A Project Report

on

HANDWRITTEN CHARACTER RECOGNITION

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

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. A. Arul
Assistant Professor
Department of Computer Science and Engineering

Submitted By

19SCSE1180058 – AMAN DWIVEDI

19SCSE1010491 – SHREYASHI CHAUDHARY

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **" Handwritten character recognition "** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **Mr.A. ARUL, Assistant Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

19SCSE1180058 – AMAN DWIVEDI

19SCSE1010491 – SHREYASHI CHAUDHARY

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Mr.A.Arul, Assistant Professor)

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voice examination of **19SCSE1180058 – AMAN DWIVEDI, 19SCSE1010491 -SHREYASHI CHAUDHARY** has been held on ____and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

ABSTRACT

We are writting this paper to design a system which predicts the handwritten words with more accuracy

The paper will describe the best approach to get more than 90% accuracy in the field of Handwritten Character Recognition (HCR). There have been plenty of research done in the field of HCR but still it is an open problem as we are still Lacking in getting the bestaccuracy.

In this paper, the offline handwritten character recognition will be done using Convolutionalneural network and Tensor flow. A method called Soft Max Regression is used for assigningthe probabilities to handwritten characters being one of the several characters as it gives the values between 0 and 1 summing up to 1. The purpose is to develop the software with a veryhigh accuracy rate and with minimal time and space complexity and also optimal.

We also aim to develop it in all three languages.

Table of Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Networks
ML	Machine Learning
DL	Deep Learning
CNN	Convolution Neural Networks

CHAPTER-1

Introduction

Understanding the handwritten characters or typed documents is simple to the human beings aswe have the ability to learn. The same ability can be induced to the Machines also by the use of Machine Learning and Artificial Intelligence. The field which deals with this problem is called as the OCR or also known as Optical Character Recognition. It's the area of study among various fields such as recognizing of pattern, also Image vision and also AI.

This is the system for changing electronic and image text into the digital character to be read by the machines.The time used in entering the data and also the storage space required by the documents can be highly reduced by the use of OCR or in other words it can be retrieved fast. By using the OCR in banking field, legal scenarios, etc. Many important and sensitive documents can be processed faster without human intervention. OCR in advance can be inferred in two ways based on type of the text and document acquisition (Figure 1).

Further if we take into consideration the text type, then OCR is further of two types, HCR (Handwritten Character Recognition) which is intelligent recognition of the handwritten text and PCR (Printed Character Recognition). We need the high recognition ability due to the varying handwriting of the humans. Many a times the writing style of same individual is different at times. Further OCR is characterized into two forms as Offline and Online recognition systems based on acquiring of the documents. Offline System deals with recognizing the pre written document acquired through various input methods. But in Online recognizing system, the writing is recognized the moment it is written.The device used for the online system is Electric pen where it is used for writing the letters or words on the device called as digitizer and on the basis of the pen movement the input is recorded.

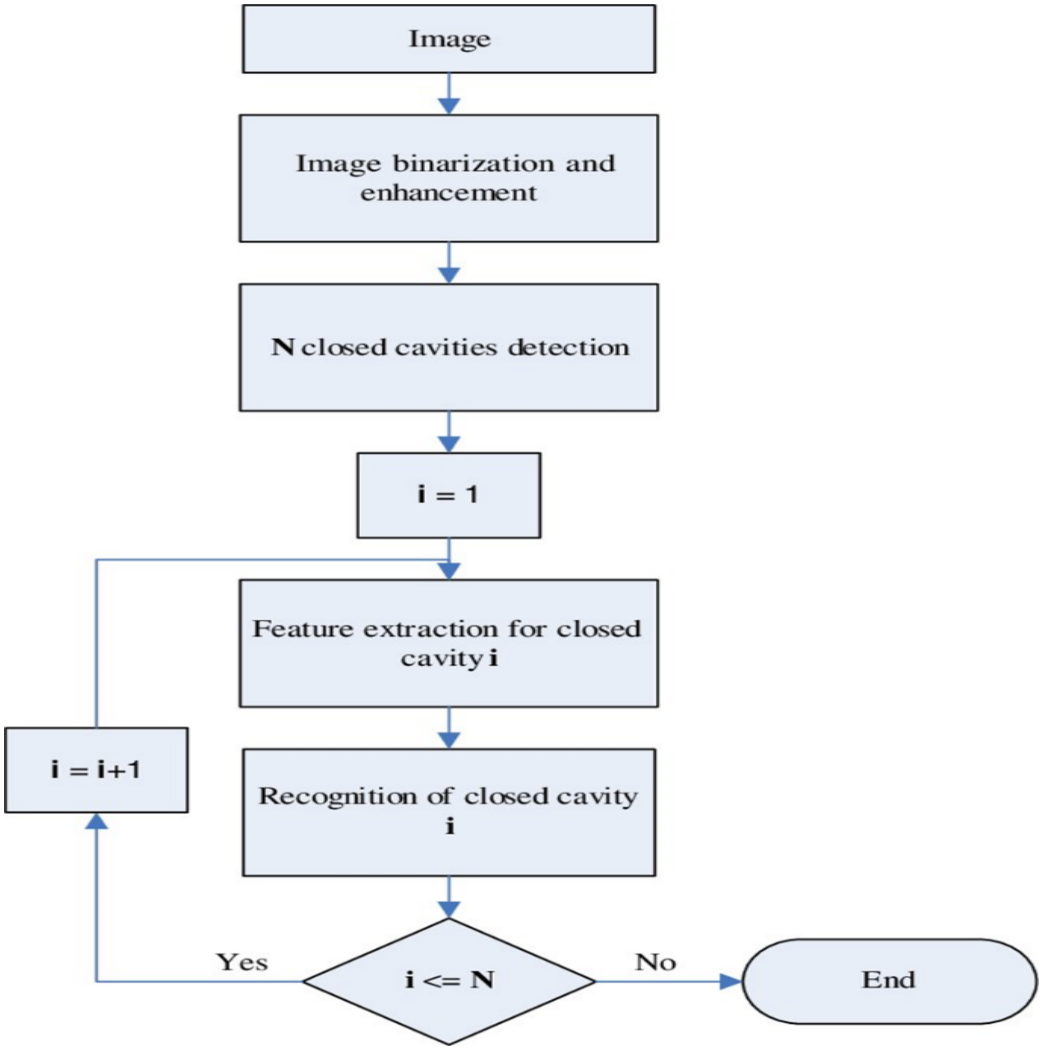


Image
Image binarization and enhancement
N closed cavities detection
i = 1
Feature extraction for closed cavity i
Recognition of closed cavity i
i = i+1
Yes
i <= N
No
End

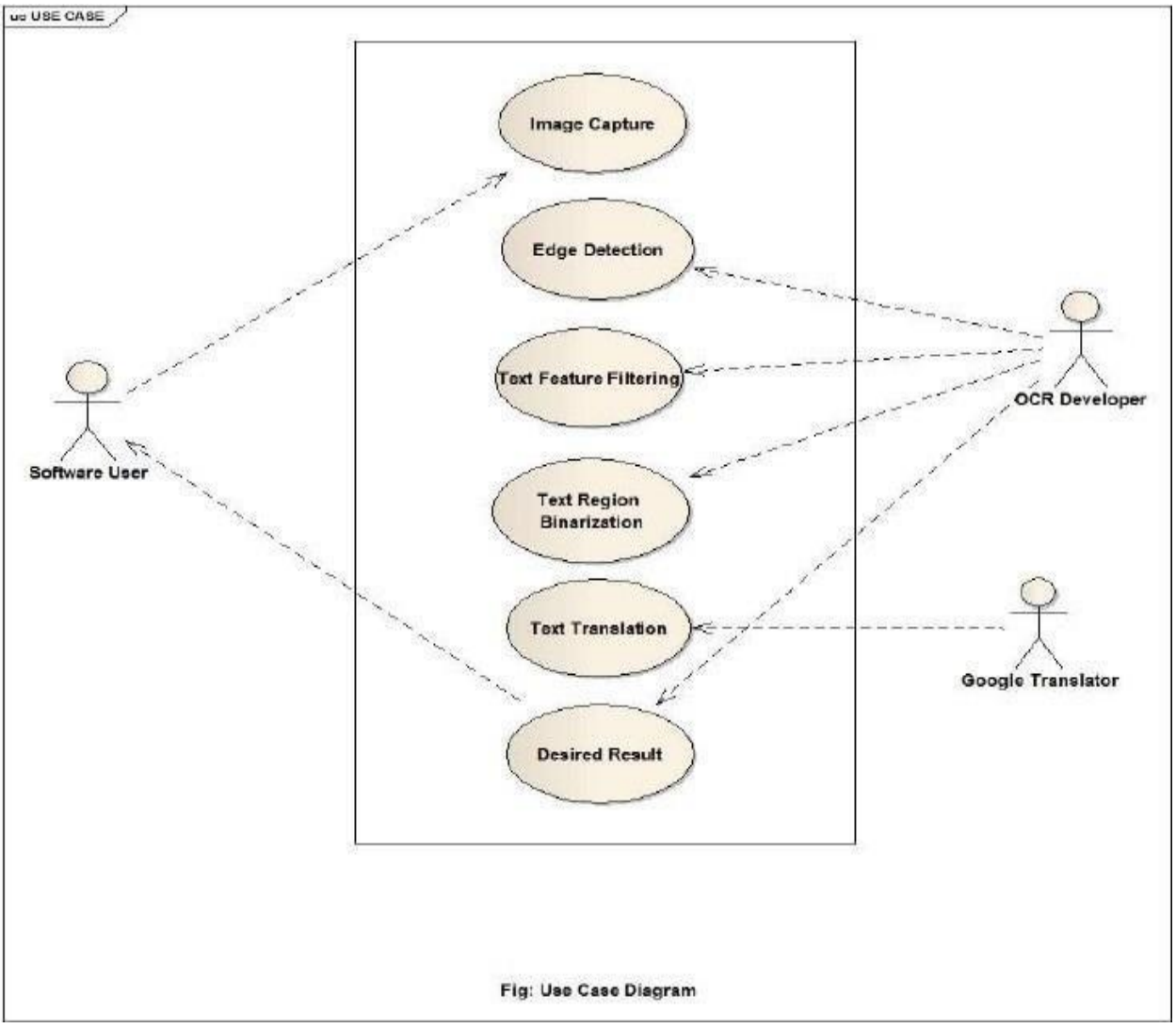


Fig: Use Case Diagram

CHAPTER 2
Literature Survey and Publication Details

Research in the region of word recognition, being done from Grimsdale in the year 1959 is soonest endeavor to perceive the handwritten character. This mid-sixty research exhibited theutilization of examination by combination strategy being proposed by the eden in 1968.He demonstrated that every single handwritten character is limited to number of schematic highlights. This hypothesis was later utilized as a part of almost all strategies for auxiliary methodologies in the region of character recognition.

K. Gaurav and Bhatia P. K [2], proposed different prehandling systems being associated with the recognition of the characters. The procedure took a shot at the various types of pictures from a basic picture-based report to a hued and changed forces including foundation. Differentsystems of pre-handling and standardization like skew remedy, differentiate evacuation, commotion expulsion and numerous other upgrade procedures were recommended. They reached the decision that a solitary procedure can't be connected for preprocessing the picture.

Yet additionally there were a few disparities that utilizing every one of these systems likewisecan't give the best exactness comes about. Salvador España-Boquera [3], The analysts proposed the utilization of hybrid or half plus half concealed markov show (HMM) to perceivethe handwritten content in disconnected mode. The optical model's basic part was prepared with markov chain procedure and a multilayer perceptron was likewise used to gauge the probabilities.

In [4], to perceive the disconnected handwritten numerals of six prominent Indian language, a changed quadratic classifier is utilized. A similar paper likewise manages perceiving the English letters in order. For both of these, a multilayer perceptron was utilized and Boundary following and Fourier descriptors were utilized for the component extraction. By examining the shape and looking at their highlights, the characters were identified. Also, to decide the quantity of concealed layers, back spread system was utilized. With this very calculation, a recognition rate of 94% have been accounted for with less preparing time.

CHAPTER 3

Requirements, Feasibility and Scope Objective

HARDWARE AND SOFTWARE REQUIREMENTS

HIGH LEVEL SPECIFICATIONS

PROCESSOR dual core or above

RAM 2GB RAM

DISK SPACE Disk Space varies depending on size of partition and installation of online help files

LOW LEVEL SPECIFICATIONS

1 Microsoft windows supported graphics accelerator card, printer and sound card

2 Microsoft word 8.0(office 97) , office 2000

3 TCP/IP is required on all platforms when using license server

FUNCTIONAL REQUIREMENTS

1 The system should process the input given by the user only if it is an image file (jpg,png)

2 System shall show error message to the user when the input given is not in the reuired format

3 System shall detect characters present in the image

4 System shall retrieve characters present in image and display them to the user

NON FUNCTIONAL REQUIREMENTS

PERFORMANCE

Handwritten characters in the input image will ve recognized with an accuracy of about 90% and more

FUNCTIONALITY

This software will deliver on the functional requirements mentioned in the document

AVAILABILITY

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

FLEXIBILTY

It provides users to load the image easily

LEARN ABILITY

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

OBJECTIVE OF THE PROJECT

The objective of this project is to identify handwritten characters with the use of neural networks. We have to construct suitable neural network and train it properly. The program should be able to extract the characters one by one and map the target output for training purpose. After automatic processing of the image, the training dataset has to be used to train “classification engine” for recognition purpose. The program code has to be written in MATLAB and supported with the usage of Graphical User Interface (GUI)

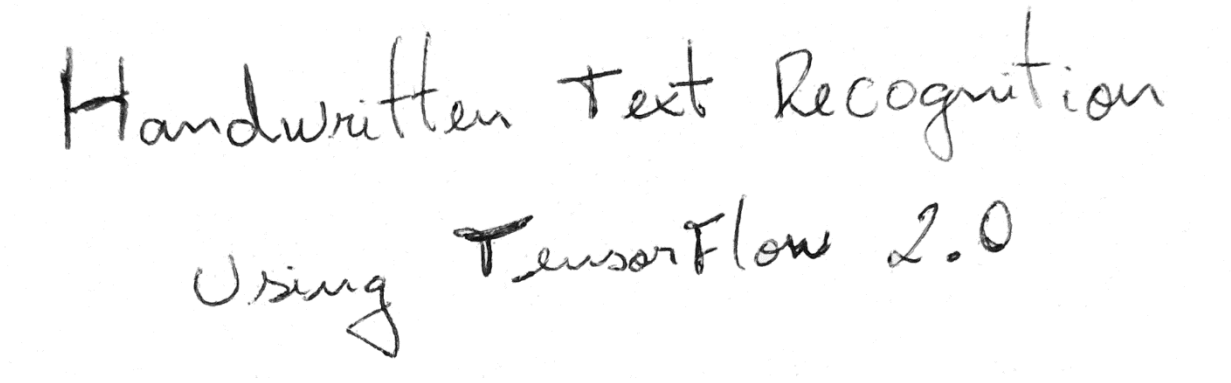
APPROACH

To solve the defined handwritten character recognition problem of classification we used MATLAB computation software with Neural Network Toolbox and Image Processing Toolbox add-on. The computation code is divided into the next categories:

• Pre-processing of the image • Feature extraction • Creating an Artificial Neural Network • Training & Testing of the network • Recognition

CHAPTER 4

IMPLEMENTATION



We are going to recognize this image

Localhost Environment
We'll make sure you have the project in your Google Drive with the datasets in HDF5. If you already have structured files in the cloud, skip this step.

↳ 3 cells hidden

2 Google Drive Environment
2.1 TensorFlow 2.x
Make sure the jupyter notebook is using GPU mode.

```
!nvidia-smi
```

```
%tensorflow_version 2.x
import tensorflow as tf

device_name = tf.test.gpu_device_name()

if device_name != "/device:GPU:0":
    raise SystemError("GPU device not found")

print("Found GPU at: {}".format(device_name))
```

2.2 Google Drive
Mount your Google Drive partition.

Note: "Colab Notebooks/handwritten-text-recognition/src/" was the directory where you put the project folders, specifically the src folder.

```
from google.colab import drive

drive.mount("./gdrive", force_remount=True)

%cd "./gdrive/My Drive/Colab Notebooks/handwritten-text-recognition/src/"
!ls -l
```

After mount, you can see the list os files in the project folder.

3 Set Python Classes
3.1 Environment
First, let's define our environment variables.

Set the main configuration parameters, like input size, batch size, number of epochs and list of characters. This make compatible with main.py and jupyter notebook:

dataset: "bentham", "iam", "rimes", "saintgall", "washington"

arch: network to run: "bluche", "puigcerver", "flor"

epochs: number of epochs

batch_size: number size of the batch

```
import os
import datetime
import string

# define parameters
source = "bentham"
arch = "flor"
epochs = 1000
batch_size = 16

# define paths
source_path = os.path.join("..", "data", f"{source}.hdf5")
output_path = os.path.join("..", "output", source, arch)
target_path = os.path.join(output_path, "checkpoint_weights.hdf5")
```

```
os.makedirs(output_path, exist_ok=True)

# define input size, number max of chars per line and list of valid chars
input_size = (1024, 128, 1)
max_text_length = 128
charset_base = string.printable[:95]

print("source:", source_path)
print("output", output_path)
print("target", target_path)
print("charset:", charset_base)
```

3.2 DataGenerator Class
The second class is DataGenerator(), responsible for:

Load the dataset partitions (train, valid, test);

Manager batchs for train/validation/test process.

```
from data.generator import DataGenerator

dtgen = DataGenerator(source=source_path,
                      batch_size=batch_size,
                      charset=charset_base,
                      max_text_length=max_text_length)

print(f"Train images: {dtgen.size['train']}")
print(f"Validation images: {dtgen.size['valid']}")
print(f"Test images: {dtgen.size['test']}")
```

3.3 HTRModel Class

The third class is HTRModel(), was developed to be easy to use and to abstract the complicated flow of a HTR system. It's responsible for:

Create model with Handwritten Text Recognition flow, in which calculate the loss function by CTC and decode output to calculate the HTR metrics (CER, WER and SER);

Save and load model;

Load weights in the models (train/infer);

Make Train/Predict process using generator.

To make a dynamic HTRModel, its parameters are the architecture, input_size and vocab_size.

```
from network.model import HTRModel

# create and compile HTRModel
model = HTRModel(architecture=arch,
                 input_size=input_size,
                 vocab_size=dtgen.tokenizer.vocab_size,
                 beam_width=10,
                 stop_tolerance=20,
                 reduce_tolerance=15)

model.compile(learning_rate=0.001)
model.summary(output_path, "summary.txt")

# get default callbacks and load checkpoint weights file (HDF5) if exists
model.load_checkpoint(target=target_path)

callbacks = model.get_callbacks(logdir=output_path, checkpoint=target_path,
verbose=1)
```

4 Training

The training process is similar to the fit() of the Keras. After training, the information (epochs and minimum loss) is save.

```
# to calculate total and average time per epoch
start_time = datetime.datetime.now()

h = model.fit(x=dtgen.next_train_batch(),
              epochs=epochs,
              steps_per_epoch=dtgen.steps['train'],
              validation_data=dtgen.next_valid_batch(),
              validation_steps=dtgen.steps['valid'],
              callbacks=callbacks,
              shuffle=True,
              verbose=1)

total_time = datetime.datetime.now() - start_time

loss = h.history['loss']
val_loss = h.history['val_loss']

min_val_loss = min(val_loss)
min_val_loss_i = val_loss.index(min_val_loss)
```

```
time_epoch = (total_time / len(loss))
total_item = (dtgen.size['train'] + dtgen.size['valid'])

t_corpus = "\n".join([
    f"Total train images:      {dtgen.size['train']}",
    f"Total validation images: {dtgen.size['valid']}",
    f"Batch:                   {dtgen.batch_size}\n",
    f"Total time:              {total_time}",
    f"Time per epoch:          {time_epoch}",
    f"Time per item:           {time_epoch / total_item}\n",
    f"Total epochs:            {len(loss)}",
    f"Best epoch               {min_val_loss_i + 1}\n",
    f"Training loss:           {loss[min_val_loss_i]:.8f}",
    f"Validation loss:         {min_val_loss:.8f}"
])

with open(os.path.join(output_path, "train.txt"), "w") as lg:
    lg.write(t_corpus)
    print(t_corpus)
```

5 Predict

The predict process is similar to the predict of the Keras:

```
from data import preproc as pp
from google.colab.patches import cv2_imshow

start_time = datetime.datetime.now()

# predict() function will return the predicts with the probabilities
predicts, _ = model.predict(x=dtgen.next_test_batch(),
                            steps=dtgen.steps['test'],
                            ctc_decode=True,
                            verbose=1)

# decode to string
predicts = [dtgen.tokenizer.decode(x[0]) for x in predicts]
ground_truth = [x.decode() for x in dtgen.dataset['test']['gt']]

total_time = datetime.datetime.now() - start_time

# mount predict corpus file
with open(os.path.join(output_path, "predict.txt"), "w") as lg:
    for pd, gt in zip(predicts, ground_truth):
        lg.write(f"TE_L {gt}\nTE_P {pd}\n")

for i, item in enumerate(dtgen.dataset['test']['dt'][:10]):
```

```
    print("=" * 1024, "\n")
    cv2_imshow(pp.adjust_to_see(item))
    print(ground_truth[i])
    print(predicts[i], "\n")
```

6 Evaluate

Evaluation process is more manual process. Here we have the ocr_metrics, but feel free to implement other metrics instead. In the function, we have three parameters:

predicts
ground_truth
norm_accentuation (calculation with/without accentuation)
norm_punctuation (calculation with/without punctuation marks)

```
from data import evaluation

evaluate = evaluation.ocr_metrics(predicts, ground_truth)

e_corpus = "\n".join([
    f"Total test images:    {dtgen.size['test']}",
    f"Total time:           {total_time}",
    f"Time per item:        {total_time / dtgen.size['test']}\n",
    f"Metrics:",
    f"Character Error Rate: {evaluate[0]:.8f}",
    f"Word Error Rate:      {evaluate[1]:.8f}",
    f"Sequence Error Rate:  {evaluate[2]:.8f}"
])

with open(os.path.join(output_path, "evaluate.txt"), "w") as lg:
    lg.write(e_corpus)
    print(e_corpus)
```

CHAPTER 5

COCLUSION AND FUTURE IMPROVEMENTS

5.1 CONCLUSION

Classification of characters and learning of image processing techniques is done in this project. Also the scheme through which project is achieved is Artificial Neural Network scheme. The result which was got was correct up to more than 90% of the cases, but it would be improved at the end. This work was basically focused on envisaging methods that can efficiently extract feature vectors from each individual character. The method I came up with gave efficient and effective result both for feature extraction as well as recognition. There are also different methods through which 'handwritten character recognition' is achieved.

5.2 FUTURE SCOPE OF THIS PROJECT

The application of this HCR algorithm is extensive. Now-a-days recent advancement in technologies has pushed the limits further for man to get rid of older equipment which posed inconvenience in using. In our case that equipment is a keyboard. There are many situations when using a keyboard is cumbersome like,

- We don't get fluency with keyboard as real word writing
- When any key on keyboard is damaged
- Keyboard have scripts on its keys in only one language
- We have to find each character on keyboard which takes time
- In touch-enabled portable devices it is difficult to add a keyboard with much ease

REFERENCES

1. Singh, Sameer, Mark Hewitt,"Cursive Digit And Character Recognition on Cedar Database", Pattern Recognition, 2000.Proceedings. 15th international conference on. Vol. 2.IEEE 2000.

2. K. Gaurav and Bhatia P. K., "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition", 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

3. Salvador España-Boquera, Maria J. C. B., Jorge G. M. andFrancisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

4. U. Pal, T. Wakabayashi and F. Kimura, "Handwritten numeralrecognition of six popular scripts," Ninth International conference on Document Analysis and Recognition ICDAR
07, Vol.2, pp.749-753, 2007.