“Handwritten Digit Recognition using CNN”

A project report of Phase – II submitted to
Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal
towards fulfillment of
The degree of Bachelor of Engineering in Computer Engineering

Guided By:
Mr. KP Singh
Associate Professor
Department of Computer Engg.

Dr. Soma Saha Assistant
Professor
Department of Computer Engg.

Submitted By:
0801CS161035 – Kshitij Gour
0801CS161075 – Sanchit Bhat
0801CS161017 – Ayush Malviya
0801CS171046 – Murtaza Ali
0801CS183D03 – Ashish Yadav

DEPARTMENT OF COMPUTER ENGINEERING
SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE
(M.P.) 2020-2021

RECOMMENDATION

The project report of Phase-II entitled **"Handwritten Digit Recognition using CNN" submitted** by: **0801CS161035 – Kshitij Gour, 0801CS161075 - Sanchit Bhat, 0801CS161017 – Ayush Malviya, 0801CS171046 – Murtaza Ali, 0801CS183D03 – Ashish Yadav,** students of B.E. IV year in the session 2020-2021, towards fulfillment of the degree of **Bachelor of Engineering in Computer Engineering** of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal is a satisfactory account of their work.

Mr. K P Singh
Project Guide
Department of Computer Engg.

Dr. Soma Saha
Project Guide
Department of Computer Engg.

Dr. Urjita Thakar
Head of Department
Department of Computer Engg

Dean (Academics)
S.G.S.I.T.S., Indore

CERTIFICATE

The project report of Phase-II entitled **"Handwritten Digit Recognition using CNN"** submitted by: **0801CS161035 - Kshitij Gour, 0801CS161075 - Sanchit Bhat, 0801CS161017- Ayush Malviya, 0801CS171046 - Murtaza Ali, 0801CS183D03 - Ashish Yadav**, students of B.E. IV year in the session 2019-2020, towards fulfillment of the degree of **Bachelor of Engineering in Computer Engineering** of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal is a satisfactory account of their work.

Internal Examiner **External Examiner**

Date:

ACKNOWLEDGEMENT

We express our profound sense of gratitude to our project Guide **Mr. K P Singh** and our project Co-guide **Dr. Soma Saha** who had advised us to do project work entitled "**Handwritten Digital Recognition using CNN**". Their continuous support and motivational ways made us deliver our best.

Having such guidance has always been an amazing experience which is a valuable gift for an engineer to progress in his/her life.

We are also grateful to **Dr. Urjita Thakar**, Head of Department of Computer Engineering and **Dr. Rakesh Saxena**, Director, S.G.S.I.T.S. Indore, for providing us with numerous facilities and academic environment during the study.

We sincerely wish to express our gratitude to all the members of staff of S.G.S.I.T.S. who have extended their cooperation at all times and have contributed in their way in developing the project.
The successful completion of a project is not an individual effort. It is an outcome of the cumulative number of people, each having their importance to the objective. We express love and respect towards our parents and all family members who are our strength in everything we do.

With a blend of gratitude, pleasure, and great satisfaction we convey our indebtedness to all those who have directly or indirectly contributed to the successful completion of our project work.

0801CS161035 - Kshitij Gour
0801CS171054 - Sanchit Bhat
0801CS171046 - Muraza Ali
0801CS161017 - Ayush Malviya
0801CS183D03 - Ashish Yadav

ABSTRACT

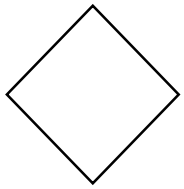
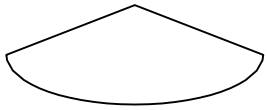
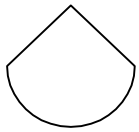
Humans can see and visually sense the world around them by using their eyes and brains. Computer vision works on enabling computers to see and process images in the same way that human vision does. Several algorithms developed in the area of computer vision to recognize images. Handwriting recognition is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touch screen devices etc. Recognition of handwritten and machine characters is an emerging area of research and find extensive applications in banks, offices and industries. The main aim of this project is to design expert system for **"Handwritten Digit Recognition using Convolutional Neural Network".** That can effectively recognize a particular character of type format using the Artificial Neural Network approach. Neural computing is a comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computers are operated in a way which is completely different from the operation of normal computers. Neural computers are trained (not Programmed) so that given a certain starting state (data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.

Keywords: Hand written digit recognition, Convolutional Neural Network (CNN), Deep learning, MNIST dataset, Epochs, Hidden Layers, Stochastic Gradient Descent, Back propagation.

TABLE OF CONTENTS

LIST OF FIGURES

Notations and Symbols/Nomenclatures

Object Notation	
Decision Notation	
Attribute Notation	
Arrow Notation	
Interface Notation	

CHAPTER 1: INTRODUCTION

Handwriting recognition is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touchscreen devices etc. Recognition of handwritten and machine characters is an emerging area of research and finds extensive applications in banks, offices and industries. The main aim of this project is to design expert system for, "HDR using Neural Network
"That can effectively recognize a particular digit of type format using the Convolution Neural Network approach. Neural computing is a comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computers are operated in a way which is completely different from the operation of normal computers. Neural computers are trained (not Programmed) so that given a certain starting state (data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.

The aim of this project is to implement a classification algorithm to recognize handwritten digits (0-9). It has been shown in pattern recognition that no single classifier performs the best for all pattern classification problems consistently.

Recognition is identifying or distinguishing a thing or an individual from the past experiences or learning. Similarly, Digit Recognition is nothing but recognizing or identifying the digits in any document. Digit recognition framework is simply the working of a machine to prepare itself or interpret the digits. Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc.

Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits. Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any
Document. Deep Learning model scan accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

Handwriting recognition of characters has been around since the 1980s. The task of handwritten digit recognition, using a classifier, has extra ordinary significance and use such as – online digit recognition on PC tablets, recognize zip codes on mail, processing bank check amounts, numeric sections in structures filled up by hand and so on. There are diverse challenges faced while attempting to solve this problem. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins. The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0-9).

1.1 Objective

The main objective of this project is to make available below functionalities on the mobile device:

1. Use neural signs in the literature domain.

2. Reduce man-power to convert old literature into digitized form manually
3. The proposed system served as a guide and worked in character recognition areas.
4. To enrich the digitized library with digits.
5. To identify the shortcomings of the system and improve to widen the scope of its utility.

CHAPTER 2: BACKGROUND

In this digital era, the most important thing is to deal with digital documents, organizations using handwritten documents for storing their information can use handwritten character recognition to convert this information into digital. Handwritten characters are more difficult for recognition due to presence of header line, conjunct characters and dissimilarity in shapes of multiple characters. This paper deals with development of a grid-based method which is a combination of image centroid zone and zone centroid zone of individual character or numerical image. In feature extraction using grid or zone-based approach individual character- actor or numerical image is divided into n-equal sized grids or zones then average distance of all pixels with respect to image centroid or grid centroid is computed. In combination of image centroid and zone centroid approach it computes average distance of all pixels present in each grid with respect to image centroid as well as zone centroid which gives feature vector of size 2xn features. This feature vector is presented to feed the forward neural network for recognition. Complete process of digit or character recognition works in stages as document pre-processing, segmentation, feature extraction using grid-based approach followed by recognition using feed forward neural network. The convolutional neural network is mainly used for the image classification and image processing. This neural network can be extracted the features from the images. The input image pixels are the input of the conventional layers. The CNN can filter the feature from the image and transfer it to the next layer. The output layer is giving the results. The CNN gives the higher accuracy for the image classification.

2.1 Circumstances Leading to the project

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 application is useful for recognizing the digits from 0 to 9 given as in the input image, by drawing on the application interface. Once input image of character is given to the proposed system, then it will recognize input digit which is given in image. Recognition and classification of characters are done by Neural Network. The main aim of this project is to effectively recognize a particular character of type format using the Artificial Neural Network approach.

The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolutional Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual's handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

2.2 Libraries used

- **Keras** : Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.
- **TensorFlow** : TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.
- **TensorFlow Lite**: TensorFlow Lite is a set of tools that enables on-device machine learning by helping developers run their models on mobile, embedded, and IoT devices.

CHAPTER 3: SYSTEM ANALYSIS

3.1 System Requirement Analysis

Requirement analysis results in the specification of software's operational characteristics indicating software's interface with other system elements and establishing constraints that soft-ware must meet. Requirement analysis allows the software engineer (sometime called Analyst or Modeler in this role) to elaborate on basis requirements during earlier requirement engineering task and build models that depict user scenarios, functional activities, problem classes and their relationships, system and class behavior and the flow of data as it is trans-formed. The requirements analysis task is a process of discovery, refinement, modelling and specification. The scope, initially established by us and refined during project planning, is refined in details. Model of the required data, information and control flow and operations behavior are created.

3.1.1 Functional Requirements

The functional requirements for a system describe what the system does.

1. The developed system should recognize handwritten English characters present in the image.
2. System shall show the error message to the user when given input is not in the required format.
3. System must provide the quality of service to the user.
4. System must provide accuracy for character recognition.

3.1.2 Normal Requirements:

These are the requirements clearly stated by the customer hence requirements must be present for customer satisfaction.
N1: Application should have graphical user interface.
N2: Input of characters with various font sizes and styles should be recognized.
N3: Database should identify computer based English characters by comparison.
N4: Application should be able for matching the stored patterns on input handwritten character.
N5: Minimum 10*50 (characters * patterns) should be available for each character.

3.1.3 Expected Requirements:

These requirements are implicit types of requirements. These requirements are not clearly stated by the customer but even though the customer expects them.
Exp1: Instead of only one character application should take aset of characters or text. Exp2 Application should be user friendly and also easy to install.
Exp3: By using Neural Network bringing more accuracy in the character recognition process. Exp4: Application also recognizes English numerals.
Exp5: Minimum 26*50 (character * patterns) should be available for each character.

3.1.4 Excited Requirements:

These requirements are neither stated by the customer nor expected. But to make the customer satisfied, developers may include some unexpected requirements.
Exc1: Application to interpret all the English alphabets through NN training process.
Exc2: Using this application Continuous handwritten characters need to be recognized.
Exc3: Alphanumeric characters with special symbols should be recognized with the proposed system.

Exc4: Development of HDR system for noisy images.

3.1.5 Non-Functional Requirements

As the name suggests these are the requirements that are not directly interacted with specific functions delivered by the system.

Performance: Handwritten characters in the input image will be recognized with an accuracy of about 99.13%.

Functionality: This software will deliver on the functional requirements.

Availability: This system will retrieve the handwritten text regions only if the image contains written text in it.

Flexibility: It provides the users to load the image easily.

Learnability: The software is very easy to use and reduces the learning work.

3.2 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Dimensions of Software Feasibility are as follows:

Technology:
Is the project technically feasible? Is it within the state of art? Can defects be reduced to a level matching an application's need?

Finance: Is it financially feasible? Can development be complete data cost the software organization and its client or market can afford?

Time:
Will the project's time to market beat competition?

Resources:
Does the organization have resources needed to succeed?
Two key considerations involved in the feasibility analysis are:

1. Financial Feasibility.
2. Techinal Feasibility.

3.2.1 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Technical feasibility assessment can be done through following ways:
1. NP-Complete
2. NP-Hard
3. Satisfiability

1. NP-Complete:
P Class: Class of all deterministic polynomial language problems.
NP Class: Class of all non-deterministic polynomial language problems. NP Complete Problems are always solved within a given time and space.

2. NP-Hard:
These are problems for which there are no efficient solutions. Generally, the complexity of these problems are more than P, NP, NP-Complete. These may include higher multiplicative constants, exponent's terms or high order polynomials.

3. SAT (Satisfiability):
Boolean formula is satisfiable if there exists at least one way of assigning value to its variable so as to make it true and we denote it by using SAT. The problem of deciding whether a given formula is satisfiable or not.

3.2.2 Financial Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchase.

3.3 System Requirements

Hardware Requirements

- Intel i3 Processor
- 128 MB RAM
- 10 GB Hard Disk.

Software Requirements

- Windows 7/8/8.1
- Language: Python, Tensor Flow.
- Eclipse

3.4 Validation of requirements

The project **"HDR using Convolutional Neural Network"** will be recognized as a successful implementation if it provides all the required images on the basis of suitable input with minimum time. The requirement specification define should be validated such that the successful implementation of the product can be recognized. Hence validation specifies classes of tests to be performed to validate function, performance and the constraints. With respect to the system under consideration the following issues are to be validated to ensure consistency of system.
V1: The Neural Network is known to be capable of providing a good recognition rate at the present as compared to other methods.

V2: Handwritten Character Recognition system gives much better results in terms of performance and accuracy in comparison with existing usual approaches due to the application of artificial way character recognition and Neural Network in detection of characters.
V3: Handwritten Character Recognition technology provides image definition, image pre-processing and image segmentation and recognition capabilities and still maintains a high level of accuracy in the field of image processing.

3.5 Information Flow Diagram

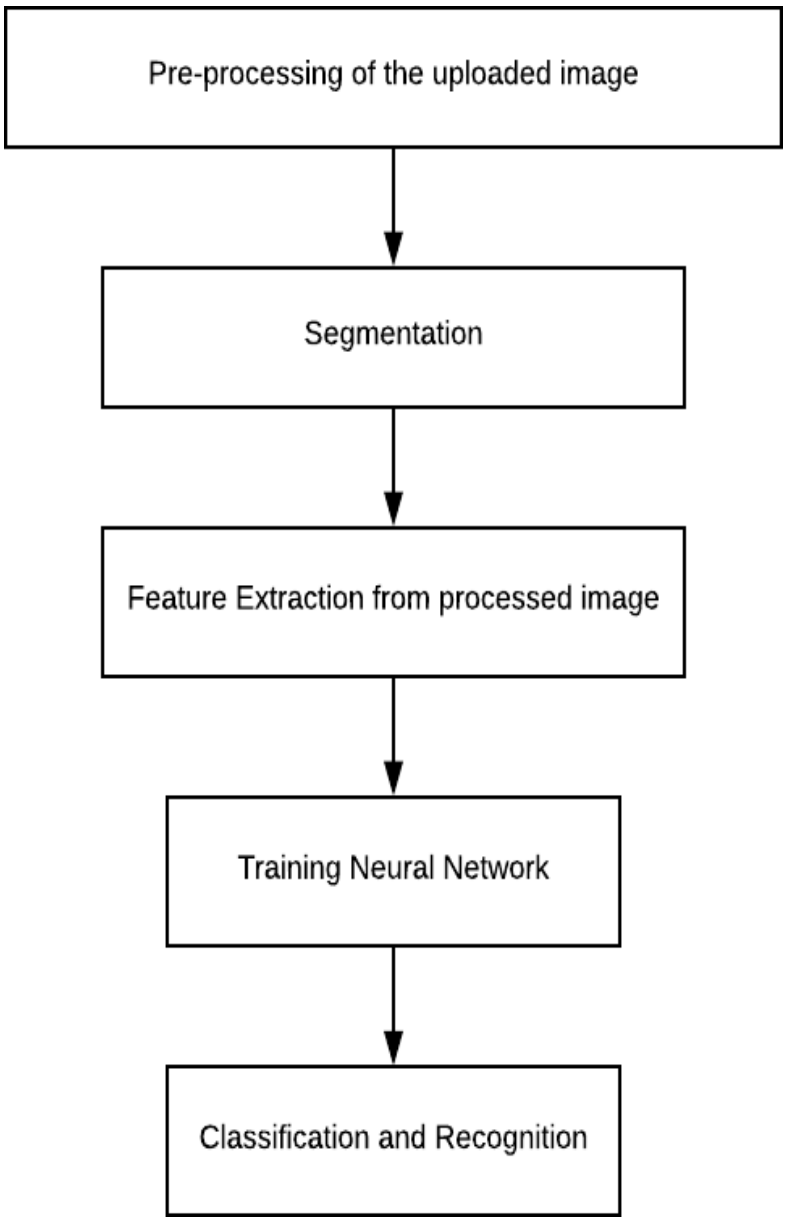


Figure 3.5: Information Flow Diagram

CHAPTER 4: DESIGN

This chapter specifies various diagrams related to the project. It includes a Process model, use case diagram, Sequence and collaboration diagrams, data flow diagrams (level 0, level 1 and level 2).

4.1 Process Model

Process Models are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. Since the process model is at the type level, is an instantiation of it. The same process model is used repeatedly for the development of many applications and thus, has many instantiations. One possible use of a process model is to prescribe how things must/should/could be done in contrast to the process itself which is really what happens. A process model is roughly anticipation of what the process will look like. What the process shall be determined during actual system development.

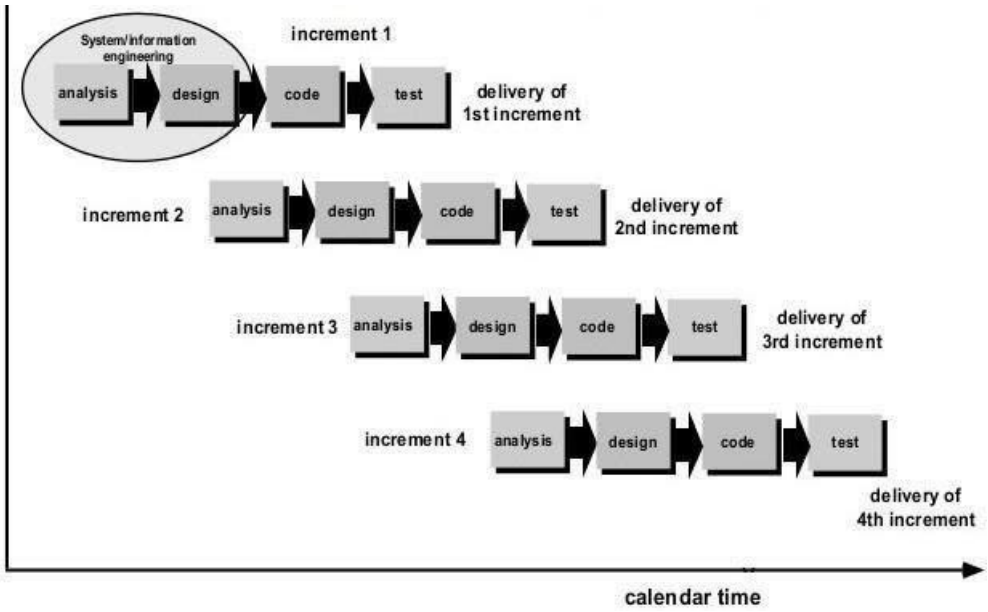


Figure 4.1: Process Model diagram

The goal of a process model is to be:

• Descriptive

1. Track what actually happens during a process.
2. Take the point of view of an external observer who looks at the way a process has been performed and determines the improvements that must be made to make it perform more effectively or efficiently.

• Prescriptive

1. Define the desired processes and how they should/could/might be performed.
2. Establish rules, guidelines, and behavior patterns which, if followed, would lead to the desired process performance. They can range from strict enforcement to flexible guidance.

• Explanatory

1. Provide explanations about the rationale of processes.
2. Explore and evaluate the several possible courses of action based on rational arguments.

3. Establish an explicit link between processes and the requirements that the model needs to fulfil.
4. Pre-defines points at which data can be extracted for reporting purposes.

4.1.1 Incremental Model

Incremental models are used as the process model in our system. To save actual problems in an industry setting, Software Engineering must incorporate a development strategy that encompasses the process, method and the tool layers; this strategy is often referred to as process model. A process model for Software Engineering is chosen based on the nature of the Project and its application. For our project, we have selected an Incremental Model.

1. Using these models, a limited set of customer requirements are implemented quickly and are delivered to customers.
2. Modified and expanded requirements are implemented step by step.
3. It combines elements of linear Sequential Model with the iterative Philosophy of prototyping.
4. Each linear sequence produces a deliverable Increment of the Software.
5. Each linear Sequence is divide into 4 parts: -
    - Analysis
    - Design
    - Code
    - Testing

1. **Analysis**
   It includes understanding of information domain, required functions, behavior, performance and interface. Requirements for the system and software are documented and reviewed with customers.

2. **Design**
   It is multiple processes that include four attributes of program data structure, software architecture, interface representation and procedural detail.

3. **Coding**
   Translation of design to machine code is done by this step.
4. **Testing**
   It focuses on Logical internals of Software and ensures that all statements are correct to uncover all hidden errors. For an incremental model, the first Increment is developed as a core model, which is used by the customer. Then as things are added after the first delivery, the product gets better and better.
    1. Generates working Software quickly and early during the software life cycle.
    2. More Flexible-less costly to change Scope and Requirements.
    3. Easier to test and debug during a smaller iteration.
    4. Customers can respond to each build.

Why do we use Incremental Models?
The main aim of using the model is the reason that we have to add more features in the existing modules to increase project reliability and usability. Using this model, we can adapt to the changing requirements of the customer which helps in developing the project in a relatively small amount of time. The next increment implements customer suggestions plus some additional requirements in the previous increment. The process is repeated until the project is completed.

Characteristics of Incremental Model

1. Using these models a limited set of customers' requirements are implemented quickly and are delivered to customers, then modified and expanded requirements are implemented step by step.
2. Each increment produces the product which is submitted to the customer and suggests some change and increment implements that changes with some extra requirements to previous.
3. Incremental models do not facilitate the development of projects in one go. This is useful for developing modules and then testing them which helps us to modularize the entire project for better handling.

So finally, it is easier to develop projects in increments. We can develop a working prototype 1st with just basic functions and then build upon this prototype in later increments. This will help to reduce the complexity of the system by dividing the entire system indifferent levels of priority. We have discarded the other process models based on following points:

4.2 USE Case Diagram

The use case diagram of the system is shown in this section.

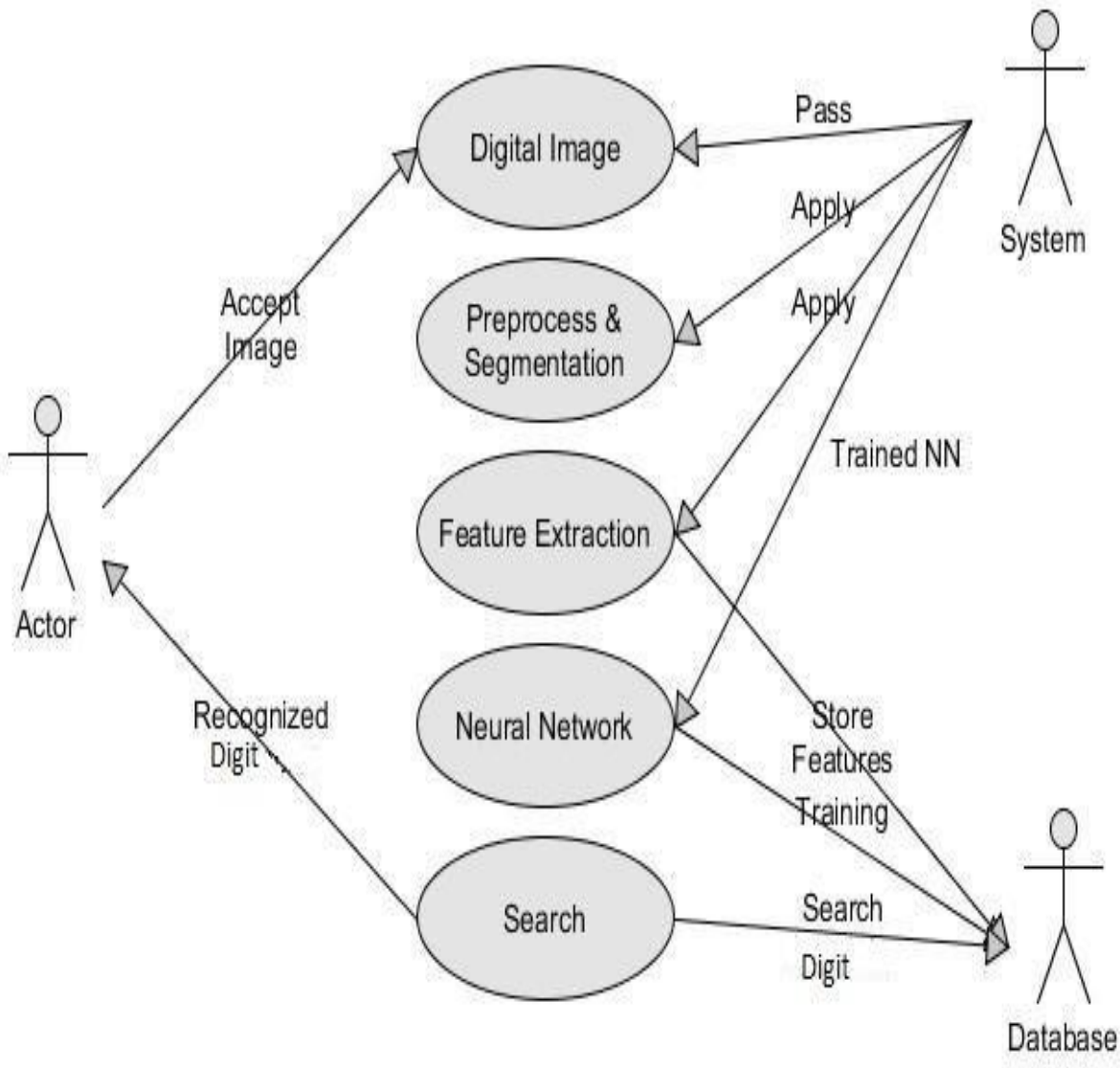


Fig 4.2: USE Case Diagram

4.3 Activity Diagram

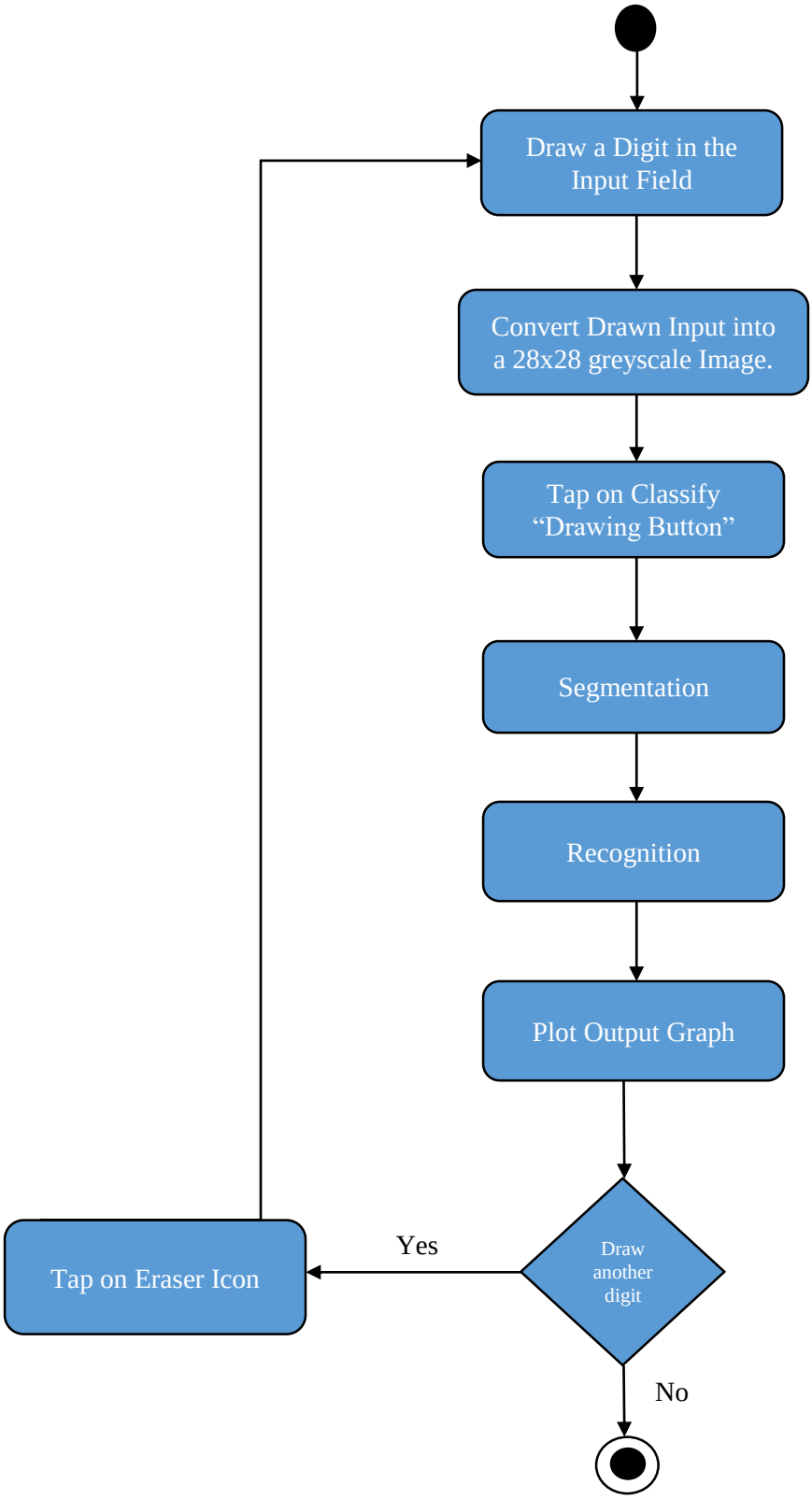


Figure 4.3: Activity Diagram

4.4 Sequence Diagram

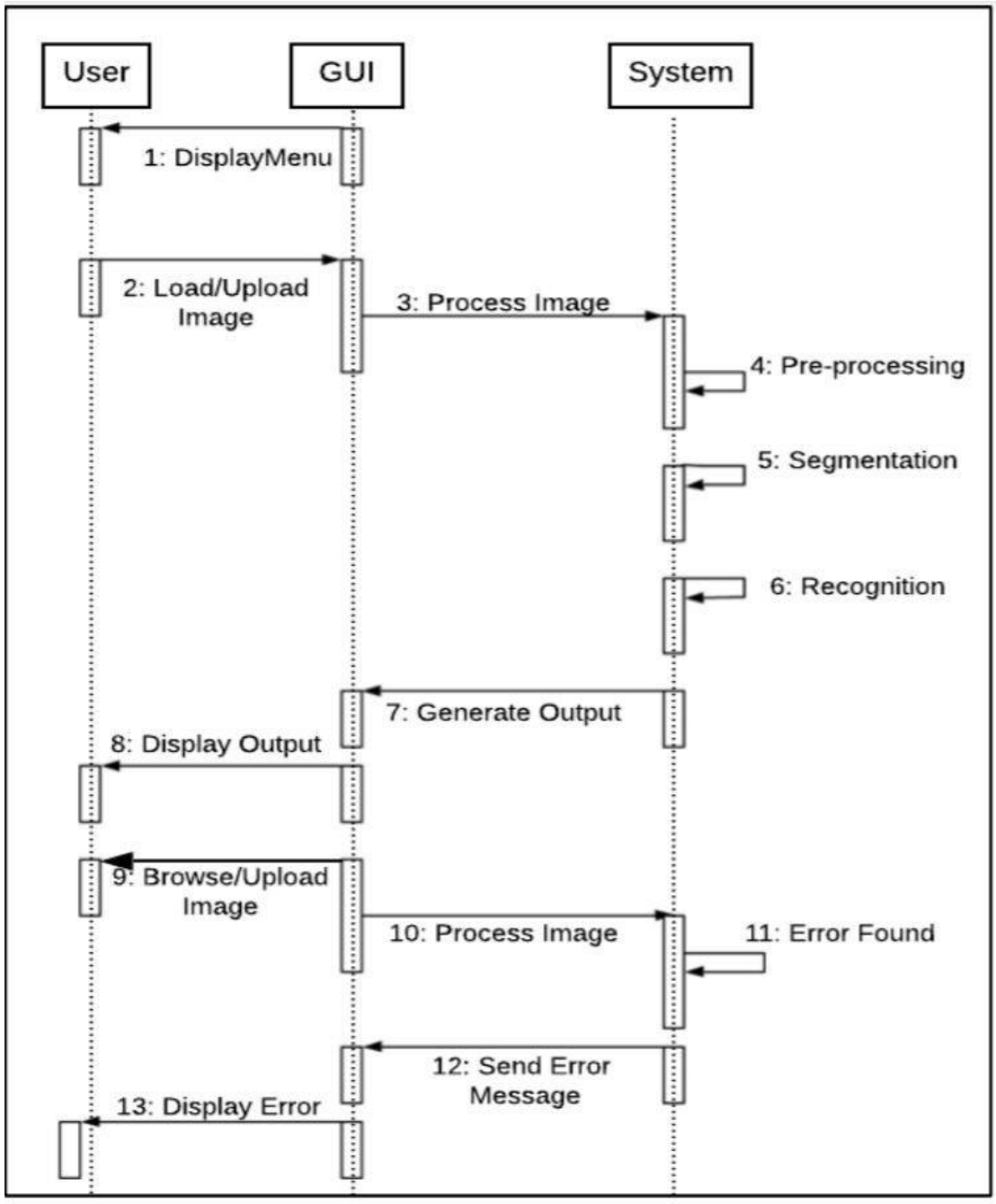


Fig 4.4: Sequence Diagram

4.5 Deployment Diagram

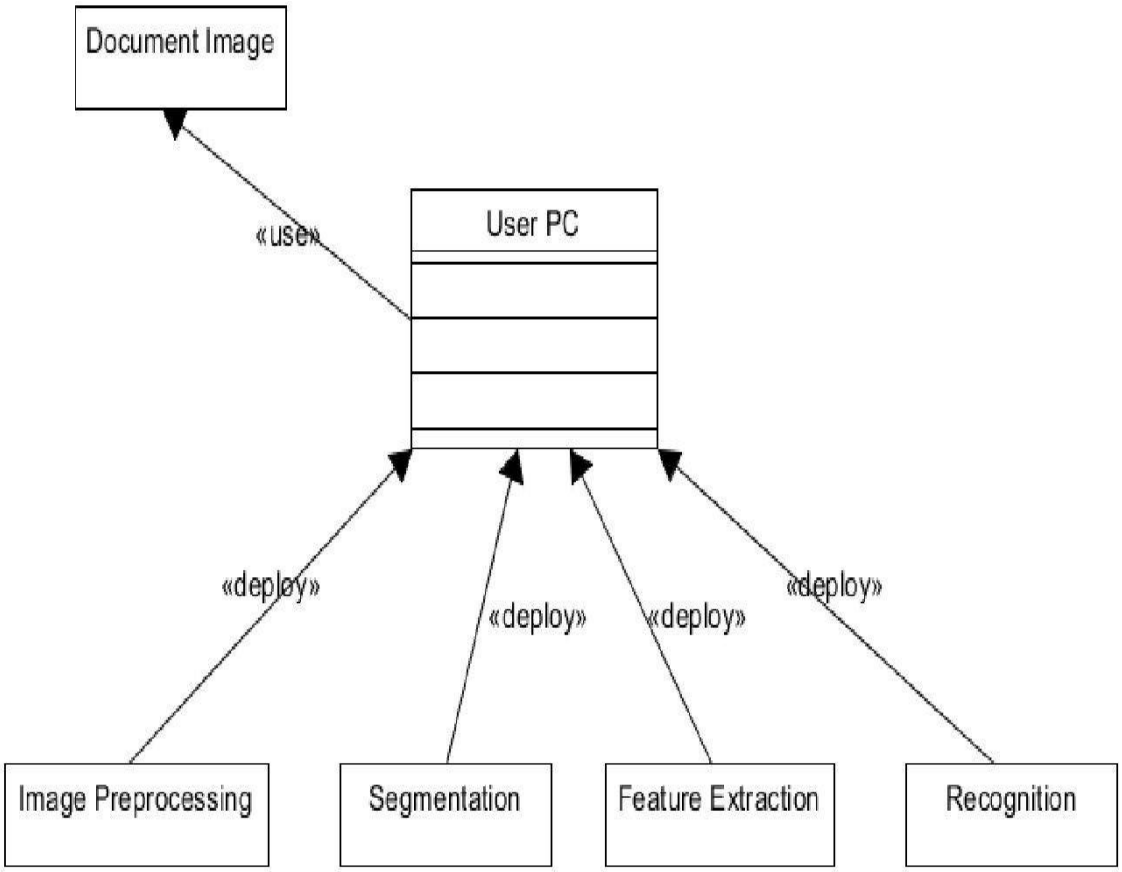


Figure 4.5: Deployment Diagram

4.6 Data Flow Diagram

In this section, dataflow diagram of the system is given. The dataflow diagram is presented in three levels i.e., Level 0, Level 1 and Level2.

- **Data Flow Diagram (Level 0):**

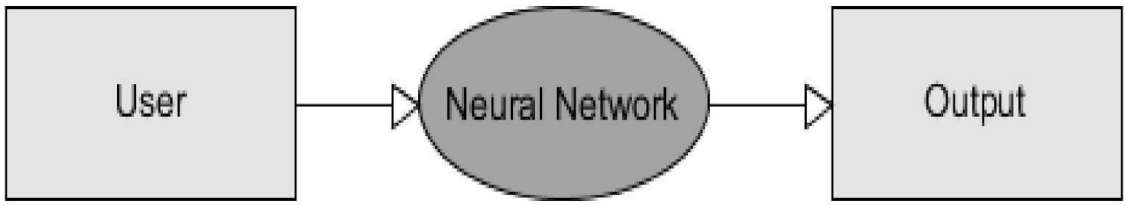


Figure 4.6.1: Data Flow Diagram Level 0

- **Data Flow Diagram (Level 1):** The Level 1 data flow diagram shows the flow of data between different modules of the system and the user.

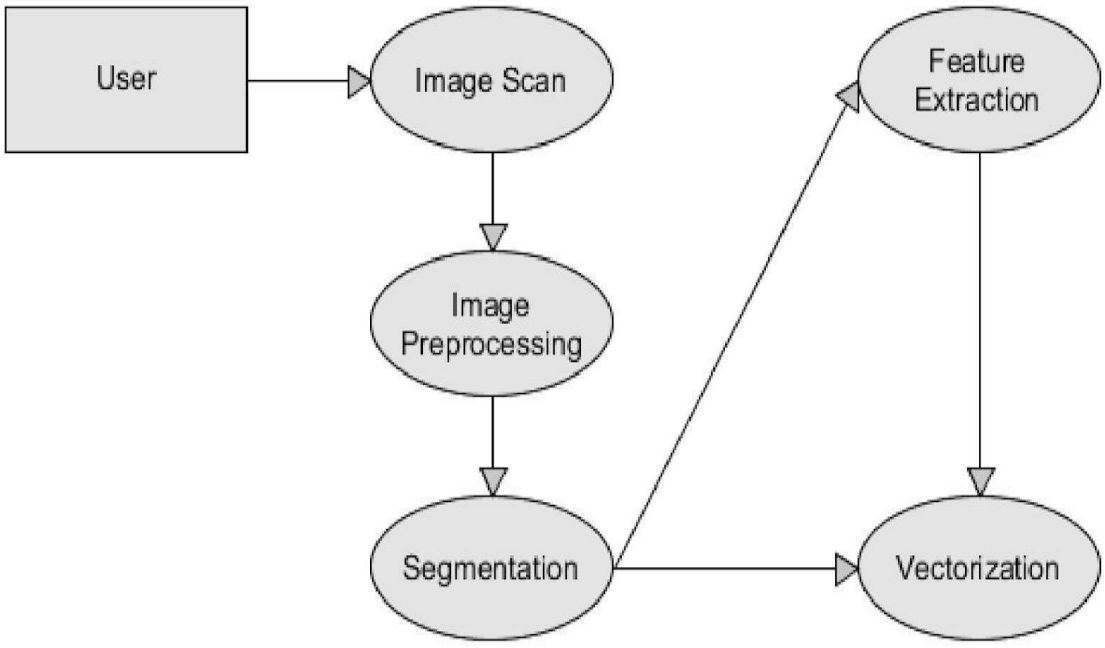


Figure 4.6.2: Data Flow Diagram Level 1

- **Data Flow Diagram of the System Model**

  The various processes performed by different modules is shown in Level 2 Data flow diagram

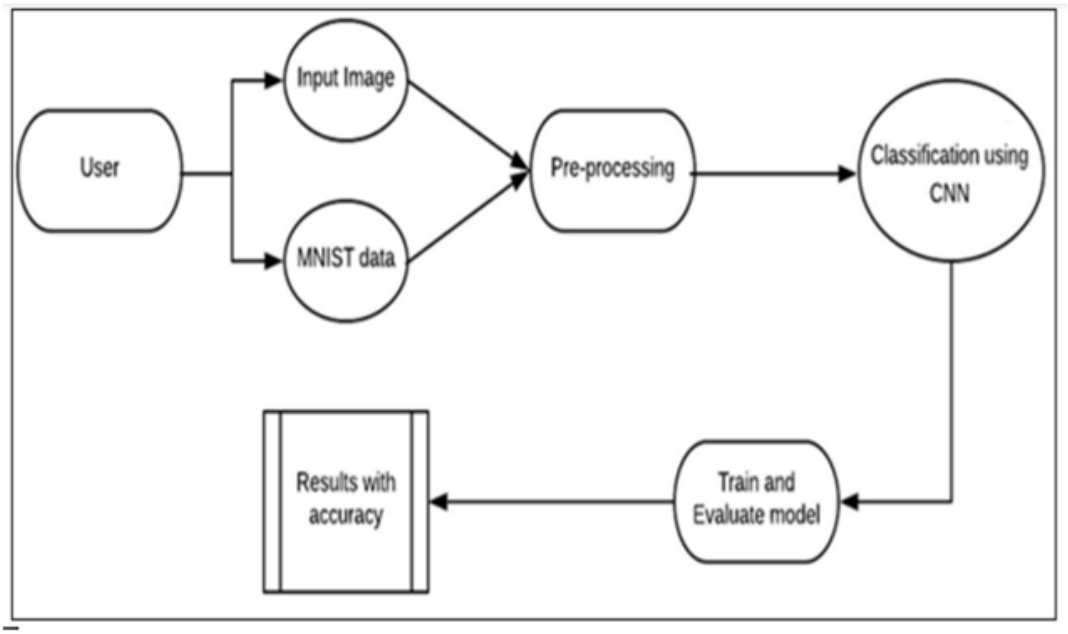


Figure 4.6.3: Data Flow Diagram of the System

4.7 Architecture

The reason behind this document is to look into the design possibilities of the proposed system, such as architecture design, block diagram, sequence diagram, dataflow diagram and user interface design of the system in order to define the steps such as pre-processing, feature extraction, segmentation, classification and recognition of digits.

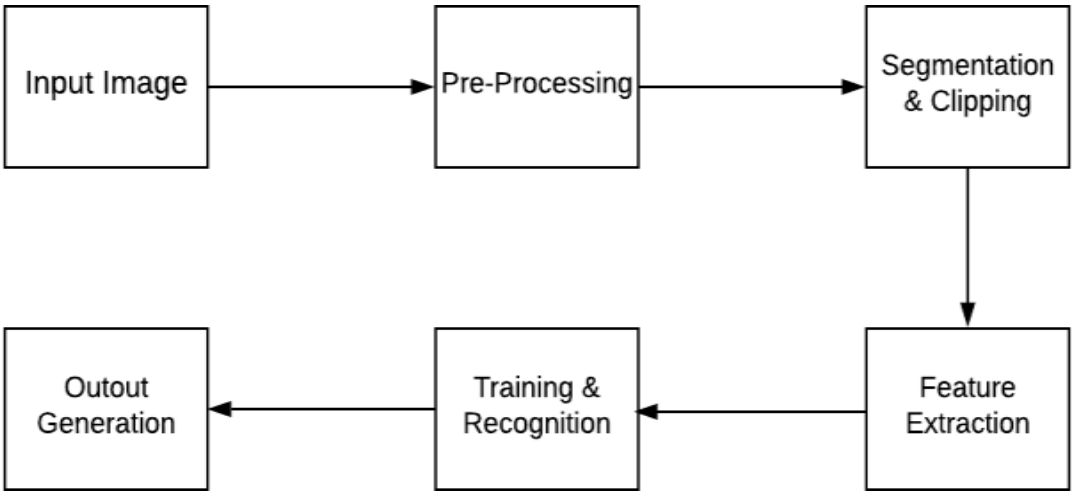


Figure 4.7: Architecture of the System

The above Figure illustrates the architecture diagram of the proposed system. The proposed model contains the four stages in order to classify and detect the digits:

A. Pre-processing
B. Segmentation
C. Feature Extraction
D. Classification and Recognition

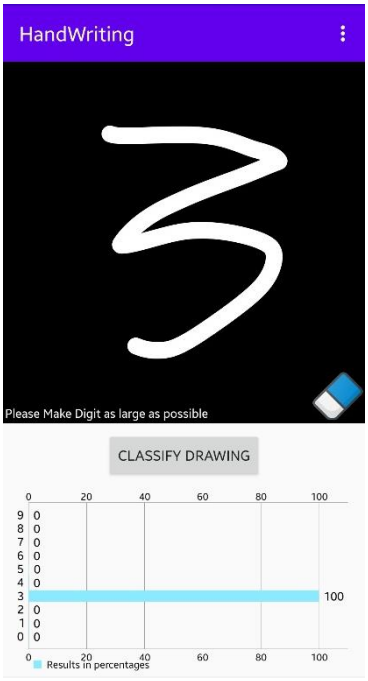
CHAPTER 5: TESTING

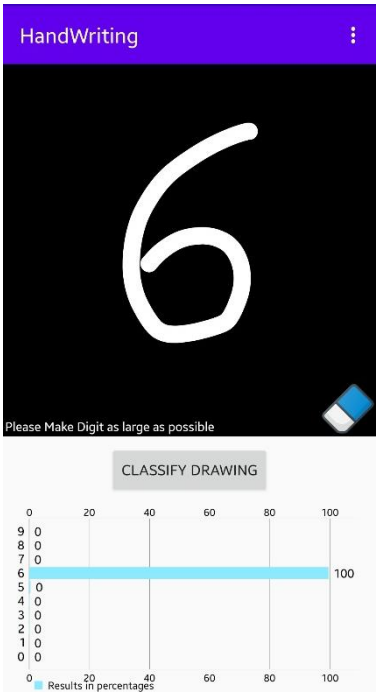
The MNIST dataset is divided into two parts:

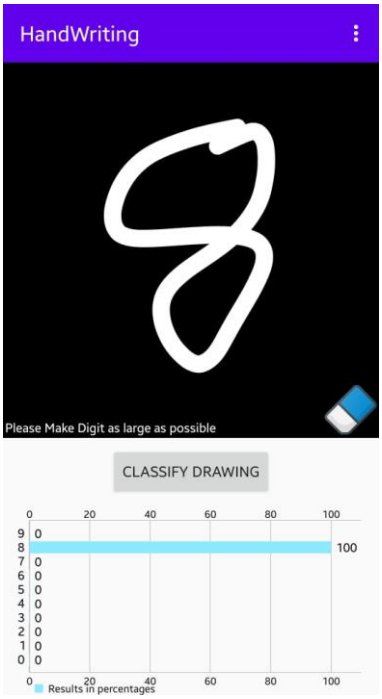
1. Training data (60,000)
2. Testing data (10,000)

The model is tested using the testing dataset, and the result after evaluation is 99.07 % on testing data.

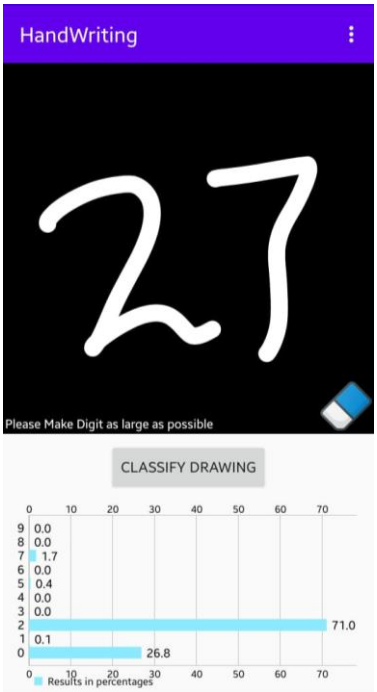
- The screenshots of the results are:



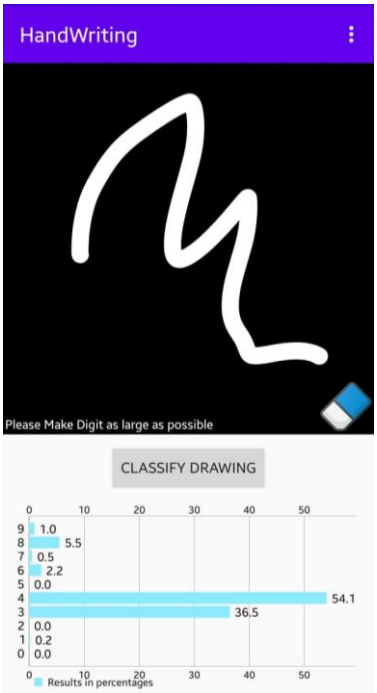


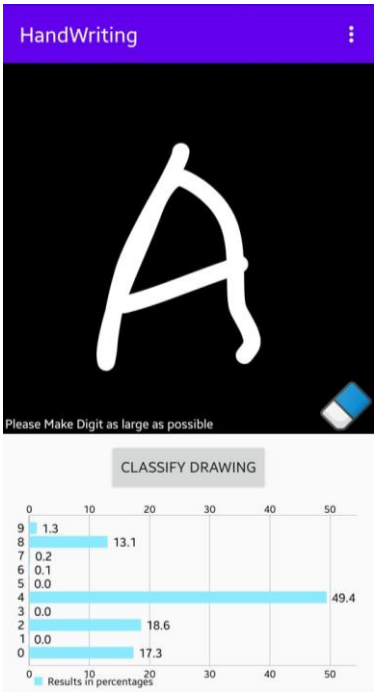


- Cases where system can be improved:



Double Digits





Unrecognized Characters

CHAPTER 6: LIMITATIONS AND IMPROVEMENTS

6.1 Limitations

1. Currently the application is only available for android devices.
2. The image of digit should be clear, otherwise the results may vary.
3. The system can recognize only digits.
4. If something else is drawn other than digits, it predicts the possibility of what digit it can be, results are unclear in this case.
5. It works on only the predefined training dataset, therefore in case of any new drawing patterns result accuracy may be low.
6. It works only for single digits.

6.2 Improvements

1. The dataset can be broadened in case of occurrence of new patterns for improved accuracy,
2. The input images can be added to the training dataset so that the model keeps learning for better accuracy.
3. The above point can increase the personal experience for users.
4. The model can be trained to recognize unknown patterns which are different from the goals of recognition and prompt the user for the same.
5. The model can be modified to recognize multiple digit numbers, which will increase the scope of application of the system.

CHAPTER 7: SUMMARY AND CONCLUSION

Handwritten Digit Recognition using Deep learning methods has been implemented. This project is deployed using CNN and have been trained and tested on the same data. The data (70000) is divided in two parts i.e., training data (60,000) and testing data (10,000). In order to acquire the comparison between the classifiers. Utilizing these deep learning techniques, a high amount of accuracy can be obtained. Compared to other research methods, this method focuses on which classifier works better by improving the accuracy of classification models. Using Keras as backend and Tensor Flow (Lite) as the software, a CNN model is able to give testing accuracy of 99.07%.

CHAPTER 8: FUTURE SCOPE

It can be extended for character recognition and real-time person's handwriting. Handwritten digit recognition is the first step to the vast field of Artificial Intelligence and Computer Vision. As seen from the results of the experiment, CNN proves to be far better than other classifiers. The results can be made more accurate with more convolution layers and a greater number of hidden neurons. It can completely abolish the need for typing.

Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning.

It can utilized in the real-time traffic systems, parking systems, etc., by widening the scope of recognized characters and the application of this basic system can be broadened to a wide range of areas already stated as the examples. This module can also be used to make machines more automated and increase accessibility in the context of human life.

BIBLIOGRAPHY

How to Develop a CNN for MNIST Handwritten Digit Classification (machinelearningmastery.com)

Digit recognizer using CNN. I build a simple Convolutional Neural….| by Nischal Madiraju | Towards Data Science

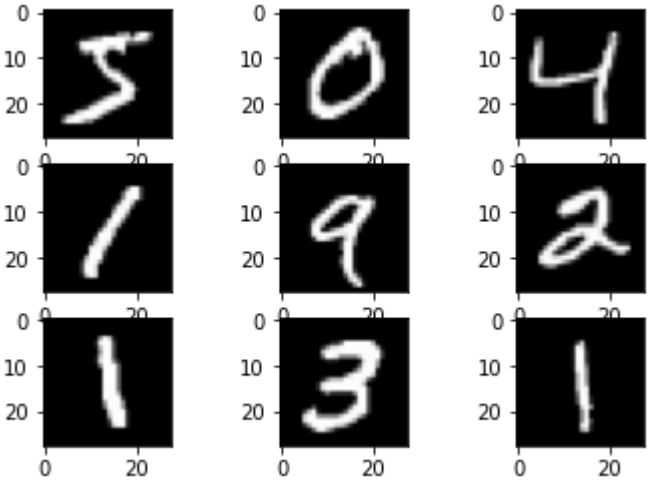
Recognition of Handwritten Digit using Convolutional Neural Network (CNN) (core.ac.uk)

APPENDIX A

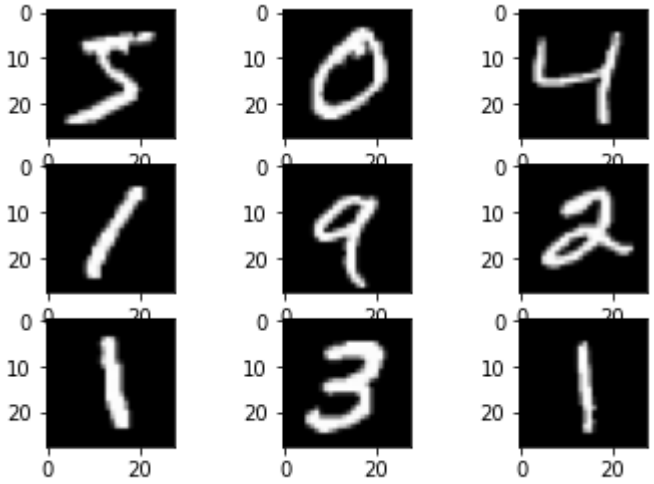
Code for the Model in Python

```
from numpy.random import seed
seed(1)
import tensorflow as tf
tf.random.set_seed(1)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPool
ing2D, GlobalAveragePooling2D, AveragePooling2D
from keras.datasets import mnist
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score

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

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step

x_train.shape
#Output:
(60000, 28, 28)


x_test.shape
#Output:
 (10000, 28, 28)


for i in range(9):
      plt.subplot(330 + 1 + i)
      plt.imshow(x_train[i], cmap=plt.get_cmap('gray'))

plt.show()
#Output:
```



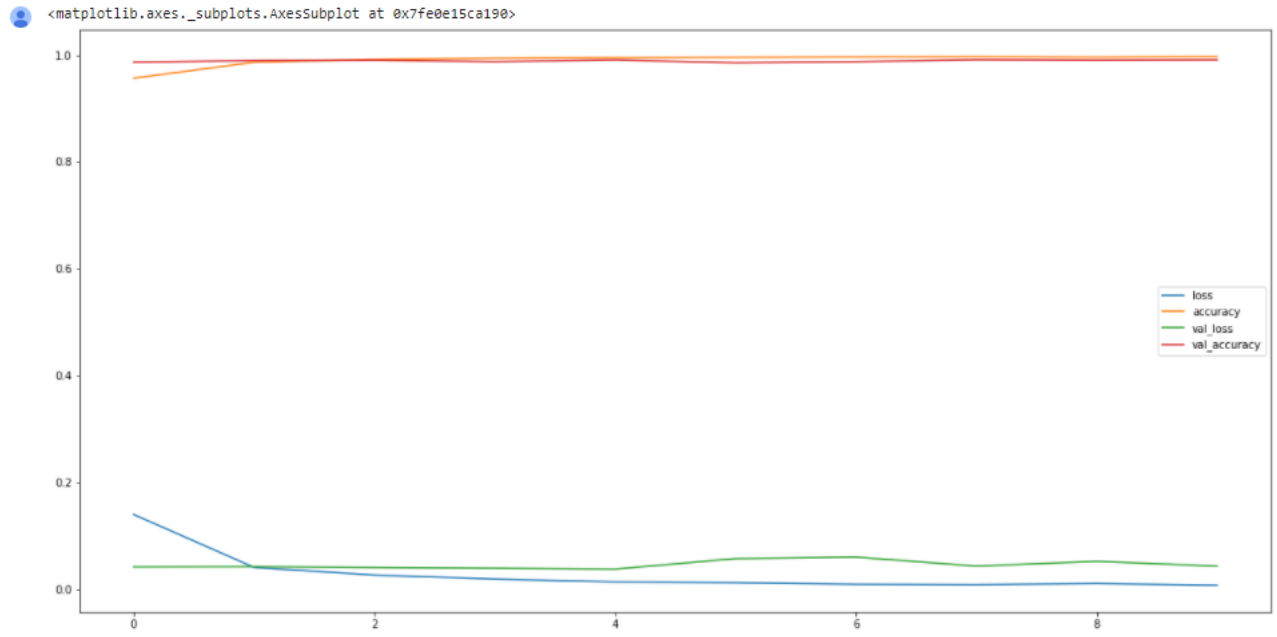
```
input_shape = (28, 28, 1)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
#normalization
x_train /= 255
x_test /= 255

for i in range(9):
      plt.subplot(330 + 1 + i)
      plt.imshow(x_train[i], cmap=plt.get_cmap('gray'))
plt.show()
#Output:
```



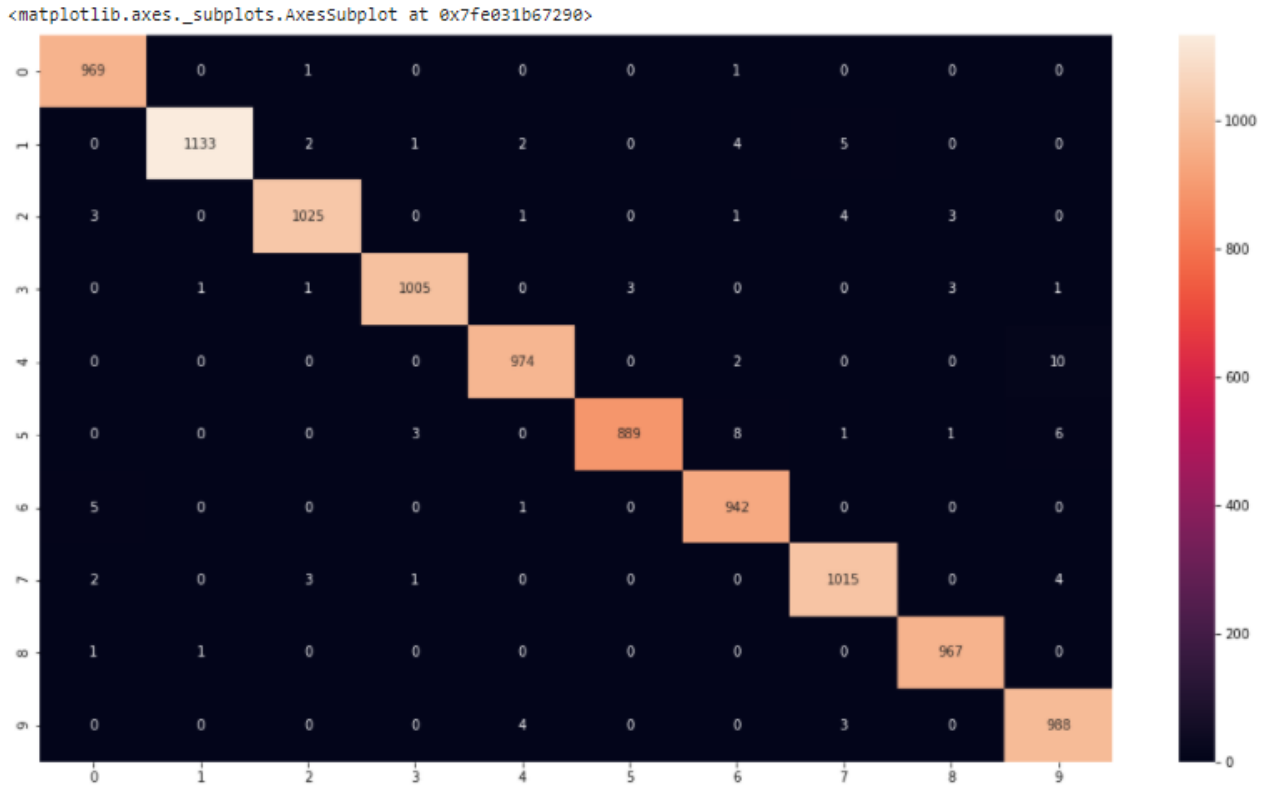
```
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
x_train.shape
#Output:
 (60000, 28, 28, 1)


from tensorflow.keras.utils import to_categorical
y_train = to_categorical(y_train)
```

```
y_train
#Output:
array([[0., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.]], dtype=float32)

def cnn_model_Max():
  model = Sequential()
  model.add(Conv2D(filters= 64, kernel_size= (3,3), strides= (1,1),padding=
'same',   activation= 'relu', input_shape= (28,28,1)))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(Conv2D(filters= 128, kernel_size= (3,3), strides= (1,1),padding=
'same', activation= 'relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(Conv2D(filters= 256, kernel_size= (3,3), strides= (1,1),padding=
'same', activation= 'relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(Flatten())
  model.add(Dense(units= 128, activation= 'relu'))
  model.add(Dense(units= 10, activation= 'softmax'))
  model.compile(optimizer = 'Adam', loss = 'categorical_crossentropy',
metrics= ['accuracy'])
  return model

model = cnn_model_Max()

model.summary()

#Output:
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 28, 28, 64)        640
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 14, 14, 128)       73856
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 7, 7, 256)         295168
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 3, 3, 256)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_4 (Dense)              (None, 128)               295040
_________________________________________________________________
dense_5 (Dense)              (None, 10)                1290
=================================================================
```

```
Total params: 665,994
Trainable params: 665,994
Non-trainable params: 0
_________________________________________________________________

history = model.fit(x=x_train,y=y_train, epochs=10, batch_size= 64,
validation_split= 0.1, shuffle=False)
Epoch 1/10
844/844 [==============================] - 11s 12ms/step - loss: 0.3337 -
accuracy: 0.8977 - val_loss: 0.0420 - val_accuracy: 0.9868
Epoch 2/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0466 -
accuracy: 0.9850 - val_loss: 0.0424 - val_accuracy: 0.9900
Epoch 3/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0290 -
accuracy: 0.9914 - val_loss: 0.0407 - val_accuracy: 0.9907
Epoch 4/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0208 -
accuracy: 0.9938 - val_loss: 0.0393 - val_accuracy: 0.9880
Epoch 5/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0139 -
accuracy: 0.9957 - val_loss: 0.0378 - val_accuracy: 0.9912
Epoch 6/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0116 -
accuracy: 0.9967 - val_loss: 0.0574 - val_accuracy: 0.9857
Epoch 7/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0091 -
accuracy: 0.9969 - val_loss: 0.0603 - val_accuracy: 0.9878
Epoch 8/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0082 -
accuracy: 0.9974 - val_loss: 0.0434 - val_accuracy: 0.9915
Epoch 9/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0111 -
accuracy: 0.9962 - val_loss: 0.0523 - val_accuracy: 0.9907
Epoch 10/10
844/844 [==============================] - 10s 12ms/step - loss: 0.0067 -
accuracy: 0.9974 - val_loss: 0.0433 - val_accuracy: 0.9913

pd.DataFrame(history.history)
pd.DataFrame(history.history).plot(figsize = (20,10))
#Output:
```




```
predict_model = model.predict(x_test)
predict_model = np.argmax(predict_model, axis= 1)
plt.figure(figsize = (18,10))
sns.heatmap(confusion_matrix(predict_model, y_test), annot = True, fmt =
'0.0f')
#Output:
```




```
accuracy_score(predict_model, y_test)

model.save("/content/drive/MyDrive/Digit Pred App")
```

```
converter =
tf.lite.TFLiteConverter.from_saved_model("/content/drive/MyDrive/Digit Pred
App") # path to the SavedModel directory
tflite_model = converter.convert()

with open('/content/drive/MyDrive/Robotronix_Projects/Digit Pred
App/model.tflite', 'wb') as f:
  f.write(tflite_model)
```

APPENDIX B

Main Activity Code for the Android application in Java:

```
package com.example.handwriting;

import android.annotation.SuppressLint;
import android.content.res.AssetFileDescriptor;
import android.graphics.Bitmap;
import android.graphics.Point;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.Display;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import com.github.mikephil.charting.charts.HorizontalBarChart;
import com.github.mikephil.charting.components.Legend;
import com.github.mikephil.charting.components.XAxis;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;
import com.github.mikephil.charting.interfaces.datasets.IBarDataSet;

import org.tensorflow.lite.Interpreter;

import java.io.FileInputStream;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.MappedByteBuffer;
import java.nio.channels.FileChannel;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private PaintView paintView;
    private Button btnClassify;
    private ImageButton btnClear;
    Interpreter tflite;
    private HorizontalBarChart chart;
    ArrayList<BarEntry> values = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        chart = findViewById(R.id.chart);
        chart.setDrawBarShadow(false);
```

```
        chart.setDrawValueAboveBar(true);

        chart.getDescription().setEnabled(false);


        chart.setMaxVisibleValueCount(11);

        XAxis xl = chart.getXAxis();
        xl.setPosition(XAxis.XAxisPosition.BOTTOM);
        xl.setDrawGridLines(false);
        xl.setCenterAxisLabels(false);
        xl.setTextSize(12);
        xl.setLabelCount(10);

        YAxis yl = chart.getAxisLeft();
        yl.setDrawAxisLine(true);
        yl.setDrawGridLines(true);
        yl.setAxisMinimum(0f);

        YAxis yr = chart.getAxisRight();
        yr.setDrawAxisLine(true);
        yr.setDrawGridLines(false);
        yr.setAxisMinimum(0f);

        chart.setFitBars(true);

        Legend l = chart.getLegend();
        l.setVerticalAlignment(Legend.LegendVerticalAlignment.BOTTOM);
        l.setHorizontalAlignment(Legend.LegendHorizontalAlignment.LEFT);
        l.setOrientation(Legend.LegendOrientation.HORIZONTAL);
        l.setDrawInside(false);
        l.setFormSize(8f);
        l.setXEntrySpace(4f);

        try {
            tflite = new Interpreter(loadModelFile());
        } catch (Exception ex) {
            ex.printStackTrace();
        }

        btnClassify = findViewById(R.id.btnClassify);
        btnClassify.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    paintView.setDrawingCacheEnabled(true);
                    paintView.buildDrawingCache();
                    Bitmap bitmap =
paintView.getDrawingCache().copy(Bitmap.Config.RGB_565, false);
                    paintView.setDrawingCacheEnabled(false);
                    Bitmap resized = Bitmap.createScaledBitmap(bitmap, 28,
28, true);
                    ByteBuffer buff =
bitmapToModelsMatchingByteBuffer(resized);
                    runInferenceOnFloatModel(buff);
                } catch (Exception ex) {
                    ex.printStackTrace();
                }
            }
        });
```

```
        btnClear = findViewById(R.id.btnClear);
        btnClear.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                paintView.clear();
            }
        });

        Display display = getWindowManager().getDefaultDisplay();
        Point size = new Point();
        display.getSize(size);
        int width = size.x;

        paintView = findViewById(R.id.paintView);
        paintView.setLayoutParams(new ConstraintLayout.LayoutParams(width,
width));
        DisplayMetrics metrics = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(metrics);
        paintView.init(metrics);

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.main, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.normal:
                paintView.normal();
                return true;
            case R.id.emboss:
                paintView.emboss();
                return true;
            case R.id.blur:
                paintView.blur();
                return true;
        }

        return super.onOptionsItemSelected(item);
    }

    private MappedByteBuffer loadModelFile() throws IOException {
        AssetFileDescriptor fileDescriptor =
this.getAssets().openFd("mnist_model.tflite");
        FileInputStream inputStream = new
FileInputStream(fileDescriptor.getFileDescriptor());
        FileChannel fileChannel = inputStream.getChannel();
        long startOffset = fileDescriptor.getStartOffset();
        long declaredLength = fileDescriptor.getDeclaredLength();
        return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset,
declaredLength);
    }

    @SuppressLint("DefaultLocale")
```

```
    private void runInferenceOnFloatModel(ByteBuffer byteBufferToClassify) {
        float[][] result = new float[1][10];
        tflite.run(byteBufferToClassify, result);

        values = new ArrayList<>();
        for (int i = 0; i < result[0].length; i++) {
            values.add(new BarEntry(i, result[0][i]*100.f));
        }

        BarDataSet set1;
        set1 = new BarDataSet(values, "Results in percentages");

        ArrayList<IBarDataSet> dataSets = new ArrayList<>();
        dataSets.add(set1);

        BarData data = new BarData(dataSets);
        data.setValueTextSize(12f);
        chart.setData(data);
        chart.animateY(400);
    }


    private ByteBuffer bitmapToModelsMatchingByteBuffer(Bitmap bitmap) {
        int SIZE = 28;
        ByteBuffer byteBuffer = ByteBuffer.allocateDirect(SIZE * SIZE * 4);
        byteBuffer.order(ByteOrder.nativeOrder());
        int[] intValues = new int[SIZE * SIZE];
        bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0,
bitmap.getWidth(), bitmap.getHeight());
        int pixel = 0;
        for (int i = 0; i < SIZE; ++i) {
            for (int j = 0; j < SIZE; ++j) {
                int pixelVal = intValues[pixel++];
                for (float channelVal : pixelToChannelValue(pixelVal)) {
                    byteBuffer.putFloat(channelVal);
                }
            }
        }
        return byteBuffer;
    }


    private float[] pixelToChannelValue(int pixel) {
        float[] singleChannelVal = new float[1];
        float rChannel = (pixel >> 16) & 0xFF;
        float gChannel = (pixel >> 8) & 0xFF;
        float bChannel = (pixel) & 0xFF;
        singleChannelVal[0] = (rChannel + gChannel + bChannel) / 3 / 255.f;
        return singleChannelVal;
    }

}
```

APPENDIX C

Code for integrating Paint View in the Application

```
package com.example.handwriting;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BlurMaskFilter;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.EmbossMaskFilter;
import android.graphics.MaskFilter;
import android.graphics.Paint;
import android.graphics.Path;
import android.util.AttributeSet;
import android.util.DisplayMetrics;
import android.view.MotionEvent;
import android.view.View;

import java.util.ArrayList;


public class PaintView extends View {

    public static int BRUSH_SIZE = 50;
    public static final int DEFAULT_COLOR = Color.WHITE;
    public static final int DEFAULT_BG_COLOR = Color.BLACK;
    private static final float TOUCH_TOLERANCE = 4;
    private float mX, mY;
    private Path mPath;
    private Paint mPaint;
    private ArrayList<FingerPath> paths = new ArrayList<>();
    private int currentColor;
    private int backgroundColor = DEFAULT_BG_COLOR;
    private int strokeWidth;
    private boolean emboss;
    private boolean blur;
    private MaskFilter mEmboss;
    private MaskFilter mBlur;
    private Bitmap mBitmap;
    private Canvas mCanvas;
    private Paint mBitmapPaint = new Paint(Paint.DITHER_FLAG);

    public PaintView(Context context) {
        this(context, null);
    }

    public PaintView(Context context, AttributeSet attrs) {
        super(context, attrs);
        mPaint = new Paint();
        mPaint.setAntiAlias(true);
        mPaint.setDither(true);
        mPaint.setColor(DEFAULT_COLOR);
        mPaint.setStyle(Paint.Style.STROKE);
        mPaint.setStrokeJoin(Paint.Join.ROUND);
        mPaint.setStrokeCap(Paint.Cap.ROUND);
        mPaint.setXfermode(null);
        mPaint.setAlpha(0xff);
```

```
        mEmboss = new EmbossMaskFilter(new float[] {1, 1, 1}, 0.4f, 6,
3.5f);
        mBlur = new BlurMaskFilter(5, BlurMaskFilter.Blur.NORMAL);
    }

    public void init(DisplayMetrics metrics) {
        int height = metrics.heightPixels;
        int width = metrics.widthPixels;

        mBitmap = Bitmap.createBitmap(width, height,
Bitmap.Config.ARGB_8888);
        mCanvas = new Canvas(mBitmap);

        currentColor = DEFAULT_COLOR;
        strokeWidth = BRUSH_SIZE;
    }

    public void normal() {
        emboss = false;
        blur = false;
    }

    public void emboss() {
        emboss = true;
        blur = false;
    }

    public void blur() {
        emboss = false;
        blur = true;
    }

    public void clear() {
        backgroundColor = DEFAULT_BG_COLOR;
        paths.clear();
        normal();
        invalidate();
    }

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.save();
        mCanvas.drawColor(backgroundColor);

        for (FingerPath fp : paths) {
            mPaint.setColor(fp.color);
            mPaint.setStrokeWidth(fp.strokeWidth);
            mPaint.setMaskFilter(null);

            if (fp.emboss)
                mPaint.setMaskFilter(mEmboss);
            else if (fp.blur)
                mPaint.setMaskFilter(mBlur);

            mCanvas.drawPath(fp.path, mPaint);

        }

        canvas.drawBitmap(mBitmap, 0, 0, mBitmapPaint);
        canvas.restore();
```

```
    }

    private void touchStart(float x, float y) {
        mPath = new Path();
        FingerPath fp = new FingerPath(currentColor, emboss, blur,
strokeWidth, mPath);
        paths.add(fp);

        mPath.reset();
        mPath.moveTo(x, y);
        mX = x;
        mY = y;
    }

    private void touchMove(float x, float y) {
        float dx = Math.abs(x - mX);
        float dy = Math.abs(y - mY);

        if (dx >= TOUCH_TOLERANCE || dy >= TOUCH_TOLERANCE) {
            mPath.quadTo(mX, mY, (x + mX) / 2, (y + mY) / 2);
            mX = x;
            mY = y;
        }
    }

    private void touchUp() {
        mPath.lineTo(mX, mY);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        float x = event.getX();
        float y = event.getY();

        switch(event.getAction()) {
            case MotionEvent.ACTION_DOWN :
                touchStart(x, y);
                invalidate();
                break;
            case MotionEvent.ACTION_MOVE :
                touchMove(x, y);
                invalidate();
                break;
            case MotionEvent.ACTION_UP :
                touchUp();
                invalidate();
                break;
        }

        return true;
    }
}
```

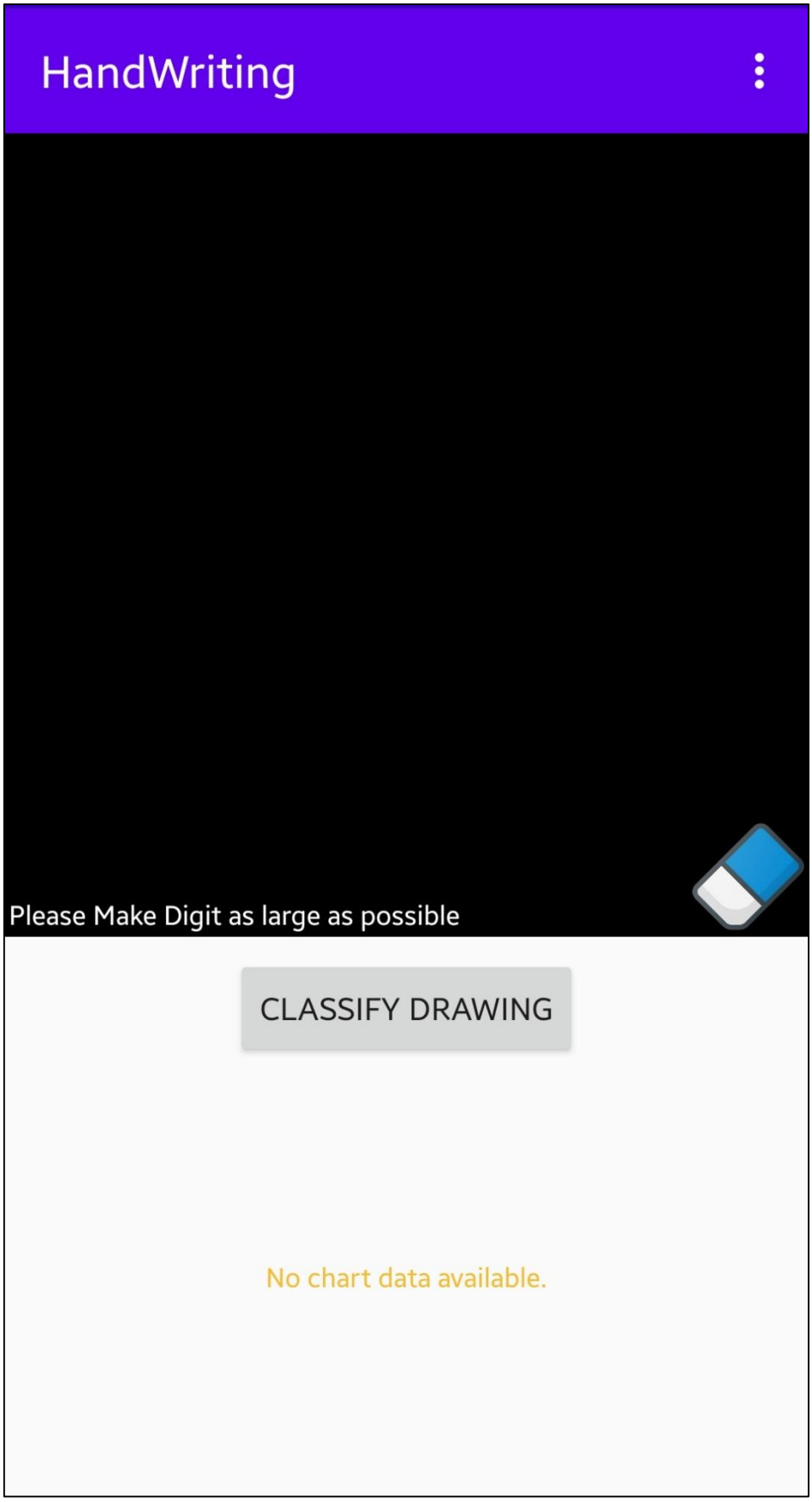
APPENDIX D

Code for Layout of the Android Application

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnClassify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Classify Drawing"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/paintView" />

    <com.example.handwriting.PaintView
        android:id="@+id/paintView"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/btnClear"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_marginEnd="2dp"
        android:layout_marginBottom="2dp"
        android:background="@android:color/transparent"
        android:scaleType="centerInside"
        app:layout_constraintBottom_toBottomOf="@+id/paintView"
        app:layout_constraintEnd_toEndOf="parent"
        app:srcCompat="@drawable/eraser" />

    <com.github.mikephil.charting.charts.HorizontalBarChart
        android:id="@+id/chart"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginStart="2dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="2dp"
        android:layout_marginBottom="2dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnClassify" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
```

```
        android:layout_height="wrap_content"
        android:layout_marginStart="2dp"
        android:layout_marginBottom="2dp"
        android:text="Please Make Digit as large as possible"
        android:textColor="#FFFFFF"
        android:textSize="12sp"
        app:layout_constraintBottom_toBottomOf="@+id/paintView"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Figure D: Screenshot of the Application Layout



APPENDIX E

Code for Menu in the Application

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/normal"
        app:showAsAction="never"
        android:title="Normal"/>

    <item
        android:id="@+id/emboss"
        app:showAsAction="never"
        android:title="Emboss"/>

    <item
        android:id="@+id/blur"
        app:showAsAction="never"
        android:title="Blur"/>

    <!--
    <item
        android:id="@+id/clear"
        app:showAsAction="never"
        android:title="Clear"/>
    -->
</menu>
```

Figure E: Screenshot of the Menu in the Application