HANDWRITTEN CHARACTER RECOGNITION

Mini Project Report

Submitted by

Arathy C P

Reg No: FIT20MCA-2033

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

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY- 683577,ERNAKULAM(DIST)

FEBRUARY 2022

DECLARATION

I, **Arathy C P** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of our original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE AND

TECHNOLOGY (FISAT)®
ANGAMALY, ERNAKULAM-683577

CERTIFICATE

This is to certify that the project report titled **"Handwritten Character Recoginition "** submitted by **Arathy C P** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by them during the year 2022.

Project Guide **Head of the Department**

Submitted for the viva-voice held on . . . . . at . . . . . .

ACKNOWLEDGEMENT

I am deeply grateful to **Dr.Manoj George** , Principal, FISAT, Angamaly for providing us with adequate facilities, way and means by which we were able to complete our mini project work and I express my sincere thanks to **Dr. C. Sheela** ,Vice Principal FISAT, Angamaly.

My sincere thanks to **Dr.Deepa Mary Mathwes**, Head of the department of MCA, FISAT, who had been a source of inspiration. I express heartiest thanks to **Dr.Deepa Mary Mathews** our project guide for their encouragement and valuable suggestion . I express our heartiest gratitude to our scrum master **Ms.Rose Mary Mathew** and the faculty members in our department for their constant encouragement and never ending support throughout the project.I would also like to express our sincere gratitude to the lab faculty members for their guidance

Finally I express our thanks to all my friends who gave me wealth of suggestion for successful completion of this project.

ABSTRACT

Handwriting recognition has been one of the active and challenging research areas in the field of image processing and pattern recognition. It has numerous applications which include, reading aid for blind, bank cheques and conversion of any hand written document into structural text form. In this paper an attempt is made to recognize handwritten characters for English alphabets without feature extraction using multilayer Feed Forward neural network. Each character data set contains 26 alphabets. Fifty different character data sets are used for training the neural network. The trained network is used for classification and recognition. In the proposed system, each character is resized into 30×20 pixels, which is directly subjected to training. That is, each resized character has 600 pixels and these pixels are taken as features for training the neural network. The results show that the proposed system yields good recognition rates which are comparable to that of feature extraction based schemes for handwritten character recognition.

CONTENTS

Chapter 1

INTRODUCTION

Handwriting recognition is the ability of a machine to receive and interpret handwrit- ten 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 finds extensive applications in banks, offices and industries.The main aim of this project is to design expert system for , **"HCR using Neural Network"**that can effectivelly recognize a particular character of type format using the Convolutional Neural Network approach. Neural computing Is comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computer are operated in way which is completely different from the operation of normal computers. Neural computer 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 optimizers.

Chapter 2

PROOF OF CONCEPT

This application is useful for recognizing all character(English) given as in input image. Once input image of character is given to proposed system, then it will recognize input character 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.

2.1 Benefits of Character Recognition

1.The idea of Neural Network in HCR will brings us the reading of various combined style of writing a character.

2.In forensic application HCR will be an effective method for evidence collection.

3.It will also help to reduce noise from the original character.

4.Our method develop accuracy in recognizing character in divert font and size.

5.More set of sample invites more accuracy rate because of heavy training and testing session.

Chapter3

IMPLEMENTATION

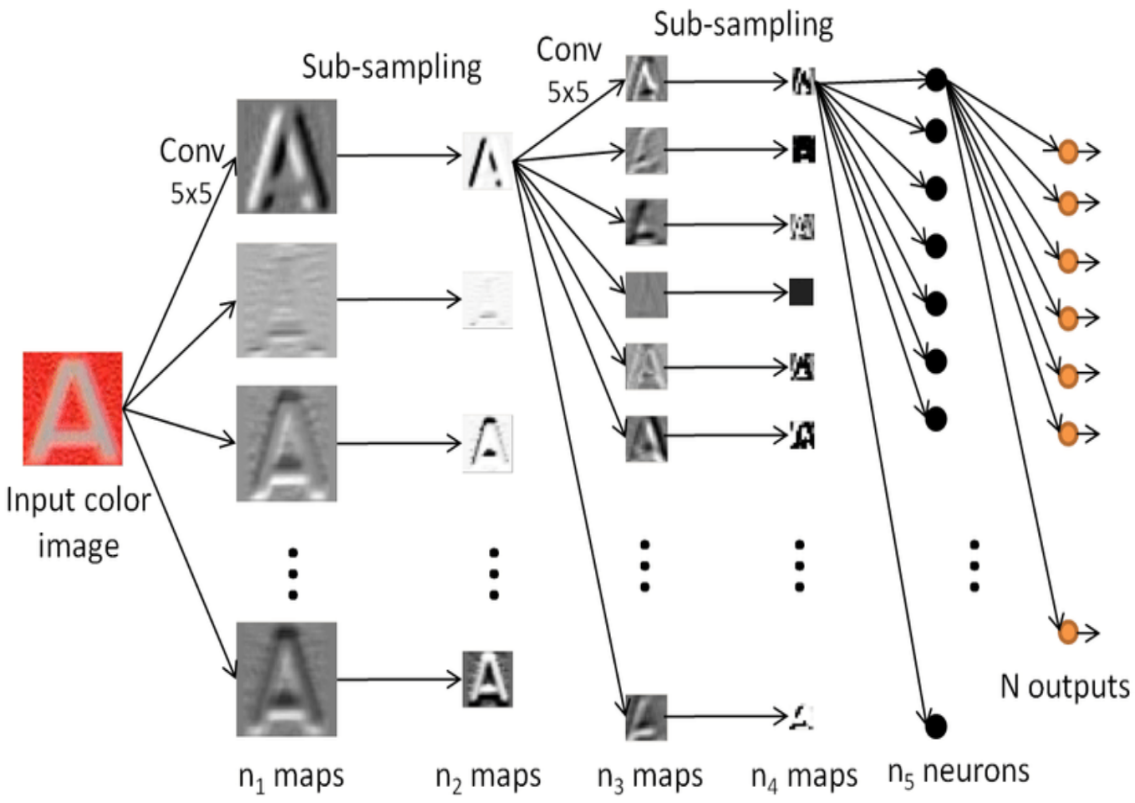
HCR works in stages as preprocessing, segmentation, feature extraction and recognition using neural network. Preprocessing includes series of operations to be carried out on document image to make it ready for segmentation. During segmentation the document image is segmented into individual character or numeric image then feature extraction technique is applied on character image. Finally feature vector is presented to the selected algorithm for recognition.Here this extracted features are provided to NN for recognition of character.

3.1 Architecture

Convolutional Neural Networks come under the subdomain of Machine Learning which is Deep Learning. Algorithms under Deep Learning process information the same way the human brain does, but obviously on a very small scale, since our brain is too complex (our brain has around 86 billion neurons).

CNN stands for Convolutional Neural Networks that are used to extract the features of the images using several layers of filters.The convolution layers are generally followed by maxpool layers that are used to reduce the number of features extracted and ultimately the output of the maxpool and layers and convolution layers are flattened into a vector of single

dimension and are given as an input to the Dense layer (The fully connected network).



Character Recognition Convolutional Neural Network architecture

3.2 Dataset

A dataset is a collection of data in which data is arranged in some order. A dataset can contain any data from a series of an array to a database table. To work with machine learning projects, we need a huge amount of data, because, without the data, one cannot train ML/AI models. Collecting and preparing the dataset is one of the most crucial parts while creating an ML/AI project.

The **dataset** for this project contains **372450 images of alphabets of 28×2**,all present in the form of a CSV file.Kaggle is one of the best sources for providing datasets for Data Scientists and Machine Learners.

The link for the Kaggle dataset used in this project is https://www.kaggle.com/datasets.

3.3 Modules

3.3.1 IMAGE PREPROCESSING

Preprocessing includes steps that are required to shape the input image into a form suitable for segmentation.

(1) Gray Scale Conversion : Color image is converted into gray scale.

(2) Image Binarization : Image transform into binary image that means in the form of black in white image.

3.3.2 SEGMENTATION

Once image preprocessing is done it is necessary to segment document into lines, lines into words and words into characters.

(1) Line Segmentation : we need to scan each horizontal pixel row starting from the top of document.This row acts as a separation between two lines.

(2) Word Segmentation : we need to scan each vertical pixel column starting from the left of line.This column acts as a separation between two words.

(3) Character Segmentation : we need to scan each vertical pixel column starting from the left of word.This column acts as a separation between two character.

3.3.3 FEATURE EXTRACTION

As individual characters has been separated, character image can be re sized to 15 x 20 pixels.

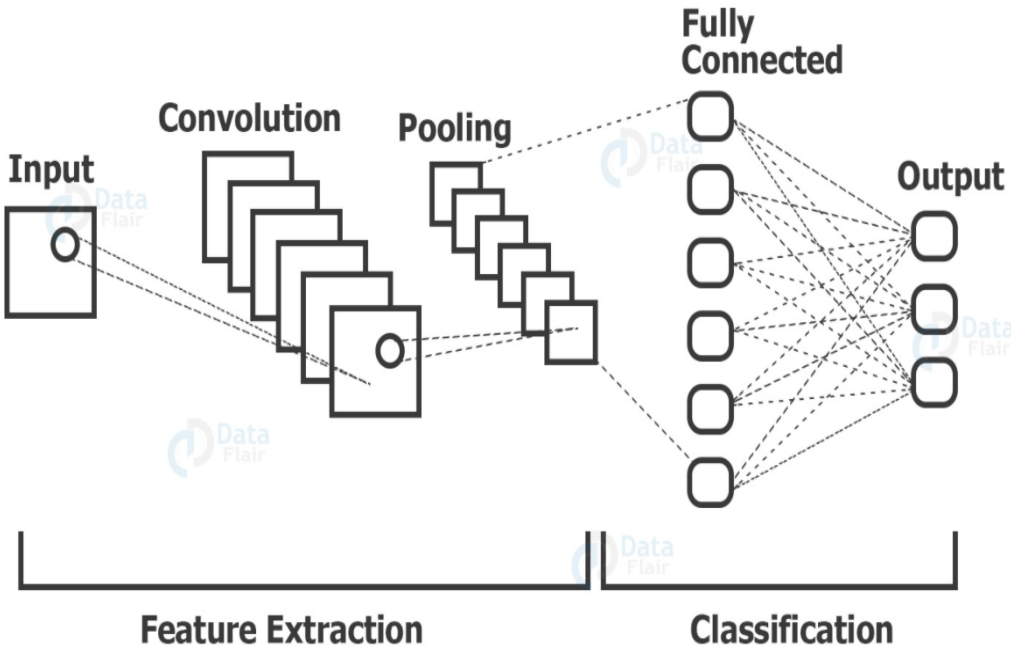
If the features are extracted accurately then the accuracy of recognition is more. Here we have use the 15 x 20 means 300 pixels as it is for feature vector.This extracted feature are stored in .dat file.

3.3.3 TRAINING AND RECOGINITION

The Features extracted from previous modules are given as an input for Neural Network.

3.4 Algorithm

Convolutional Neural Network(ConvNet/CNN) algorithm used for image classification and recognition because of high accuracy. The convolution layers are generally followed by maxpool layers that are used to reduce the number of features extracted and ultimately the output of the maxpool and layers and convolution layers are flattened into a vector of single dimension and are given as an input to the Dense layer (The fully connected network).


Fully
Connected
Convolution
Pooling
Input
Output
Feature Extraction
Classification

Chapter 4

RESULT ANALYSIS

The result of the proposed project Handwritten Character Recoginition using CNN algorithm lies in developing a model that can successfully recognize the charcters and perform task based on it. Handwritten characters have been recognized with more than 97% test accuracy. This can be also further extended to identifying the handwritten characters of other languages too.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

A neural network based off line handwritten character recognition system feature extraction has been introduced in this paper for classifying and recognizing the 26 English alphabets. The pixel values derived from the resized characters of the segmentation stage have been directly used for training the neural network. As a result, the proposed system will be less complex compared to the offline methods using feature extraction techniques. Of the several neural networks architectures used for classifying the characters, the one with two hidden layers each having 100 neurons has been found to yield the highest recognition accuracy of 90.19%. The handwritten recognition system described in this project will find potential applications in handwritten character recognition, document reading, conversion of any handwritten document into structural text form and postal address recognition.

5.2 Future scope

This work further extended to the character recognition for other languages. It can be used to convert the fax and news papers into text format.In order to recognize words,sentences or paragraphs we can use multiple CNN for classification.It can be used in post office for reading postal address.

Chapter 6

APPENDIX

6.1 Coding

6.1.1 Utils.py

```
import matplotlib.pyplot as plt

import cv2

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Dropout

from keras.optimizers import SGD, Adam

from keras.callbacks import ReduceLROnPlateau, EarlyStopping

from keras.utils import to_categorical

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.utils import shuffle
```

```
# Read the data

data = pd.read_csv(r"D:\a-z alphabets\A_Z Handwritten Data.csv").astype('float32')

print(data.head(10))

# Split data into images and their labels

X=data.drop('0',axis=1)

y = data['0']

# Reshaping the data in the csv file so that it can be displayed as an image

train_x, test_x, train_y, test_y = train_test_split(X, y, test_size = 0.2)

train_x = np.reshape(train_x.values, (train_x.shape[0], 28,28))

test_x = np.reshape(test_x.values, (test_x.shape[0], 28,28))

print("Train data shape: ", train_x.shape)

print("Test data shape: ", test_x.shape)

# Plotting the number of alphabets in the dataset

y_int = np.int0(y)

count = np.zeros(26, dtype='int')

for i in y_int:

count[i] +=1

alphabets = []

for i in word_dict.values():
```

```
alphabets.append(i)

fig, ax = plt.subplots(1,1, figsize=(10,10))

ax.barh(alphabets, count)

plt.xlabel("Number of elements ")

plt.ylabel("Alphabets")

plt.grid()

plt.show()
```

Shuffling the data

```
shuff = shuffle(train_x[:100])

fig, ax = plt.subplots(3,3, figsize = (10,10))

axes = ax.flatten()

for i in range(9):

_, shu = cv2.threshold(shuff[i], 30, 200, cv2.THRESH_BINARY)

axes[i].imshow(np.reshape(shuff[i], (28,28)), cmap="Greys")

plt.show()
```

Reshaping the training & test dataset so that it can be put in the model

```
train_X = train_x.reshape(train_x.shape[0],train_x.shape[1],train_x.shape[2],1)

print("New shape of train data: ", train_X.shape)

test_X = test_x.reshape(test_x.shape[0], test_x.shape[1], test_x.shape[2],1)
```

```
print("New shape of train data: ", test_X.shape)

#Categorical

train_yOHE = to_categorical(train_y, num_classes = 26, dtype='int')

print("New shape of train labels: ", train_yOHE.shape)

test_yOHE = to_categorical(test_y, num_classes = 26, dtype='int')

print(“New shape of test labels: “,test_yOHE.shape)

#Model creation

model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))

model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))

model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'valid'))

model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Flatten())

model.add(Dense(64,activation ="relu"))

model.add(Dense(128,activation ="relu"))

model.add(Dense(26,activation ="softmax"))

# Compiling & Fitting Model

model.compile(optimizer = Adam(learning_rate=0.001),
```

```
loss='categorical_crossentropy', metrics=['accuracy'])

history=model.fit(train_X,yOHE,epochs=1,

validation_data=(test_X.test_yOHE))

model.summary()

model.save(r'model_hand.h5')
```

#Getting the Train & Validation Accuracies & Losses

```
print("The validation accuracy is :", history.history['val_accuracy'])

print("The training accuracy is :", history.history['accuracy'])

print("The validation loss is :", history.history['val_loss'])

print("The training loss is :", history.history['loss'])
```

6.1.2 Recognition.py

#Doing Some Predictions on Test Data

```
fig, axes = plt.subplots(3,3, figsize=(8,9))

axes = axes.flatten()

for i,ax in enumerate(axes):

img = np.reshape(test_X[i], (28,28))

ax.imshow(img, cmap="Greys")

pred = word_dict[np.argmax(test_yOHE[i])]
```

```
ax.set_title("Prediction: "+pred)

ax.grid()
```

#Doing Prediction on External Image

```
img = cv2.imread(r'/content/A.jpeg')

img_copy = img.copy()

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

img =cv2.resize(img(400,440)

img_copy = cv2.GaussianBlur(img_copy, (7,7), 0)

img_gray = cv2.cvtColor(img_copy, cv2.COLOR_BGR2GRAY)

img_thresh = cv2.threshold(img_gray, 100, 255, cv2.THRESH_BINARY_INV)

img_final = cv2.resize(img_thresh, (28,28))

img_final=pp.reshape(img,final,(1,28,28,1)

img_pred = word_dict[np.argmax(model.predict(img_final))]

cv2.putText(img, "Character is _ _ _ ", (20,25), cv2.FONT_HERSHEY_TRIPLEX, 0.7,

color = (0,0,230))

cv2.putText(img, "Prediction:  "+img_pred,(20,410) ,cv2.FONT_HERSHEY_DUPLEX, 1.3,

color = (255,0,30))

from google.colab.patches import cv2_imshow

cv2_imshow(img)
```

#UI Design

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta http-equiv="Cache-Control" content="no-cache" />
  <meta http-equiv="Pragma" content="no-cache" />
  <meta http-equiv="Expires" content="0" />
  <meta http-equiv="Pragma-directive: no-cache" />
  <meta http-equiv="Cache-directive: no-cache" />

  <style>
   .main {
    height: 90vh;
    width: 100vw;
    overflow: hidden;
    display: flex;
    align-items: center;
    justify-content: center;
   }
  </style>
  <link
   href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
```

```
    rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
    crossorigin="anonymous"
   />

  <title>Handwritten Charcter Recognition</title>
 </head>
 <body>
  <section>
   <div class="main container">
    <div>
     <h1>Handwritten Character Recognition</h1>
     <hr />
     <!-- <h4>Upload Your Image</h4> -->
     <form action="/predict" method="post" enctype="multipart/form-data">
      <div class="mb-3">
       <label for="formFileSm" class="form-label"
        >upload your image to Recoginize</label
       >
       <input
        class="form-control form-control-sm"
        name="file"
        id="file"
```

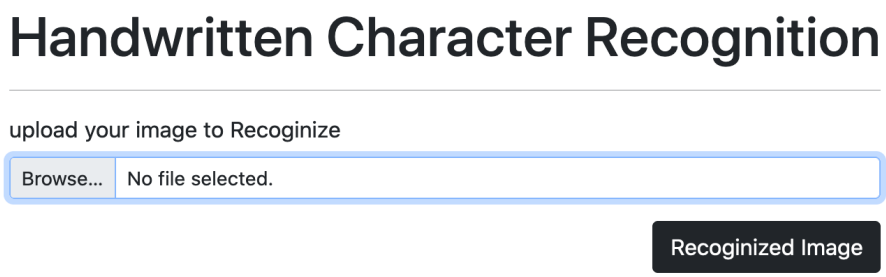
```
                type="file"
                accept="image"
               />
              </div>
              <button style="float: right" type="submit" class="btn btn-dark">
               Recoginized Image
              </button>
             </form>
            </div>
           </div>
          </section>
          <footer>
           <h6 class="text-center">Mini Project- Arathy C P</h6>
          </footer>

          <script
           src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
           integrity="sha384-
        ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
           crossorigin="anonymous"
          ></script>
         </body>
</html>
```

Chapter 7

SCREENSHOT

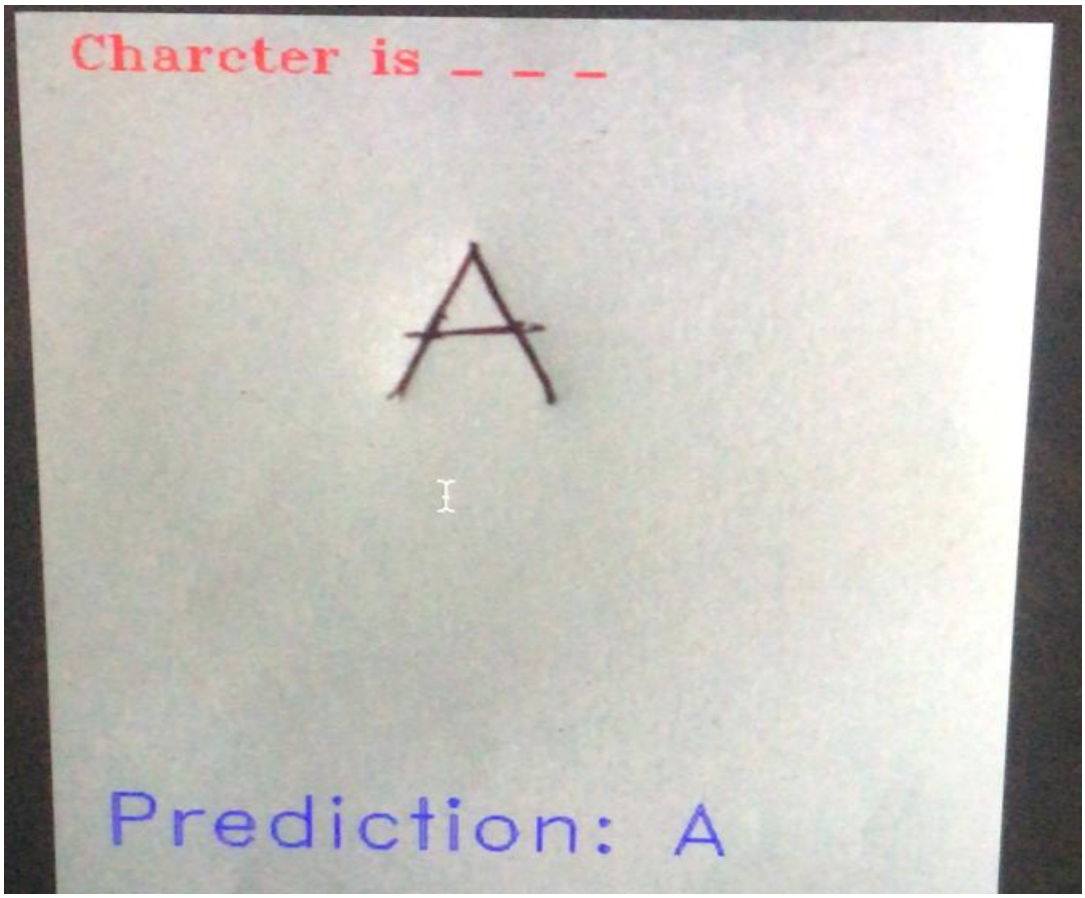
Here are some sample screenshots of the proposed system which

includes: Home Screen, Image Upload Form, and the Prediction Screen



Mini Project- Arathy C P

Home screen



Prediction of A

Chapter 8

REFERENCES

(a) https://ieeexplore.ieee.org

(b) www.google.com

(c) www.youtube.com

(d) www.wikipedia.com