OFFLINE MALAYALAM HANDWRITING RECOGNITION

B Tech Major Project

Submitted in partial fulfillment for the award of the Degree of

Bachelor of Technology in Electrical and Electronics Engineering

By

AISWARYA SASIKALADHARAN B120351EE

RAVI AMAR B120256EE

KRITHIKA ARJUN B120671EE

SHARAT T.OOMMEN B120241EE

Under the guidance of

Dr. Jeevamma Jacob

Department of Electrical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

NIT Campus P.O. Calicut - 673601, India

2016

CERTIFICATE

This is to certify that the thesis entitled "OFFLINE MALAYALAM HANDWRITING RECOGNITION" is a bona fide record of the major project done by ***AISWARYA SASIKALADHARAN(B120351EE), RAVI AMAR (B120256EE), KRITHIKA ARJUN (B120671EE) and SHARAT T OOMMEN(B120241EE)*** *under my supervision and guidance, in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Electrical & Electronic Engineering from National Institute of Technology Calicut for the year 2015.*

Dr Jeevamma Jacob
(Project Guide)
Professor & Head
Department Of Electrical Engineering

Place:

Date:

ACKNOWLEDGMENT

We express our sincere gratitude to our project guide Dr. Jeevamma Jacob for her support and guidance throughout the project. We thank her for providing us appropriate guidance to pursue a challenging project in the field of pattern recognition. This helped us expand our knowledge in various aspects. We would also like to thank her for providing us with this learning platform under her department. We would like to express our sincere gratitude towards Dr. Paul Joseph for providing us with excellent reference materials for getting started with character recognition. We thank our fellow classmates for their help and encouragement.

ABSTRACT

Handwriting recognition is a very challenging area of research in the field of pattern recognition. The state of research in handwriting character recognition is well developed and mature for English as well as other foreign languages like Japanese and Chinese, but the problem is much more complex for Indian languages. The complexity increases for South Indian languages because of the large character set and the presence of vowel modifiers and joint characters. This project attempts to create an offline handwriting recognition system for Malayalam and develop it into an application with a graphical user interface that can be used by a person with basic computer operation knowledge.

CONTENTS

LIST OF ABBREVIATIONS

HCR	: Handwritten Character Recognition
OCR	: Optical Character Recognition
SSM	: State Space Map
SSPD	: State Space Point Distribution
WEF	: Wavelet Energy Feature
ELM	: Extreme Learning Machine
RLC	: Run Length Count
MQDF	: Moderate Quadratic Discriminate Function
ARTMAP	: Adaptive Resource Theory MAP
SVM	: Support Vector Machine
RBF	: Radial Basis Function
GUI	: Graphical User Interface

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

Character recognition has been one of the most active areas of research in pattern recognition for several decades. It is the process of converting an image representation of a document into a digital format. The document can be a printed document or hand written. Handwritten data can be recognized in two ways, online and offline. Online handwriting recognition is done using a special pen on an electronic surface such as a digitizer combined with a LCD; Offline recognition is carried out on a scanned/photographed image of the handwritten document. In case of online handwriting, the order of strokes and spatial arrangements are easily available while only the final image is available in offline handwriting recognition, making offline handwriting recognition more complex. In this project we aim to create an offline handwriting recognition system to identify Malayalam character images and make this system more accessible by supplementing it with an end user application that has an easy to use graphical user interface.

1.2 MOTIVATION

Malayalam character recognition poses a difficulty due to the large number of characters and various instances of highly similar characters. Malayalam is one of 22 scheduled languages of India, with rich literary heritage [1]. It is the official language of the State of Kerala and Union territories of Lakshadweep and Mahe. It's spoken by around 35 million people and it is ranked eighth in terms of the number of speakers in India. Most of the alphabets have circular shapes. It is phonetic because words are written exactly as they are pronounced. It is syllabic in nature and alphabets are classified into vowels and consonants. The script consists of 15 vowels and 36 consonants, called basic characters. In addition to this basic set, the script contains vowel modifiers, half consonants and a large number of compound characters.

The set of vowels and consonants are depicted in Fig. 1.1

vowels				consonants									
അ	ആ	ഇ	ഈ	ക	ഖ	ഗ	ഘ	ങ	ച	ഛ	ജ	ഝ	ഞ
ഉ	ഊ	ഋ	എ	ട	ഠ	ഡ	ഢ	ണ	ത	ഥ	ദ	ധ	ന
ഏ	ഐ	ഒ	ഓ	പ	ഫ	ബ	ഭ	മ	യ	ര	ല	വ	ശ
ഔ	അം	അഃ		ഷ	സ	ഹ	ള	ഴ	റ				

Figure 1.1 - Vowels and consonants used in Malayalam

The state of HCR systems is still underdeveloped in case of Malayalam as compared to other scripts like Chinese and English. We wanted to change this scenario and bring our native language on par with the rest. The Unicode fonts for Malayalam are well developed and we felt that we should take full advantage of them and make instant digitization of handwritten Malayalam a reality. We see a lot of scope for this system in electronic form filling, restoration of old literary works as well as real-time translation and text to speech applications. The pivotal point in all of these applications is a well-developed handwriting recognition system, and we aim to develop a reliable, efficient and fast HCR system through this project.

1.3LITERATURE REVIEW

Offline character recognition is the process of recognizing handwritten text from a scanned sheet of paper. During the past few years, many works have been reported in Offline character recognition of Malayalam. This review summarizes the important works that have been done so far for offline Malayalam character recognition.

The first work in Malayalam OCR was reported by Lajish[2]. It used fuzzy-zoning and normalized vector distance measures for the recognition of segmented characters. The size normalized image was divided into 3x3 uniform sized zones. The normalized vector distance for each zone is computed and fuzzification is performed on these. The 9 features, thus obtained were classified using class modular neural network. The system had attained an overall accuracy of 78.87% for the 44 Malayalam handwritten characters. State space Point Distribution (SSPD) parameters obtained from gray scale based State space map (SSM) of character samples were used for classification. SSPD parameters were calculated from State-space maps with eight directional space variations. The 16 features, thus obtained were classified using class modular neural network. The system had an accuracy of 73.03%.

Wavelets were applied by G. Raju [3] for the recognition of isolated Malayalam characters. He used db4, a member of the Daubechie wavelet family with order 4, for decomposition into ten sub-images. The count of zero-crossing in each of the ten sub bands was used for classification. From the analysis of zero crossings, 25 consonant characters that were taken in the data set could be classified into 11 sets. Also, performance analysis of wavelet feature using twelve different wavelet filters was done. In this work, the wavelet feature and projection profile were combined by subjecting the projection profiles to n levels of wavelet transform. A Multilayer Perceptron Network, which consists of multiple layers of computational units, interconnected in a feed-forward manner is used as the classifier. The average recognition accuracy obtained was 76.8% for selected 33 Malayalam characters. The above work was also extended by adding an additional feature, aspect ratio which further improved the recognition accuracy, the average being 81.3%.

Rahiman proposed a recognition technique based on the HLH intensity pattern of characters [4]. The characters were grouped into ”ra”, ”pa” or special type. characters. An overall accuracy of 88.6% was achieved for the system. They have also proposed another technique for the extraction of features based on the analysis of position and count of the horizontal and vertical lines [5]. One of the main advantages of this method was that it could identify characters, even if they were written on a coloured background. An accuracy of 91% was achieved in this work.

Chacko and Anto [6] proposed a method for producing smooth skeletons of Malayalam handwritten characters. Here skeleton pruning was done by contour portioning with discrete curve evolution. The overall accuracy of the system was 90.18% for 33 character classes. The recognition of handwritten Malayalam characters using the wavelet energy feature (WEF) and extreme learning machine (ELM) was proposed in [7].ELM algorithm learns much faster than the traditional learning algorithms for feed forward neural networks.

In [8], Jomy John et al. proposed a method based on chain code and image centroid for the recognition of Malayalam vowels. From the chain code representation of the character, a chain code histogram and Normalized chain code histogram were constructed which were used for classification process. An average accuracy of 72.1% was obtained for the system. They have also proposed another system which uses visual image queries for retrieving similar images from database of Malayalam handwritten characters.

In [9], Bindu S Moni et al. proposed a character recognition scheme using run length count (RLC). RLC is the count of contiguous group of 1s encountered in a left to right / top to bottom scan of a character image. Modified Quadratic Discriminate function (MQDF) was used for classification. They had achieved a recognition rate of 94.18% with 72 features for 30 selected character classes. They have also proposed another character recognition system using directional features and MQDF obtaining a recognition rate of 95.42% using a total of 432 features for 44 Malayalam characters.

Recently, Vidya V [10] proposed another approach based on Probabilistic Simplified fuzzy ARTMAP(PSFAM). Different features like cross feature, fuzzy depth, distance and Zernike moment feature are extracted for each character glyph. An accuracy of 87.81% was attained for 142 Malayalam characters.

In [11], M.Zahid , M.Ashraful Amin and Hong Yan proposed the method of cell projection as a method of feature extraction. An accuracy of up to 92.43 % was obtained in the recognition of Bangla numerals.

1.4 RESEARCH GAP

From the review of the literature it has been identified that there still does not exist an HCR system for Malayalam character set which recognizes characters with a high level of accuracy. This improvement in accuracy, without compromising the speed, can be achieved by the right combination of feature extraction and classification technique. Most papers stop at creating an efficient system and calculating accuracies. In addition to improving performance, this project aims to further develop the system to create an end-user application with an easy to use GUI that takes a scanned image of the document as the input, recognizes it and further gives a digital version of the document as the output.

1.5 OBJECTIVES

The main objectives of this project are to:

1. To design an HCR system for handwritten Malayalam documents by optimizing feature extraction and classification techniques.
2. To develop an end-user application with a graphical user interface.

1.6 OUTLINE OF THE REPORT

The first chapter of this report gives an introduction to the Malayalam character set, handwriting recognition and outlines the motivation behind the pursuance of this project. The literature review undertaken has been deliberated followed by the noticed research gap and project objectives.

The second chapter discusses the character recognition system, the two aspects of character recognition – image processing and classification. Each aspect has been broken down into sub sections and discussed in detail.

The third chapter reports how the implementation of the character system was carried out, details of the proposed system, how the GUI was designed and this is accompanied by results obtained and corresponding analysis.

Concluding remarks have been given in the fourth chapter followed by the references and appendices.

CHAPTER 2

THE CHARACTER RECOGNITION SYSTEM

2.1 INTRODUCTION

This chapter describes a character recognition system and discusses its two main aspects – image processing and classification in detail. Each section has been broken down into steps and elaborated on.

The major steps involved in a character recognition system can be broken down as follows

1. Image acquisition
2. Image Pre-Processing
3. Segmentation
4. Feature Extraction
5. Classification
6. Post Processing

The architecture of a general character recognition system is shown in Figure 2.1

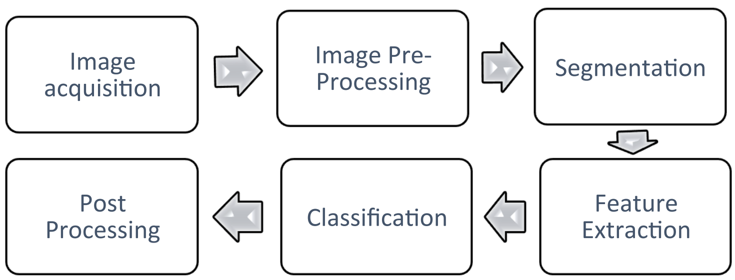

Figure 2.1 - Architecture of a general HCR system

2.2 IMAGE PROCESSING

Image processing refers to the steps starting from obtaining the image for recognition to enhancing of the image in order to obtain features from it. Image processing was implemented using OpenCV with C++ interface. The major steps involved in the Image Processing stage are

1. Image Acquisition
2. Image Pre-Processing
    2.1. Binarization
    2.2. Noise Removal
    2.3.Skeletonization
    2.4. Skew Correction
    2.5. Normalisation

3. Feature Extraction

Fig 2.2 shows a block diagram representation of the different stages involved in image processing

Figure 2.2–Different stages of image processing

2.2.1 IMAGE ACQUISITION

This is the stage where data is collected as part of the recognition process, documents in the image format are given as input to the system. These images will generally be in the JPEG or PNG format, they're obtained from scanners/ digital cameras.

2.2.2 IMAGE PRE-PROCESSING

2.2.2.1 **Binarization**

Binarization is the process of converting grayscale images to binary images. It is done in order to identify the objects of interest from the image. It separates the foreground pixels from the background pixels. Local thresholding methods are based on applying different threshold values to different regions of the image. Global thresholding methods apply one threshold value to the entire image. Otsu's algorithm is a commonly used approach for global thresholding.

2.2.2.2**Noise Removal**

A large amount of noise may occur in the image obtained after scanning. This may be due to the poor quality of the scanner or the use of degraded documents. Gaussian noise & Salt and Pepper noise are two such common noises. Filtering techniques such as linear filtering, median filtering or adaptive filtering can be used to remove noise to a certain extent. The main objective of this phase is to remove as much noise as possible while retaining the original signal.

2.2.2.3 **Skeletonization**

Skeletonization or thinning is a morphological operation in which a single pixel wide representation of an image is obtained without changing its connectivity. The purpose of thinning is to reduce the image components so that they contain only essential information.

2.2.2.4 **Skew Correction**

During document scanning, skew is introduced in the image. Skew angle is the angle that the text lines of the image make with the horizontal direction. The aim of the skew detection operation is to align an image before processing. Some of the major techniques available for skew estimation are projection profile method and Hough transform method. The image is then rotated based on the detected skew angle.

2.2.2.5 **Normalisation**

Normalization is the process of converting the image into a standard size. Bilinear and Bicubic interpolation techniques can be used for size normalization. In this case, each of the images has been converted to a 20x20 pixels image.

2.2.3. FEATURE EXTRACTION

The 20x20 pixel image from normalization stage has a total of 400 pixels per image. With a data set of 62 characters, cell projection is the feature extraction method adopted. In this proposed feature extraction method of horizontal projections, a character is partitioned into k regions and projection is taken for each region. Using a similar technique vertical celled projection can be formulated, summing to a total of 160 features. In comparison to other feature extraction methods, this method requires a small number of logical and arithmetic operations and only need to consider all the pixels of image in worst case.

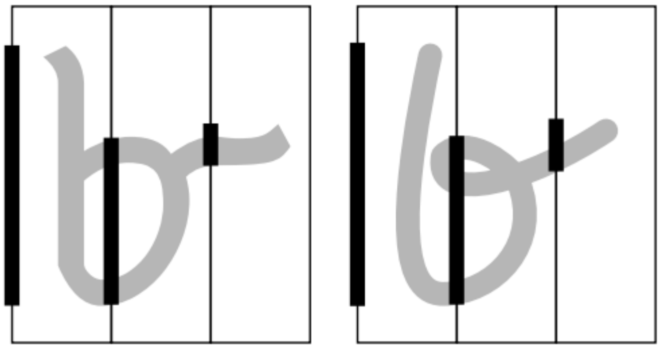

Fig 2.3 Cell Projection –Standard form Vs Handwritten form

2.3 CLASSIFICATION

Classification is the stage that follows image processing, once the relevant features of the image has been extracted, the HCR system needs to be able to relate these features to a character. In this project, a Support Vector Machine using an RBF function has been used as a classifier.

2.3.1 SUPPORT VECTOR MACHINE

The objective of any machine that is capable of learning is to achieve good generalization performance, given a finite amount of training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets. With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data. The principle of an SVM is to map the input data onto a higher dimensional feature space nonlinearly related to the input space and determine a separating hyper plane with maximum margin between the two classes in the feature space. A support vector machine is a maximal margin hyper plane in feature space built by using a kernel function. This results in a nonlinear boundary in the input space. The optimal separating hyper plane can be determined without any computations in the higher dimensional feature space by using kernel functions in the input space.

2.3.2 RADIAL BASIS FUNCTION

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that$\varphi(x) = \varphi(||x||)$ or alternatively on the distance from some other point *c*, called a centre, so that$\varphi(x,c) = \varphi(||x-c||)$. Any function φ that satisfies the property $\varphi(x)= \varphi(||x||)$ is a radial function. The norm is usually Euclidean distance, although other distance functions are also possible. Sums of radial basis functions are typically used to approximate given functions. This approximation process can also be interpreted as a simple kind of neural network. RBFs are also used as a kernel in support vector classification.

After the identification of the unique features for each of the characters, we use Support Vector Machines (SVM) to classify the characters based on the features identified. The obtained class labels are converted to corresponding Unicode of Malayalam and written to text file.

2.4 SUMMARY

This chapter discussed Image Processing and Classification, the two main aspects of a character recognition system. It was decided to use all the pixel intensities of the processed image as features to train the SVM. The final class labels returned by SVM after classification are converted to Malayalam Unicode and written to a text file.

The next chapter discusses the implementation of the project in detail; it outlines the training done and the testing procedure used.

CHAPTER 3

IMPLEMENTATION

3.1 INTRODUCTION

This chapter discusses the procedure followed in implementing the HCR system, it describes the training done and the results obtained when the system was tested with an arbitrary input.

3.2 TRAINING

To carry out the training, a training of set of 62 characters, with 100 instances per character was created. This image was then put through the pre-processing steps mentioned in chapter 2 and the final character instance images were obtained. Training was carried out using functions from LibSVM for classification.

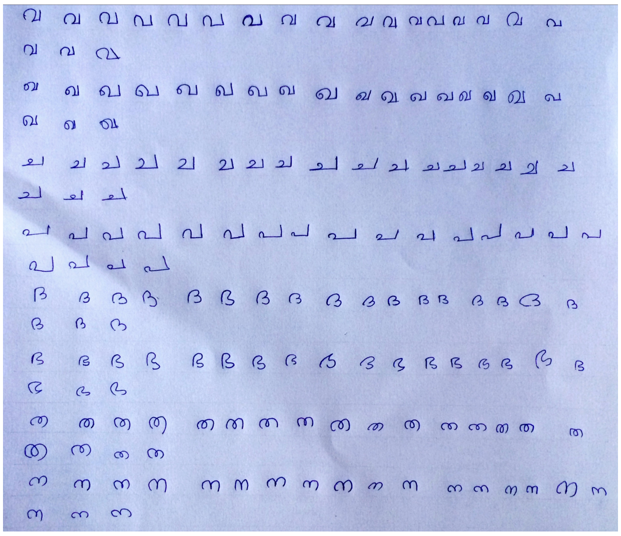

Figure 3.1 –InputSheet before processing

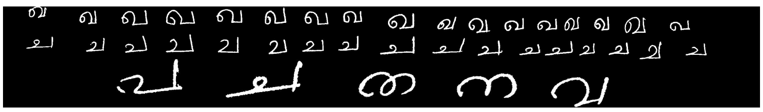

Figure 3.2 –Line segmentation (after processing)

Figure 3.3 – Character segmentation (after processing)

Figure 3.3 shows one instance each of the 62 characters that were used to train the system. 100 such instances for each of these letters were obtained, the total training set consisted of 6200 total instances and each of them was described by 160 features.

3.3 TESTING

Once the SVM was trained using these features, a new image was created for the purpose of testing the HCR system. Figure 3.4 shows the image that was used to test the system.

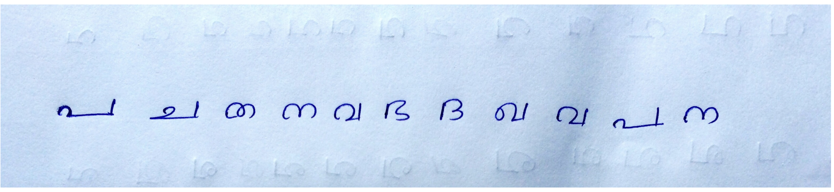

Figure 3.4 –Test Set

The program is designed to print the recognized characters into a text file, one after the other in their order of appearance from left to right.

Figure 3.5 shows the output obtained when the test set was given as input to the system.

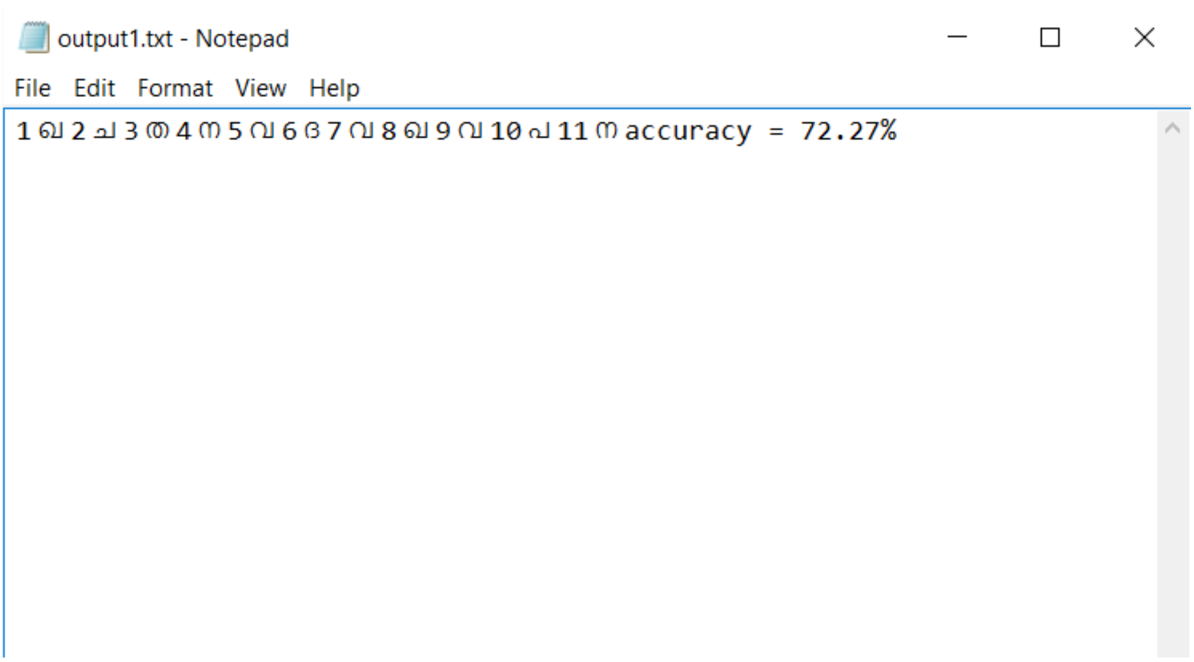

Figure 3.5 –Output Text File

3.4 PROPOSED APPLICATION

The application is essentially divided into two parts, a training end and the user end. The training end is carried out by the team handling the system. As and when more samples are created, the training process is repeated to obtain the updated classification model. This model can be made available to the user on a periodic basis as updates to the application. The user end consists of the GUI application executable and the downloaded model file. This approach allows the system to be constantly improved while keeping the complex training process out of the user's area of concern.

3.5 GRAPHICAL USER INTERACE (GUI)

A graphical user interface is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators as opposed to text-based interfaces or text navigation. Well-designed GUI can free the user from learning complex command languages.

A GUI was designed for the project in order to make it accessible to wider audience. Using the simple and user-friendly design implemented, a person with basic computing knowledge can operate and take benefit of this offline handwriting recognition system.

The GUI was designed in Qt Creator, a cross-platform C++ IDE. It's part of the Qt GUI Application development framework. It includes a visual debugger, an integrated GUI layout and a forms designer.

The GUI essentially simplifies the whole classification process, it reduces it to a 3 step process for the user.

1. Browse for and select the input image. The image is shown in the gray scale format to the user so as to ensure the right image is being processed.
2. Choose Output Directory
3. Begin Recognition

Post recognition, the output file will be available in the directory chosen. Figure 3.6 shows a screenshot of the GUI.

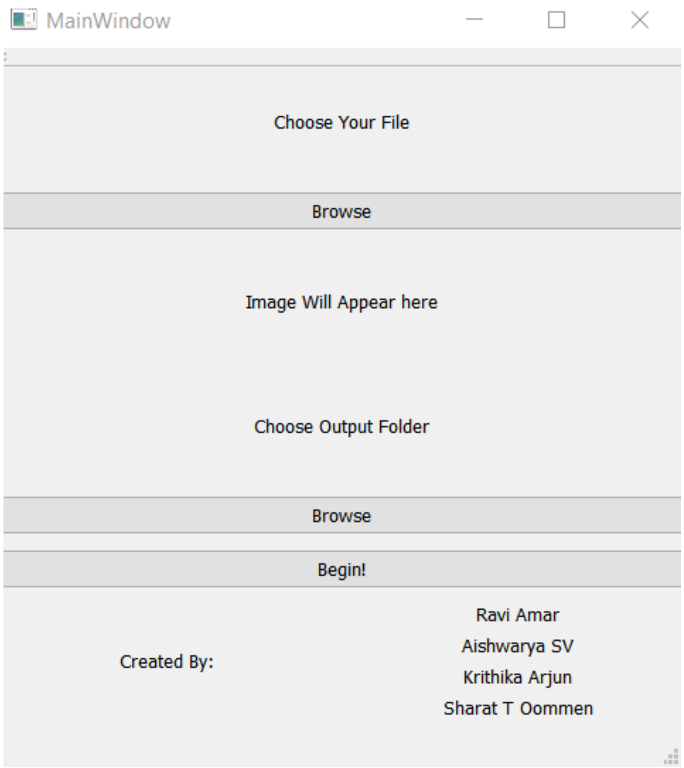

Figure 3.6

3.6 RESULT AND ANALYSIS

An SVM Classifier with RBF Kernel was designed to classify up to 62 Malayalam characters using the method of cell projection. An accuracy of about x was obtained for different test samples. This accuracy can be further increased by using increased number of samples and enhanced feature extraction methods.

CHAPTER 4

CONCLUSION

4.1 CONCLUSION

While the field of character recognition is quite challenging, with a large enough training set and the right combination of feature extraction and classification, it is possible to make instantaneous digitisation of Malayalam handwriting a reality. Recognition of vowel modifiers and joint characters also poses a serious challenge to the HCR system.

Using a specialised feature extraction method, the designed system is able to recognize up to 62 Malayalam characters with x% accuracy. The GUI makes it much easier for someone with basic computer knowledge to use the character recognition system.

4.2 FUTURE SCOPE

The system performance can be improved by training using more samples as well as different feature extraction techniques. Once the digitization is carried out with sufficient accuracy, it becomes possible to incorporate features like translation into other languages. The system can be widely adopted in digitization of old hard to read texts, extraction of data from handwritten forms for government servies and digitization of old handwritten records in organizations.

REFERENCES

[1] Govindaraju, V. and S. Setlur, Guide to OCR for Indic Scripts: Document Recognition and Retrieval. 2009: Springer Publishing Company, Incorporated. 325.

[2] Lajish V. L., Handwritten character recognition using perpetual fuzzy zoning and class modular neural networks

[3] G. Raju, Recognition of unconstrained handwritten Malayalam characters using zero-crossing of wavelet coefficients

[4] M. A. Rahiman et. al., Isolated handwritten Malayalam character recognition using HLH intensity patterns, 2010 .

[5] Abdul Rahiman M, M S Rajasree Masha N, Rema M , Meenakshi R, Manoj Kumar G,"Recognition of Handwritten Malayalam Characters using Vertical Horizontal Line Positional Analyzer Algorithm".

[6] Binu P. Chacko, Babu Anto P, Discrete Curve Evolution Based Skeleton Pruning for Character Recognition, Seventh International Conference on Advances in Pattern Recognition, 2009.

[7] Jomy John, Pramod K. V, Kannan Balakrishnan "Offline Handwritten Malayalam Character Recognition Based on Chain Code Histogram", "Unconstrained Handwritten Malayalam Character Recognition using Wavelet Transform and Support vector Machine Classifier.

[8] Simily Joseph, Jomy John, Kannan Balakrishnan, Pramod K. V, Content Based Image Retrieval System for Malayalam Handwritten Characters, International Conference on Network and Computer Science, 2011.

[9] Bindu S Moni, G Raju,"Modified Quadratic Classifier for Handwritten Malayalam Character Recognition using Run length Count".

[10] Vidya V, Indhu T R, Bhadran V K,R Ravindra Kumar, "Malayalam Offline Handwritten Recognition using Probabilistic Simplified Fuzzy ARTMAP", Advances in Intelligent Systems and Computing Volume 182, 2013.

[11] M Zahid Hossain , M Ashraful Amin , Hong Yan "Rapid Feature Extraction for OCR",2012.

APPENDIX – A

Matlab Code

```
function[model]=training(f,n)
%function to train the svm
%returns the training model
segment(f);
create_txt(n);
[trainlabels,trainfeatures]=libsvmread('features1.txt');
model=svmtrain(trainlabels,trainfeatures,'-s 0 -t 2 -c 100 ');
end

function[]=segment(f)
%function to extract characters from text
f=rgb2gray(f);
level=graythresh(f);
f1=im2bw(f,level);
[r,c]=size(f1);
j=0;

f1=imcomplement(f1);

S=sum(f1,2);
for i=1:r
if(S(i)==0)
if(i~=r)
if(S(i+1)~=0)
      j=j+1;
end
end
end
end
r1=r/j;
r1=round(r1);
g=zeros(r1,c,j);
k=0;
for i=2:r

if(S(i)~=0)

if (S(i-1)==0)
      k=k+1;
      p=0;
end


      p=p+1;
```

```
for l=1:c
        g(p,l,k)=f1(i,l);
end
end

end
k1=0;
for i=1:j
   S=sum(g(:,:,i),2);
for k=1:r1
if(S(k)~=0 && S(k+1)==0)
        l=k+1;
end
end
   f1=zeros(l,c);
for j=1:l
for k=1:c
        f1(j,k)=g(j,k,i);
end
end
   S1=sum(f1,1);

for j=3:c
if(S1(j)~=0 && S1(j-1)==0 && S1(j-2)==0)
        p=j-2;
end
if(S1(j)==0 && S1(j-1)==0 && S1(j-2)~=0)
        q=j;
        k1=k1+1;
     k=0;
     f2=zeros(l,q-p+1);
for t=p:q
        k=k+1;
        f2(:,k)=f1(:,t);
end
     se=strel('square',2);
     f2=imdilate(f2,se);
     f2=imresize(f2,[20,20]);
     imwrite(f2,sprintf('segv%i.jpg',k1));
end
end


   imwrite(f1,sprintf('seg%i.jpg',i));
end
end
```

```
function[]=create_txt(n)
%converts the features of training set to text file
A=zeros(n,400);
for i=1:n
   f=imread(sprintf('segv%i.jpg',i));

   level=graythresh(f);
   f=im2bw(f,level);
for j=1:400
      A(i,j)=f(j);
end
end
A=sparse(A);
B=zeros(n,1);
for i=1:n
if(i<21)
      B(i,1)=1;
elseif(i<41)
      B(i,1)=2;
elseif(i<61)
      B(i,1)=3;
elseif(i<81)
      B(i,1)=4;
elseif(i>81&&i<102)
      B(i,1)=5;
elseif(i>103&&i<124)
      B(i,1)=6;
elseif(i>=124&&i<144)
      B(i,1)=7;
elseif(i>=144 && i<164)
      B(i,1)=8;
else B(i,1)=0;
end
end
libsvmwrite('features1.txt',B,A);
end

function[]=testing(f,n,labels,model)
%test new samples
segment(f);
create_txt1(n,labels);
[testlabels,testfeatures]=libsvmread('features2.txt');
[label,accuracy,~]=svmpredict(testlabels,testfeatures,model);
conv2txt(label,accuracy,n);
end


function[]=create_txt1(n,labels)
%extract features of test input into a text file
A=zeros(n,400);
```

```
for i=1:n
    f=imread(sprintf('segv%i.jpg',i));

    level=graythresh(f);
    f=im2bw(f,level);
for j=1:400
        A(i,j)=f(j);

end
end

A=sparse(A);
libsvmwrite('features2.txt',labels,A);
end

function[]=conv2txt(label,accuracy,n)
%output text
fid=fopen('output1.txt','w','ieee-be','UTF-8');
for i=1:n
switch label(i)
case 1
            fprintf(fid,'%d %c \n',i,hex2dec('0D35'));
case 2
            fprintf(fid,'%d %c \n',i,hex2dec('0D16'));
case 3
            fprintf(fid,'%d %c \n',i,hex2dec('0D1A'));
case 4
            fprintf(fid,'%d %c \n',i,hex2dec('0D2A'));
case 5
            fprintf(fid,'%d %c \n',i,hex2dec('0D26'));
case 6
            fprintf(fid,'%d %c \n',i,hex2dec('0D2D'));
case 7
            fprintf(fid,'%d %c \n',i,hex2dec('0D24'));
case 8
            fprintf(fid,'%d %c \n',i,hex2dec('0D28'));
case 0
            fprintf(fid,'%d junk \n',i);
end

end
fprintf(fid,'accuracy = %d \n',accuracy(1));
fclose(fid);
end
```