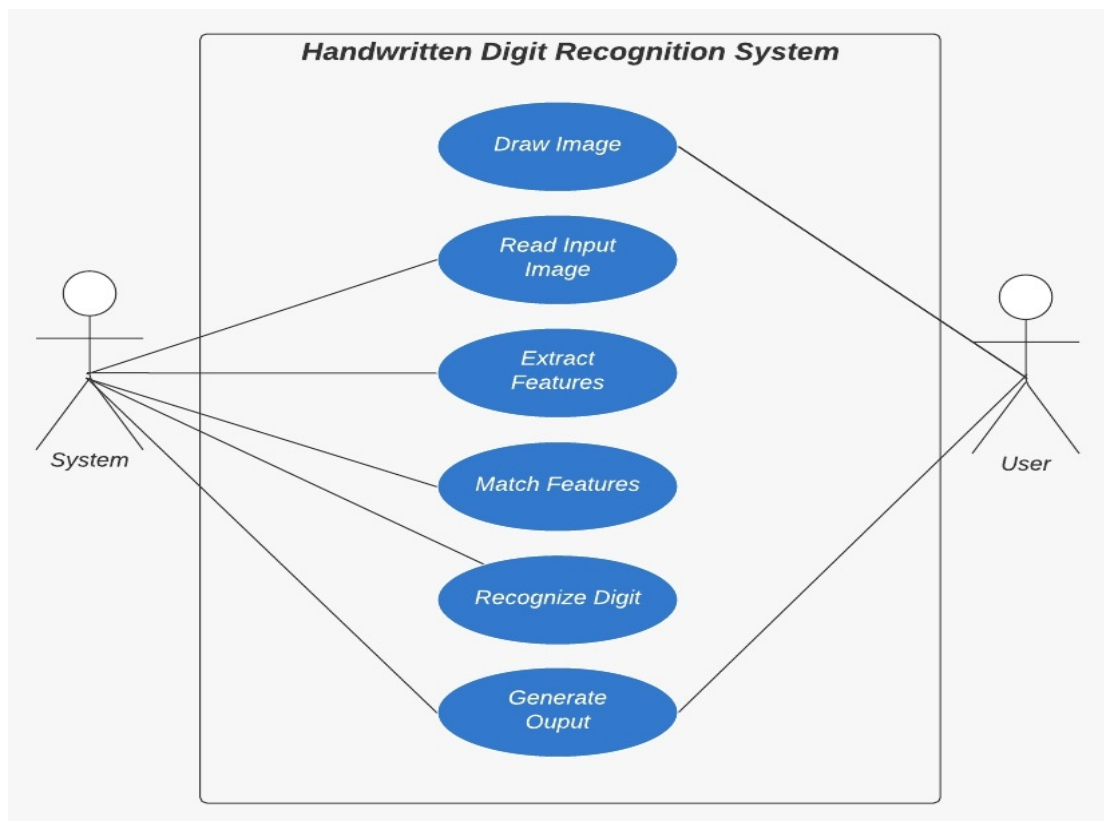
Level 1:



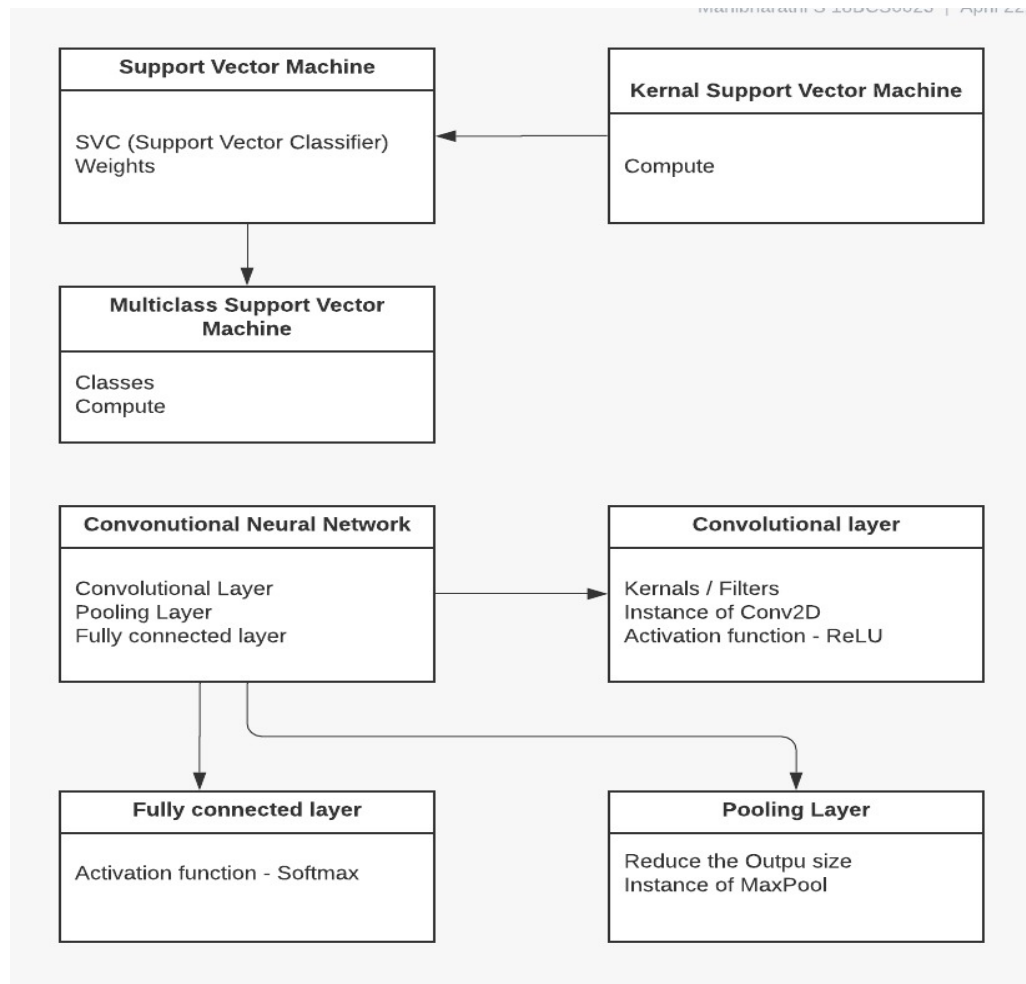
Level 2:



### 3.4 USE CASE DIAGRAM:



### 3.5 UML DIAGRAM:



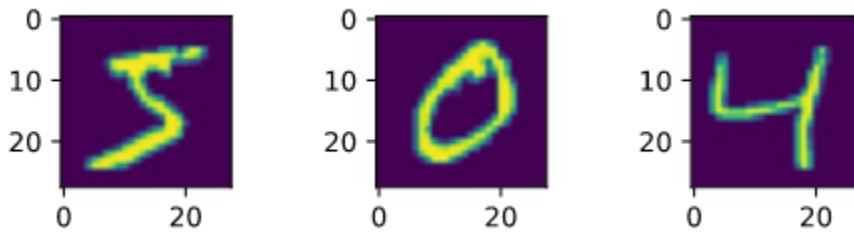
### 3.4 MODULE DESCRIPTION:

#### 3.6.1 DATA SET:

Handwritten character recognition is a research area that already contains detailed ways of implementation which include major learning datasets. MNIST data set (Modified National Institute of Standards and Technology database) is the subset of the NIST data set. MNIST is a combination of two of NIST's databases Special Database 1 and Special Database 3, which consist of digits written by high school students and employees of the United States Census Bureau, respectively. MNIST has a total of 70,000 handwritten digit images, out of which 60,000 are used for training set and 10,000 are used for test set. Each image has a dimension of 28x28 pixel and is anti-aliased. All these images have corresponding label values which tells the value of the digit.

### 3.6.3 DATA VISUALIZATION

#### Implementation:



Sample images from MNIST Data set

### 3.6.4 DATA PREPROCESSING

Initially, the data set is separated into four different categories namely trainX, trainY, testX, testY such that X represents the feature and Y represents the label. Since a digit always comes within a category of 0 to 9, the label can be converted from a simple digit to an array of size 10 where the index which represents the digit is given 1 and the rest of the indexes are given a value of 0. For example 5 would be represented as [0,0,0,0,0,1,0,0,0,0]. Then a normalization process is done, where each pixel is converted from a value range of 0 to 255 to a value range of 0 to 1. Pixels are converted into float type and then divided by 255.



## Chapter 4

# Implementation

### 4.1 IMPORTING AND PLOTTING OF DATASET:

```
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot
# Data set is Downloaded from Internet
(trainX, trainY), (testX, testY) = mnist.load_data()
print(f'Training: ')
print(f'x: ', trainX.shape, end='\n')
print(f'y: ', trainY.shape)
print(f'Testing: ')
print(f'x: ', testX.shape, end='\n')
print(f'y: ', testY.shape)
# Plot first few Training Images
for i in range(3):
    pyplot.subplot(330 + 1 + i)
    pyplot.imshow(trainX[i])    # cmap=pyplot.get_cmap('gray') --> to make preview as
grey scale
pyplot.show()
for i in range(3):
    print(trainY[i], end=" ")
Training:
x: (60000, 28, 28)    y: (60000,)
Testing:
x: (10000, 28, 28)    y: (10000,)

from tensorflow.keras.utils import to_categorical
# reshape data set to have a single channel
trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
testX = testX.reshape((testX.shape[0], 28, 28, 1))
```

```
trainY = to_categorical(trainY)
```

```
testY = to_categorical(testY)
```

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

# Dataset is Downloaded from Internet
(trainX, trainY), (testX, testY) = mnist.load_data()

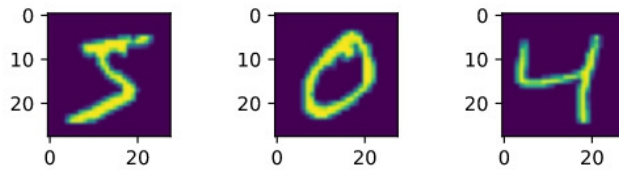
print(f'Training: ')
print(f'x: ', trainX.shape, end='\n')
print(f'y: ', trainY.shape)

print(f'Testing: ')
print(f'x: ', testX.shape, end='\n')
print(f'y: ', testY.shape)

# Plot first few Training Images
for i in range(3):
    pyplot.subplot(330 + 1 + i)
    pyplot.imshow(trainX[i]) # cmap=pyplot.get_cmap('gray') --> to make preview as grey scale
pyplot.show()

for i in range(3):
    print(trainY[i], end=" ")
```

```
Training:
x: (60000, 28, 28)      y: (60000,)
Testing:
x: (10000, 28, 28)      y: (10000,)
```



```
5 0 4
```

```
from tensorflow.keras.utils import to_categorical

# reshape dataset to have a single channel
trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
testX = testX.reshape((testX.shape[0], 28, 28, 1))

trainY = to_categorical(trainY)
testY = to_categorical(testY)
```

[illegible]

## 4.2 CNN MODEL:

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

```
from tensorflow.keras.utils import to_categorical
```

```
from tensorflow.keras.models import Sequential,load_model
```

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

```
from tensorflow.keras.layers import MaxPooling2D
```

```
from tensorflow.keras.layers import Dense
```

```
from tensorflow.keras.layers import Flatten
```

```
from tensorflow.keras.optimizers import SGD
```

```
def load_dataset():
```

```
    (trainX, trainY), (testX, testY) = mnist.load_data()
```

```
    # reshape dataset to have a single channel
```

```
    trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
```

```
    testX = testX.reshape((testX.shape[0], 28, 28, 1))
```

```
    trainY = to_categorical(trainY)
```

```
    testY = to_categorical(testY)
```

```
    return trainX, trainY, testX, testY
```

```
def process_pixels(train, test):
```

```
    # convert from integers to floats
```

```
    train_norm = train.astype('float32')
```

```
    test_norm = test.astype('float32')
```

```
    # normalize the pixel range from 0-255 to range 0-1
```

```
    train_norm = train_norm / 255.0
```

```
    test_norm = test_norm / 255.0
```

```
    return train_norm, test_norm
```

```
def create_modal():
```

```
    model = Sequential()
```

```
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform',  
input_shape=(28, 28, 1)))
```

```

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform'))

model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform'))

model.add(MaxPooling2D((2, 2)))

model.add(Flatten())

model.add(Dense(100, activation='relu', kernel_initializer='he_uniform'))

model.add(Dense(10, activation='softmax'))

# compile model

opt = SGD(lr=0.01, momentum=0.9)

model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

return model

```

**def evaluate\_model():**

```

# load dataset

trainX, trainY, testX, testY = load_dataset()

# prepare pixel data

trainX, testX = process_pixels(trainX, testX)

# load model

model = load_model('/content/new_model.h5')

_,acc = model.evaluate(testX,testY,verbose=0)

print(f'Accuracy: {(acc*100.0):.3f}')
```

print(model.summary())

**def main():**

```
trainX, trainY, testX, testY = load_dataset()

trainX, testX = process_pixels(trainX, testX)

model = create_model()

model.fit(trainX, trainY, epochs=10, batch_size=32, verbose=0)

model.save('new_model.h5')

evaluate_model()
```

**if \_\_name\_\_ == '\_\_main\_\_':**

```
    main()
```

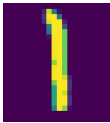


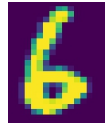

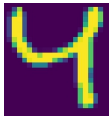

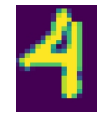
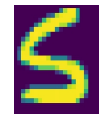
#### 4.2.1 SUMMARY OF THE MODEL:

Accuracy: 99.280  
Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_2 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_4 (Conv2D)	(None, 11, 11, 64)	18496
conv2d_5 (Conv2D)	(None, 9, 9, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_2 (Dense)	(None, 100)	102500
dense_3 (Dense)	(None, 10)	1010
Total params: 159,254		
Trainable params: 159,254		
Non-trainable params: 0		
None		

#### 4.2.2 TESTING:

##### Test Cases

Test No	Input Image	Expected Output	Actual Output	Status
1		1	7	fail
2		3	3	pass
3		8	6	fail
4		6	0	fail
5		5	5	pass
6		9	4	fail
7		4	4	pass
8		7	1	fail
9		4	4	pass
10		5	6	fail

### 4.3 SVM MODEL:

```
import joblib

from sklearn.svm import SVC

from sklearn.metrics import confusion_matrix, accuracy_score

import gzip

from tensorflow.keras.datasets import mnist
```

```
def load_dataset():
```

```
    (trainX, trainY), (testX, testY) = mnist.load_data()

    # reshape dataset to have a single channel

    nsamples, nx, ny = trainX.shape

    trainX = trainX.reshape((nsamples, nx*ny ))

    nsamples, nx, ny = testX.shape

    testX = testX.reshape((nsamples, nx*ny))

    return trainX, trainY, testX, testY
```

```
def process_pixels(train, test):
```

```
    # convert from integers to floats

    train_norm = train.astype('float32')

    test_norm = test.astype('float32')

    # normalize the pixel range from 0-255 to range 0-1

    train_norm = train_norm / 255.0
```



```

test_norm = test_norm / 255.0

return train_norm, test_norm


def main():

    print('loading dataset...')

    trainX, trainY, testX, testY = load_dataset()

    trainX, testX = process_pixels(trainX, testX)


    print('training model...')

    classifier = SVC()

    classifier.fit(trainX, trainY)


    print('Evaluating model... ')

    accuracy = classifier.score(testX, testY)


    print('Accuracy: ', accuracy*100)

    joblib.dump(classifier, 'svm_model.gz', compress=('gzip', 3))

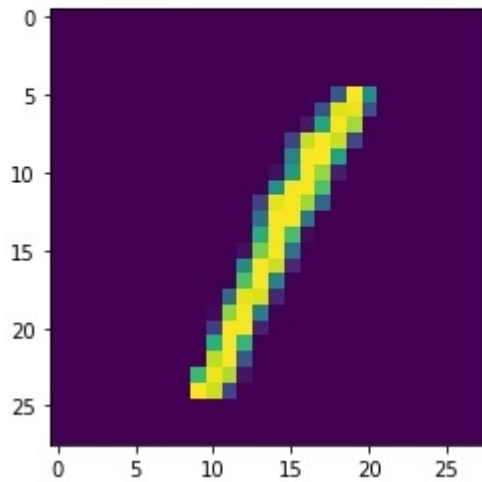
    print('Model saved')

main()

```

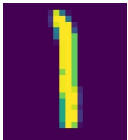
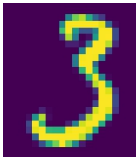
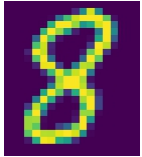
#### 4.3.1 SUMMARY OF THE MODEL:

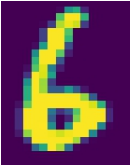
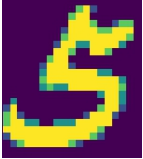
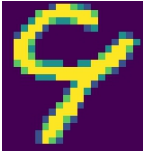
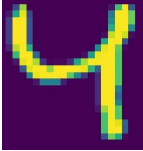
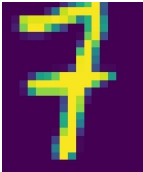
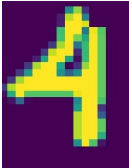
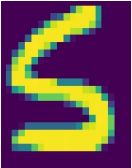
```
Accuracy: 0.9792
Model saved
loading dataset...
loading model...
actual    : 1
predicted : 1
```



#### 4.3.2 TESTING:

##### Test Cases

Test No	Input Image	Expected Output	Actual Output	Status
1		1	1	pass
2		3	3	pass
3		8	8d	pass

4		6	6	pass
5		5	5	pass
6		9	9	pass
7		4	4	pass
8		7	7	pass
9		4	9	fail
10		5	5	pass

## Chapter 5

### Conclusion & Future Work

In this project, we built a model for which a handwritten image of a digit is given as the input. The model then produces the corresponding digit as the result for the handwritten digit given as the input. This model recognises the digits with an accuracy of 99.28%. This project is our first step that we took in the world of Data science and Artificial Intelligence. The accuracy of the model rockets up if more number of layers are added. Through this project, single digit numerals can be identified easily but characters and other special characters can't be recognised. If we build a model that can recognise the characters and the special characters, mathematical operations can be performed, medical transcripts can be scanned and transformed to text from the image, vehicle number plate recognition can be performed. Having this in mind, in our future project, we are planning to develop a real-time model that recognises characters, digits and special characters so that it will help in education, health and business sectors.

# Appendices

## Appendix 1

### **Convolutional Neural Network(CNN):**

Convolutional Neural Network is a subset of Deep learning, which is commonly used for Image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It takes the image as an input in the form of small chunks rather than taking one pixel at a time, so the network can detect irregular patterns in the image more efficiently. CNN contains 3 layers namely, an input layer, an output layer, and multiple hidden layers.

Hidden layers include three types of layers:

1. Convolutional layer - This layer uses a filter (also called as kernel) which is an array of weights to extract features from the input image. One layer can have many filters.
2. Pooling layer - This layer reduces the dimensions of the data coming from the Convolutional layer which in return reduces the computations, number of parameters, reduces overfitting and therefore making the entire process much faster. Typically, the pooling layer is inserted between convolutional layers. It discards the activation of previous layers and hence forcing the next convolutional layers to learn from a limited variety of data
3. Fully connected layer - Output of previous layers is flattened and sent into this layer, which then uses that to classify the image into label.

## Appendix 2

### **Support Vector Machine (SVM) :**

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression, but SVMs are more commonly used in classification problems . The idea behind SVM is to find a hyperplane that divides a dataset into two classes.

The hyperplane must be a optimal line so that the distance from closest node to the Hyperplane, (margin), is maximum.

## REFERENCES

- [1] Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The state of the art in online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 12(8), 787-808.
- [2] Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, 22(3), 418-435.
- [3] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5), 855-868.
- [4] Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1), 63-84.
- [5] Hu, J., Brown, M. K., & Turin, W. (1996). HMM based online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10), 1039-1045.
- [6] Marti, U. V., & Bunke, H. (2002). The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1), 39-46.
- [7] Jaeger, S., Manke, S., Reichert, J., & Waibel, A. (2001). Online handwriting recognition: the NPen++ recognizer. *International Journal on Document Analysis and Recognition*, 3(3), 169-180.
- [8] Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014, September). Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition* (pp. 285-290). IEEE.
- [9] Bunke, H. (2003, August). Recognition of cursive Roman handwriting: past, present and future. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (pp. 448-459). IEEE.
- [10] Plötz, T., & Fink, G. A. (2009). Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(4), 269.

- [11] Starner, T., Makhoul, J., Schwartz, R., & Chou, G. (1994, April). On-line cursive handwriting recognition using speech recognition methods. In Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing (pp. V-125). IEEE.
- [12] Marti, U. V., & Bunke, H. (1999, September). A full English sentence database for off-line handwriting recognition. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318) (pp. 705-708). IEEE.
- [13] Adankon, M. M., & Cheriet, M. (2009). Model selection for the LS-SVM. Application to handwriting recognition. *Pattern Recognition*, 42(12), 3264-3270.
- [14] Aparna, K. H., Subramanian, V., Kasirajan, M., Prakash, G. V., Chakravarthy, V. S., & Madhvanath, S. (2004, October). Online handwriting recognition for Tamil. In Ninth international workshop on frontiers in handwriting recognition (pp. 438-443). IEEE.
- [15] Keysers, D., Deselaers, T., Rowley, H. A., Wang, L. L., & Carbune, V. (2016). Multi-language online handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1180-1194.
- [16] El-Hajj, R., Likforman-Sulem, L., & Mokbel, C. (2005, August). Arabic handwriting recognition using baseline dependant features and hidden markov modeling. In Eighth International Conference on Document Analysis and Recognition (ICDAR'05) (pp. 893-897). IEEE.
- [17] Kozielski, M., Doetsch, P., & Ney, H. (2013, August). Improvements in rwth's system for off-line handwriting recognition. In 2013 12th International Conference on Document Analysis and Recognition (pp. 935-939). IEEE.
- [18] Amma, C., Georgi, M. & Schultz, T. Airwriting: a wearable handwriting recognition system. *Pers Ubiquit Comput* 18, 191–203 (2014).
- [19] Jianying Hu, Sok Gek Lim, Michael K. Brown, Writer independent on-line handwriting recognition using an HMM approach, *Pattern Recognition*.
- [20] Jäger, S., Liu, C. L., & Nakagawa, M. (2003). The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition. *Document Analysis and Recognition*, 6(2), 75-88.
- [21] Milgram, J., Cheriet, M., & Sabourin, R. (2006, October). "One against one" or "one against all": Which one is better for handwriting recognition with SVMs?. In tenth international

workshop on Frontiers in handwriting recognition. Suvisoft.

[22] Liwicki, M., Graves, A., Fernández, S., Bunke, H., & Schmidhuber, J. (2007). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007.

[23] Menasri, F., Louradour, J., Bianne-Bernard, A. L., & Kermorvant, C. (2012, January). The A2iA French handwriting recognition system at the Rimes-ICDAR2011 competition. In Document Recognition and Retrieval XIX (Vol. 8297, p. 82970Y). International Society for Optics and Photonics.

[24] Baldominos, A., Saez, Y., & Isasi, P. (2018). Evolutionary convolutional neural networks: An application to handwriting recognition. *Neurocomputing*, 283, 38-52.

[25] Oh, I. S., Lee, J. S., & Suen, C. Y. (1999). Analysis of class separation and combination of class- dependent features for handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10), 1089-1094.

[26] Kulik, S. D. (2015). NEURAL NETWORK MODEL OF ARTIFICIAL INTELLIGENCE FOR HANDWRITING RECOGNITION. *Journal of Theoretical & Applied Information Technology*, 73(2).

[27] Yin, F., Wang, Q. F., Zhang, X. Y., & Liu, C. L. (2013, August). ICDAR 2013 Chinese handwriting recognition competition. In 2013 12th international conference on document analysis and recognition (pp. 1464-1470). IEEE.

[28] Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C. V. (2018, August). Improving cnn-rnn hybrid networks for handwriting recognition. In 2018 16th international conference on frontiers in handwriting recognition (ICFHR) (pp. 80-85). IEEE.

[29] Kala, R., Vazirani, H., Shukla, A., & Tiwari, R. (2010). Offline handwriting recognition using genetic algorithms. *arXiv preprint arXiv:1004.3257*.

[30] Carbune, V., Gonnet, P., Deselaers, T., Rowley, H. A., Daryin, A., Calvo, M., ... & Gervais, P. (2020). Fast multi-language LSTM-based online handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 1-14.