



HÖGSKOLAN
Dalarna

Developing Character Recognition for Ethiopic Scripts

Fitsum Demissie

2011

**Master
Thesis
Computer
Engineering
Nr: E4031D**



DEGREE PROJECT

Computer Engineering

Programme	Reg number	Extent
Masters Programme in Computer Engineering - Applied Artificial Intelligence	E4031D	15 ECTS
Name of student	Year-Month-Day	
Zewidie, Fitsum Demissie	2011-05-16	
Supervisor	Examiner	
Hasan Feleyh		
Company/Department	Supervisor at the Company/Department	
Title		
Developing Character Recognition for Ethiopic Scripts		
Keywords		
Ethiopic, Geez, Amharic, SVM, OCR, Latin, Non-Latin.		

Abstract

The Amharic language is the Official language of over 70 million people mainly in Ethiopia. An extensive literature survey and the government report reveal no single Amharic character recognition is found in the country. The Amharic script has 33 basic characters each with seven orders giving 310 distinct characters, including numbers and punctuation symbols. The characters are visually similar; there is a typeface, but no capitalization. Beside this there is no any standard font to use the language in the computer but they use different fonts developed by different stakeholders without keeping a standard on their own way and interest and this create a problem of incompatibility between different fonts and documents.

This project is to investigate the reason why Amharic optical character recognition is not addressed by local and international researchers and developers and finally to develop Amharic optical character recognition uses the features and facilities of Microsoft windows Vista or 7 using Unicode standard

Keywords: Ethiopic, Geez, Amharic, SVM, OCR, Amharic Optical Character Recognition.

ACKNOWLEDGMENT

I would like to express my gratitude to all those who encouraged with their support to complete this work. First and foremost, I am grateful to my supervisor, Dr. Hasan Fleyeh, for his great support with his patience and knowledge helping me to finish my thesis. I also attribute the level of my Masters degree to his fatherhood encouragement and guidance, and without him this thesis would have been a difficult task. And I owe my deepest gratitude to Dr. Siril Yella. I am very thankful to him and all Dalarna university Community for their motivation, guidance, and support specially Asif, Erfan, Roger and Dialla.

It is honor to express my thankfulness to my family members for being blessed by their heartfelt wishes and endless support throughout my life.

Also, I am pleased to have a friendly and cheerful group of fellow students and seniors Iman Abdurahman Mohammed and Bezawit Jemberu who were always been supportive. Special thanks to all of them.

Finally, I would like to express my gratitude to my parents Desta Chaniyalew and Demissie Zewidie.

Tables of Contents

Chapter 1.....	1
Introduction	1
Background Information	2
Problem Definition	2
Alternative Proposed Solution	3
Structure of the System.....	4
Objectives of the Project	5
Limitations and Constraints.....	5
<i>Limitation:-</i>	5
<i>Constraints</i>	6
Beneficiary	6
Chapter 2.....	7
Literature Review.....	7
Previous Local Works	7
International Multilingual OCR Tools Vendors.....	8
Outcomes of the Literature Review	11
Chapter 3.....	12
Technical Overview	12
Machine Learning.....	12
Support Vector Machine (SVM)	12
Generalization	13
Soft Margin Classifier.....	18
Kernel Trick.....	18
Kernel:	18
Kernel Trick: Dual Problem.....	19
Kernel Trick: Inner Product summarization.....	20
Kernel Functions.....	20
SVM for Classification.....	21
Multi class classification using SVM	21
Chapter 4.....	23
Ethiopic Optical Character Recognition System Design	23
Part I.....	23
The character set.....	23

The Ethiopic Glyphs:	26
Part II	27
System Design	27
The scanner	29
Skew Detection and Image Enhancement	32
Image enhancement for old documents	35
Segmentation	36
Character Extraction.....	37
Training Set.....	38
Mapping	42
Binary Representation.....	43
Data Normalization and Dimension Reduction	44
Data Representation	44
Features and facilities of Microsoft Windows vista / 7	45
Unicode	45
Summery	46
Chapter 5.....	48
Results Analysis	48
Noisy and old images.....	48
Skew Detection and Correction.....	50
Documents in different fonts	51
Classification using different dimension reduction tool	52
Classification using Different kernel trick with the selected dimension reduction tool.	53
Reason for Failure Analysis	61
Result Analysis Summery	62
The Number and Type of OCR Engines Available	62
Recognition Speed.....	62
Supported Output Formats	62
Support for Unicode Fonts	62
File Enhancement Features.....	62
Availability of advanced features	62
Chapter 6.....	64
Conclusion.....	64
References	67

List of Figures

Figure 1: Structure of Ethiopic optical character recognition model	4
Figure 2: Function order in Increasing Complexity	12
Figure 3: Illustration for the Need of SVM.....	13
Figure 4: Illustration of Linear SVM	14
Figure 5: Representation of hyper planes.....	15
Figure 6: Representation Support Vector	16
Figure 7: Importance of using Kernel	17
Figure 8: Multi class classification illustration	21
Figure 9: Ethiopic Character with Their Unicode Value	25
Figure 10: System design for Ethiopic Character recognition	27
Figure 11: a scanner machine.....	30
Figure 12: Skew Detected Input Image.....	33
Figure 13: Skew Corrected output/Input Image	34
Figure 14: This Document is very old and Noisy	35
Figure 15: After Applying Different Image Enhancement	35
Figure 16: Sample Scanned Document	36
Figure 17: Segmented Image Loaded to OCR Application	36
Figure 18: Sample Extracted Character.....	37
Figure 19: Flowchart for Training.....	39
Figure 20: Flowchart for Testing	40
Figure 21: follow from the classifiers up to the office applications.....	41
Figure 22: Sample Scanned Document	42
Figure 23: Segmented Image Loaded to OCR Application	42
Figure 24: Output Displayed in Browser	42
Figure 25 Sample Extracted binary image Character.....	43
Figure 26: Segmented Image in Binary Value or Pixel.....	43
Figure 27: experiment analysis for testing with noisy images	49
Figure 28: experiment analysis for skewed documents	51
Figure 29: experiment analysis for classifications accuracy for Nyala.....	59
Figure 30: experiment analysis for classifications accuracy for power	59
Figure 31: experiment analysis for classifications accuracy for agafari	60
Figure 32: Diagrammatical Illustration of Classification	63

List of Tables

Table 1 Ethiopic Numeral character.....	5
Table 2 Summery of International OCR developers[7]	9
Table 3 training image, desired output and actual character	38
Table 4 chosen and applicable parameter for experiments	46
Table 5 old image testing failure analysis	49
Table 6 Skew result analysis	50
Table 7 General Result Analysis.....	52
Table 8 a combination of Nyala font and PCA DR with the three kernel tricks.....	53
Table 9 a combination of power font and PCA DR with the three kernel tricks.	54
Table 10 a combination of Nyala font and MDS DR with the three kernel tricks.....	54
Table 11 a combination of Nyala font and NPE DR with the three kernel tricks.....	54
Table 12 a combination of power font and MDS DR with the three kernel tricks.	55
Table13 a combination of power font and NPE DR with the three kernel tricks.	55
Table 14 a combination of agafari font and PCA DR with the three kernel tricks.	55
Table 15 a combination of agafari font and MDS DR with the three kernel tricks.	56
Table 16 a combination of agafari font and NPE DR with the three kernel tricks.	56
Table 17 Confusion matrix for training.....	57
Table 18 Confusion matrix for testing.	58

Chapter 1

Introduction

Optical Character Recognition (OCR) is a technology that is used to translate scanned images of text into computer editable and searchable text. OCR Software and ICR Software technology are analytical artificial intelligence systems that consider only sequences of characters rather than whole words or phrases and do not cross-validate data during the recognition process (D - L I B, 2009)¹ Based on the analysis of sequential lines and curves, OCR and ICR make 'best guesses' at characters using database look-up tables to closely associate or match the strings of characters that form words. For these systems to effectively recognize hand printed or machine printed forms, words must be separated into individual characters. That is why most typical administrative forms require people to either hand print into neatly spaced boxes or use combs (tick marks) at the bottom of input lines to force spaces between letters entered on a form. Without the use of combs or boxes, conventional technologies reject fields if people do not follow the structure when filling out forms, resulting in significant administrative overhead and costs to forms processing organizations. (OCRopus, 2009)

Among others, the following are the major advantages of the OCR technology:

- It can be used to scan and preserve historical documents.
- It can be used for scanning data entry forms in a faster and less error prone manner.
- It can be used with other computer applications, such as Archives and Records Management Systems, to convert scanned documents into searchable text.

At present the recognition of Latin-based characters from well-conditioned documents can be considered as a relatively feasible and well developed technology. On the other hand, the processing of non-Latin scripts like some Asian and African language scripts is still a subject of active research.

Background Information

Ethiopia (/i:θi'ooɪə/) (Ge'ez: ኢትዮጵያ ፕሮጃፕላ) is a landlocked country located in the Horn of Africa, and officially known as the **Federal Democratic Republic of Ethiopia**. It is the second-most populous nation in Africa, with over 85.2 million people (Ethiopia, 2007) and the tenth-largest by area, with its 1,100,000 km². The capital is Addis Ababa. Ethiopia is bordered by Eritrea to the north, Sudan to the west, Djibouti and Somalia to the east, and Kenya to the south (Ethiopia, 2007)

Amharic (**Amharic**: አማርኛ *amarəñña*) is a Semitic language spoken in North Central Ethiopia by the Amhara. It is the second most-spoken Semitic language in the world, after Arabic, and the official working language of the Federal Democratic Republic of Ethiopia. Thus, it has official status and is used nationwide. Amharic is also the official or working language of several of the states within the federal system, including the Amhara Region and the multi-ethnic Southern Nations, Nationalities, and People's Region, among others. It has been the working language of government, the military, and of the Ethiopian Orthodox Tewahedo Church throughout medieval and modern times. Outside Ethiopia, Amharic is the language of some 2.7 million emigrants (notably in Canada, United States, and Sweden) (Ethiopia, 2007). About the characters and alphabets of the Amharic language is described on chapter three under the character set subtitle.

Problem Definition

The motivation that the fact for initiating this project is the absences of locally and or internationally developed single production of optical character recognition software for Ethiopic scripts. The language is not supported by ASCII standard to use it on the computer. Due to this problem different developers developed about 11 different fonts. All these fonts have some similar characters and some others are different, their differences create incompatibility problems among them and it is difficult to swap from one font to another font. This shows No standardized font is found in the country. Due to lack of this standard font, the country lacks different types of services and applications like OCR. However, Thanks to

technology Amharic scripts is included in a 16 bit Unicode version and this Unicode standard get rides the problem almost totally. Hence there is no any research based on Unicode for Ethiopic font Optical character recognition. In general Ethiopic script-based OCR processing is currently among the least developed ICT disciplines in the country. Developments in this area are mainly limited to preliminary research activities undertaken at different institutions of higher education's, such as the former School of Information Science for Africa (SISA).

Alternative Proposed Solution

This paper proposed different 3 solutions and implements one of the best solutions.

Alternative1. Developing OCR system for most frequently and popularly used types of font out of the available 11 different fonts.

Alternative2. Developing OCR application as per the number of fonts for each fonts individual OCR.

Alternative3. Developing an OCR application based on Unicode encoding standard that supports the form and structure of all the available fonts using features of Microsoft windows vista/7.

Finally, Out of the proposed alternative solution ***Alternative3*** is chosen and this solution is ECONOMICALLY, OPERATIONALLY, TECHINCHALLY, and SCHEDUALLY feasible.

Structure of the System

Developing an OCR application for the country like Ethiopia is a very important activity to for office automation and digital document processing. [Figure 1] is a building block to develop and implement the proposed solution. Any scanned document is loaded to the OCR system then the system segmented the image line by line and then extract character and calculate the mean and mapping value to for classification, finally the learning machine will classify and returns the exact class label and mapping to the lookup table will be applied.

In this project, a recognition model is implemented, which constructs the learning machine using a new pattern recognition technology the so called Support Vector Machine (SVM). The classification is processed by the SVM

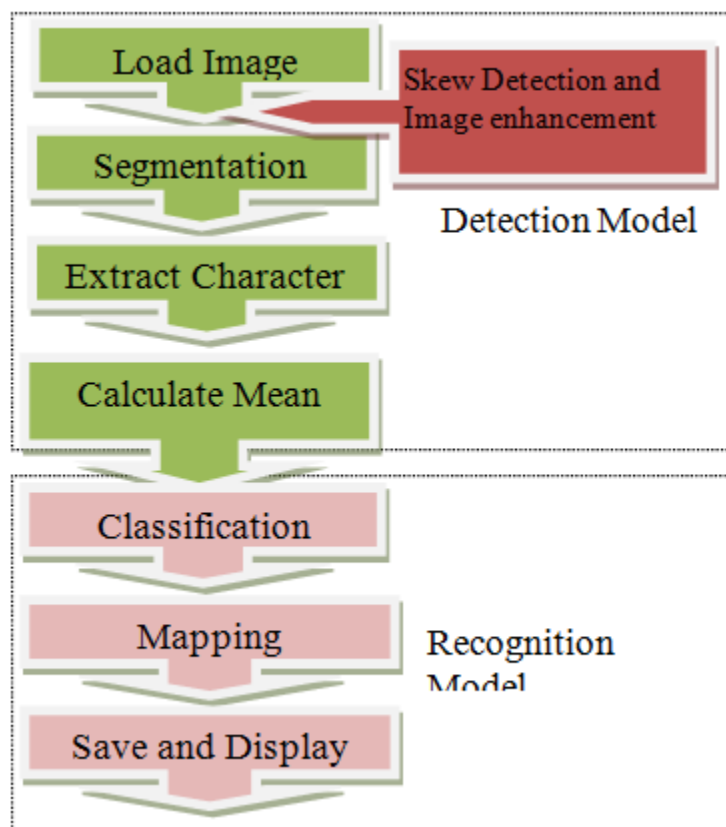


Figure 1 Structure of the Amharic optical character recognition system

Objectives of the Project

In Ethiopia there is no any single quality OCR application for Ethiopic scripts. In general OCR technologies are a well solved problem and matured technology for Latin scripts, but for non Latin scripts like some Asian and African scripts still it is a non solved problem and active research area. Ethiopic scripts are non Latin scripts. And relatively with Latin scripts the Ethiopic scripts are about 380 and 310 are most popularly and frequently used.

This paper has 2 major General objectives, First, to identify and find out the reason *why* local developers and international software developers does not address the OCR for Ethiopic scripts. Secondly After identifying the reason and clearly understands the problem trying to solve and implement the Ethiopic scripts OCR application using features and facilities of the newly released Microsoft windows operating systems like windows vista and windows 7 in effective and efficient way using a standard font called Unicode. The implementation detail is explained in Chapter 4 and the success measurement or result analysis is described under Chapter 5.

Finally, the result of this research is to put a baseline foundation for Ethiopic Character Recognition application using a standard Unicode encoding technique and experimental results.

Limitations and Constraints

Limitation:-

Ethiopic language has its own symbols and syllables for numbers, and mostly for writing purpose the language shares the Arabic numeric styles such as 0,1,2,3,4,5,6,7,8,9. Due to this reason the Ethiopian numeric styles are not included in this research.

Table 1 Ethiopic Numeral character

The second limitation is it works only for plain texts not for shapes, lines, tables, other images and does not keep any font styles sizes and formats just recognize and convert to a normal font style. The third limitation of this work is it works on Microsoft windows vista / 7. Because the Unicode Amharic font is found in these operating system only. Or the operating system should support **Nyala** font in Unicode standard.

Constraints

The research faces different constraints

1. Unavailability of different training images from different fonts different shapes and noisy images.
2. The largeness of the total number of characters.
3. The largeness of visually similar characters.

Beneficiary

Future researchers, Ministry of Information Technology of Ethiopia and any other stakeholders are the primary beneficiary of this research work.

Chapter 2

Literature Review

In this section the paper explains review of the local efforts undertaken in the country in this area and any related activities internationally. At present no single production quality or commercial OCR application exists that processes Ethiopic documents.

Previous Local Works

Some students of the departments of Information Science, Computer Science and Electrical and Computer Engineering of the Addis Ababa University have produced research output that is of interest to the fields of Optical Character Recognition Software development to Ethiopic documents. And, some international students from India and Sweden Universities did doctoral researches.

The students have used different algorithms to reach to a better result in the recognition of characters from scanned documents. Some of the algorithms and techniques that reviewed are:

- ✓ **Recognition of Printed Amharic Documents** Million Meshesha, C. V. Jawahar *Center for Visual Information Technology, International Institute of Information Technology* - (IEEE (. D., 2008)
- ✓ **Amharic Character Recognition using a Fast Signature Based Algorithm** Dr JOHN COWELL Dr FIAZ HUSSAIN *Dept. of Computer Science, Dept. of Computing & Information Systems De Montfort University, University of Luton, The Gateway, Leicester, LE1 9BH, Park Square, Luton, LU1 3JU, England.* (IEEE, Recognition of Printed Amharic Documents, 2008)
- ✓ **Recognition of Modification-based Scripts Using Direction Tensors** Lalith Premaratne Yaregal Assabie Josef Bigun *School of Information Science, Computer and Electrical Engineering Halmstad University, S-301 18 Halmstad, Sweden* (IEEE, Recognition of Modification-based Scripts Using Direction Tensors, 2007)

Currently in the country standard national fonts and keyboard mapping is not found, a national standard font and keyboard mappings must be designed and developed and or we should work on the basis of Unicode standard. This will help the test results that are obtained through the learning process (e.g. Neural Networks, SVM) to produce better results as the variation of font types and styles becomes less.

As stated above in Ethiopia there is no standard font and all the above researchers trying to develop their researches using different fonts like geez1, power geez, Agafari, Alex, dawit express, and Afro typeface. All these fonts have some similarity and differences in the structure of the alphabet and encoding technique. Besides this, a document written by one of them mostly is not understandable and readable by the others and understood as undefined symbol. In general there is no standard font in Ethiopia. There are number of reason for lack of standards. The major one is the number of character that used in the language, is about 380, It is non-Latin and no ASCII definition due to this size, I also as Masters student in the area of artificial intelligence want to work some researches and improve the results. Beyond this I have a plan by coordinating with other researchers to develop a production quality or commercial OCR beyond the academic exercise. The output of this project can be considered as step forward in this regard.

International Multilingual OCR Tools Vendors

In this paper some of the major developers of OCR Software Development Kits (SDK) identified in order to identify the requirements of incorporating additional scripts in such SDKs. One of the major inputs demanded by such vendors is the availability of a standard Typeface that would enable the OCR tools learn the features of the Ethiopic characters. The following are among the major providers of OCR Software Development Kits.

The focus in this regard on the identification of Programmable tools that provide interfaces for customization, as opposed to end-user OCR products.

❖ Microsoft Office Document Imaging Library (MODI) (Microsoft, 2003)

Microsoft has recently incorporated an OCR processing technology in its MS Office 2003 packages known as Microsoft Office Document Imaging Library (MODI).MODI provides

OCR Application Development Interfaces (API) that allows developers incorporate OCR capabilities into their products.

Although MODI does not provide support for Ethiopic, it provides support to other non-Latin scripts. The present version of MODI supports the following Languages:

- Chinese
- Czech
- Japanese
- Korean
- Russian, and
- A number of European Languages

One of the advantages of MODI is that it is freely available to developers that have installed an MS Office product.

❖ OmniPage Capture SDK (Nuance Communications, 2010)

OmniPage Capture SDK is a popular product that supports OCR processing capabilities to non-European languages such as the Japanese, Chinese and Korean scripts. The OmniPage Capture SDK provides different OCR engines such as print OCR (OCR, OCR-A, OCR-B and MICR), Handprint (ICR), Check Mark (OMR) and Barcode recognition engines. It also provides image file enhancement tools as well as facilities for exporting processed outputs to different formats such as PDF and XML.

❖ Leadtools OCR Programming Tools (Leadtools, 2010).

Leadtools OCR Programming Tools is another popular OCR SDK that supports multilingual OCR processing. It provides facilities for exporting processed output into different file formats; provides different OCR engines; and includes image file enhancement facilities.

In the following table [Table 2] 15 different OCR applications are listed with their name, version, and date of release. They support different languages each but Ethiopic script is not included at all.

Table 2 Summary of International OCR developers

No.	Name	Latest version	Release year	No of languages
1	ExperVision TypeReader & OpenRTK	8.0	2010	20
2	ABBYY FineReader	9	2009	267
3	OmniPage	17	2009	128
4	[PDF OCR X]	1.4	2010	33
5	Readiris	2009	12 Pro & Corporate	144
6	Readiris	12 Pro & Corporate Middle-East	2009	3
7	Readiris	12 Pro & Corporate Asian	2009	6
8	CuneiForm	12	2007	25
9	GOOCR	0.14	2009	1 Arabic
10	Kirtas Technologies Arabic OCR		2009	15 left-to-right
11	MoreData	1.0	2008	3
12	Microsoft Office Document Imaging	Office 2007	2007	3
13	NEOPTEC DATA-SCAN	5.7	2009	3
14	Microsoft Office OneNote 2007	NovoDynamics VERUS/ Middle East Professional	2005	3
15	ReadSoft			8

Outcomes of the Literature Review

As stated above articles and products locally and internationally reviewed. All the researcher and international developers are doing well on their way, but from Ethiopic scripture optical character recognition point of view they did a little, and some of the problem that discovered is stated as follows.

Problem of the local Researches: - all the researcher tries to work their study based on single font out of the number of available fonts in the country. This makes the research font dependent and inapplicable in the country.

Internationally, as you see on the above table top 15 best and popularly used types of OCR software is listed; Each OCR software supports different number of languages. Surprisingly the Ethiopic script is not supported by any of them. And this create a question like “why Ethiopic script is not included?”, this question is not addressed by any researcher. Generally this paper can suggest 5 reasons why Ethiopic language is not included.

1. The total number of the characters are large
2. The availability of Visually similar characters
3. The absence of responsible person who promote the language in the world for the international software developers, and finally
4. Luck of cooperation between developers and local researchers.
5. The most important and the most significant one is luck of standard fonts.

These both show that the Ethiopic Amharic script recognition is active research area.

Chapter 3

Technical Overview

OCR is a well developed field of study, but for non-Latin scripts like Ethiopic script it is active research area. In this research tried to develop an OCR application for Ethiopic scripts and below described the Machine learning tool that is used for the proposed solution. There exist a number of methodologies of addressing the issues of character recognition. Support Vector Machine and Neural Network approaches are nowadays the most popular machine learning methods for recognizing patterns. This subsection describes the basics of SVM.

Machine Learning

Machine Learning is considered as a subfield of Artificial Intelligence and it is concerned with the development of techniques and methods which enable the computer to learn. In simple terms development of algorithms which enable the machine to learn and perform tasks and activities. Machine learning overlaps with statistics in many ways. Over the period of time many techniques and methodologies were developed for machine learning tasks.

Support Vector Machine (SVM)

Support Vector Machine (SVM) was first heard in 1992, introduced by Boser, Guyon, and Vapnik in COLT-92. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. In another terms, Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Support Vector machines can be defined as systems which use hypothesis space of a linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. Support vector machine was initially popular with the NIPS community and now is an active part of the machine learning research around the world. SVM becomes famous when, using pixel maps as input; it gives accuracy comparable to sophisticated neural networks with elaborated features in a handwriting recognition task

(science, 2010) It is also being used for many applications, such as hand writing analysis, face analysis and so forth, especially for pattern classification and regression based applications. The foundations of Support Vector Machines (SVM) have been developed by Vapnik (Vapnik, 1995) and gained popularity due to many promising features such as better empirical performance. The formulation uses the Structural Risk Minimization (SRM) principle, which has been shown to be superior, (Vapnik, 1995), to traditional Empirical Risk Minimization (ERM) principle, used by conventional neural networks. SRM minimizes an upper bound on the expected risk, whereas ERM minimizes the error on the training data. It is this difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning. SVMs were developed to solve the classification problem, but recently they have been extended to solve regression problems (Burges, 1998).

Generalization

The ability of a hypothesis to correctly classify data not in the training set is known as its generalization. SVM performs better in terms of not over generalizing easily (V. Vapnik, 1997). Another thing to observe is to find where to make the best trade-off in trading complexity with the number of epochs; the illustration brings to light more information about this.

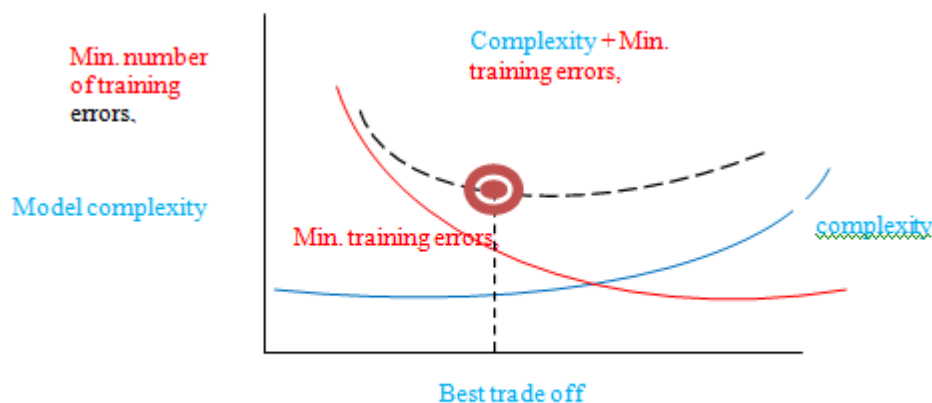


Figure 2 Function order in increasing complexity

Firstly working with neural networks for supervised and unsupervised learning showed good results while used for such learning applications. Multilayer Perceptron (MLP)'s uses feed forward and recurrent networks. MLP properties include universal approximation of continuous nonlinear functions and include learning with input-output patterns and also involve advanced network architectures with multiple inputs and outputs (Mitchell, 1997.). There can be some issues noticed. Some of them are having many local minima and also finding how many neurons might be needed for a task is another issue which determines whether optimality of that Neural Network (NN) is reached. Another thing to note is that even if the NN solutions used tends to converge, this may not result in a unique solution (Campbell, 2009). Now look at another example where we plot the data and try to classify it and we see that there are many hyper planes which can classify it. But which one is better?

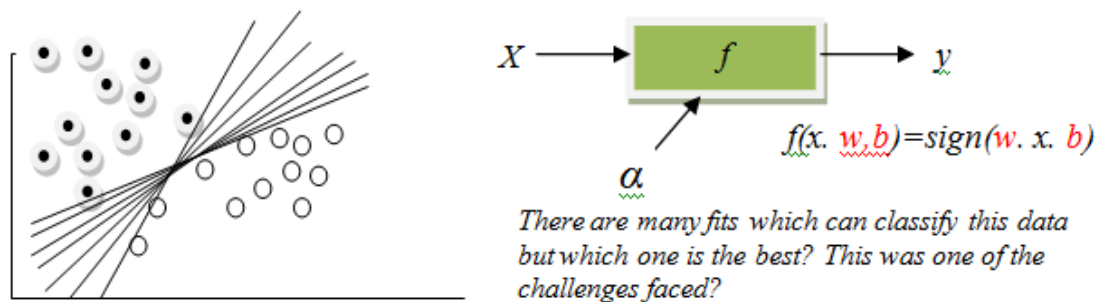


Figure 3: Here we see that there are many hyper planes which can be fit in to classify the data but which one is the best is the right or correct solution. The need for SVM arises. (Taken Andrew W. Moore 2003) [9]. Note the legend is not described as they are sample plotting to make understand the concepts involved.

From above illustration, there are many linear classifiers (hyper planes) that separate the data. However, only one of these achieves maximum separation. The reason it need is because if it use a hyper plane to classify, it might end up closer to one set of datasets compared to others and it do not want this to happen and thus it see that the concept of maximum margin classifier or hyper plane as an apparent solution. The next illustration gives the maximum margin classifier example which provides a solution to the above mentioned problem.

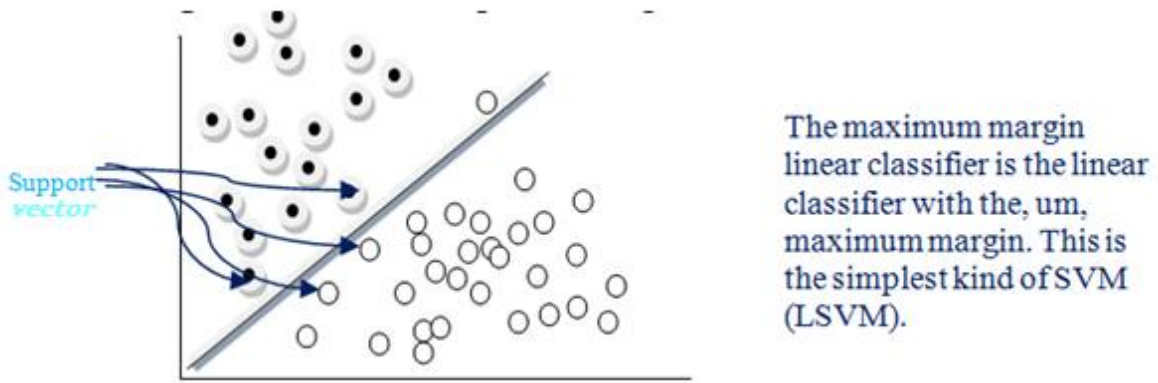


Figure 4: Illustration of Linear SVM. (Taken from Andrew W. Moore slides 2003) [9]. Note the legend is not described as they are sample plotting to make understand the concepts involved.

Expression for Maximum margin is given as (Burges, 1998):

$$\text{margin} = \arg \min_{\mathbf{x} \in D} d(\mathbf{x}) = \arg \min_{\mathbf{x} \in D} \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\sum_{i=1}^d w_i^2}}$$

The above illustration is the maximum linear classifier with the maximum range. In this context it is an example of a simple linear SVM classifier. Another interesting question is why maximum margin? There are some good explanations which include better empirical performance. Another reason is that even if we've made a small error in the location of the boundary this gives least chance of causing a misclassification. The other advantage would be avoiding local minima and better classification. The goals of SVM are separating the data with hyper plane and extend this to non-linear boundaries using kernel trick. For calculating the SVM we see that the goal is to correctly classify all the data. For mathematical calculations we have,

[a] If $Y_i = +1$; $w x_i + b \geq 1$

[b] If $Y_i = -1$; $w x_i + b \leq -1$

[c] For all i ; $y_i (w x_i + b) \geq 1$

In this equation \mathbf{x} is a vector point and \mathbf{w} is weight and is also a vector. So to separate the data [a] should always be greater than zero. Among all possible hyper planes, SVM selects the one where the distance of hyper plane is as large as possible. If the training data is good and

every test vector is located in radius r from training vector. Now if the chosen hyper plane is located at the farthest possible from the data (V. Vapnik, 1997). This desired hyper plane which maximizes the margin also bisects the lines between closest points on convex hull of the two datasets. Thus we have [a], [b] & [c].

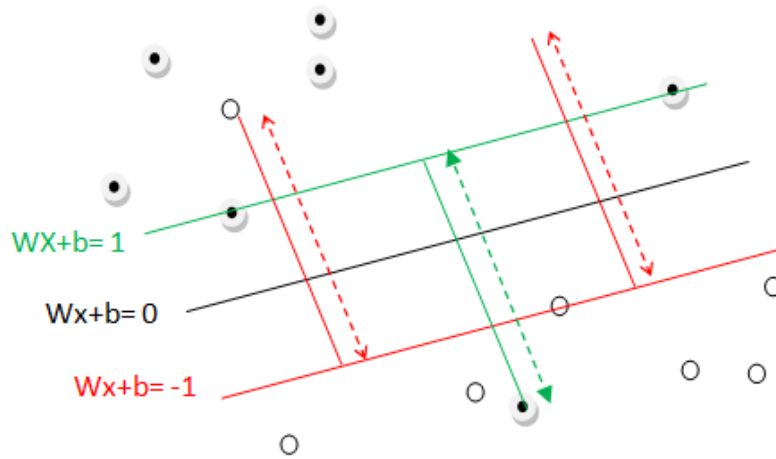


Figure 5 Representation of hyper planes

Distance of closest point on hyperplane to origin can be found by maximizing the x as x is on the hyper plane. Similarly for the other side points we have a similar scenario. Thus solving and subtracting the two distances we get the summed distance from the separating hyperplane to nearest points. Maximum Margin = $M = 2 / \|w\|$

Now maximizing the margin is same as minimum. Now we have a quadratic optimization problem and we need to solve for w and b . To solve this we need to optimize the quadratic function with linear constraints. The solution involves constructing a dual problem and where a Lagrange's multiplier α_i is associated. It is needed to find w and b such that $\Phi(w) = \frac{1}{2} \|w'\|^2$ is minimized;

$$\text{And for all } \{(x_i, y_i)\}: y_i (w \cdot x_i + b) \geq 1.$$

Now solving: we get that $w = \sum \alpha_i \cdot x_i$; $b = y_k - w \cdot x_k$ for any x_k such that $\alpha_k \neq 0$

Now the classifying function will have the following form: $f(x) = \sum \alpha_i y_i x_i \cdot x + b$

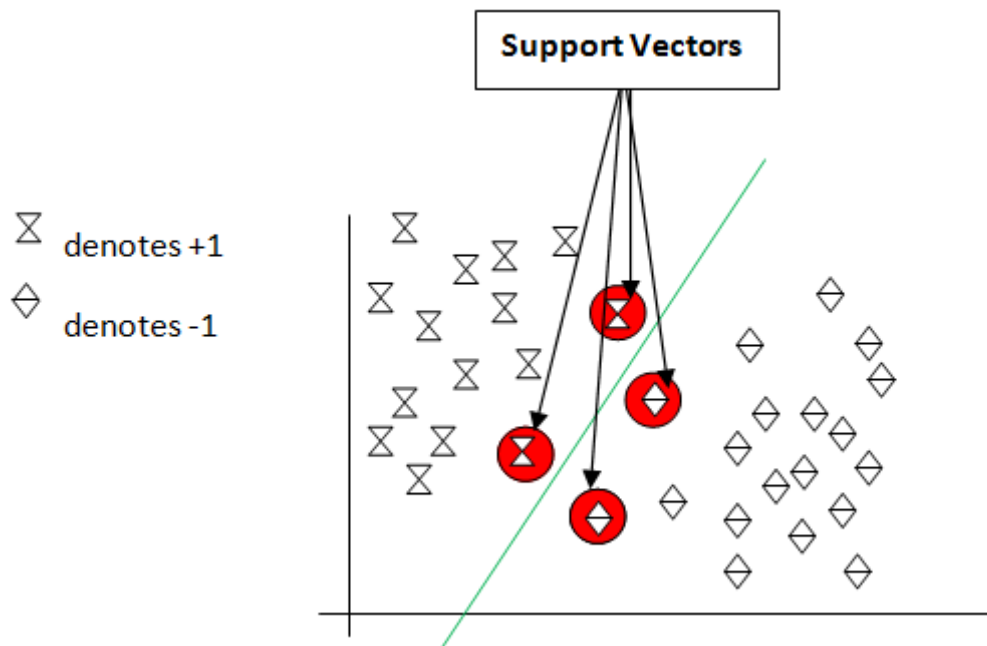


Figure 6 Representation of support vector

Besides of separating the data into different categories, the objective of SVM is to find an optimal hyper plane that correctly classifies the data as much as possible and separates the data as far as possible. [Figure 6] shows two ways that separate the data with two categories, one is represented by filled collate and the other is represented by sort. The lines mark the boundaries that run parallel to the separating line and the closest vectors to the line. The distance between two lines is called margin, while the vectors marked with red circles are the support vectors, they constrain the width of the margin. To define the optimal hyper plane the SVM analyze and find the hyper plane that is to maximize the margin. Because of the nature of the feature space in which these boundaries are found, Support Vector Machines can exhibit a large degree of flexibility in handling classification tasks of varied complexities. General type of SVM model including linear, polynomial, radial basis function, and Gaussian will be introduced in this paper. SVM models works very similarly to classical neural networks. Actually, a SVM model using a sigmoid kernel function is equivalent to a two-layer, feed forward neural network. However, comparing with traditional neural network approaches, the generalization theory of SVM enables the models to avoid overfitting the data.

Soft Margin Classifier

In real world problem it is not likely to get an exactly separate line dividing the data within the space. And we might have a curved decision boundary. It might have a hyperplane which might exactly separate the data but this may not be desirable if the data has noise in it. It is better for the smooth boundary to ignore few data points than be curved or go in loops, around the outliers. This is handled in a different way; here we hear the term slack variables being introduced. Now we have, $y_i(w'x + b) \geq 1 - S_k$ (Mitchell, 1997.). This allows a point to be a small distance S_k on the wrong side of the hyper plane without violating the constraint. Now we might end up having huge slack variables which allow any line to separate the data, thus in such scenarios we have the Lagrangian variable introduced which penalizes the large slacks.

$$\min L = \frac{1}{2} w'w - \sum \lambda_k (y_k (w'x_k + b) + s_k - 1) + \alpha \sum s_k$$

Where reducing α allows more data to lie on the wrong side of hyper plane and would be treated as outliers which give smoother decision boundary (Campbell, 2009).

Kernel Trick

Kernel: If data is linear, a separating hyper plane may be used to divide the data. However it is often the case that the data is far from linear and the datasets are inseparable. To allow for this kernels are used to non-linearly map the input data to a high-dimensional space. The new mapping is then linearly separable. A very simple illustration of this is shown below in [Figure 5] (Vapnik, 1995).

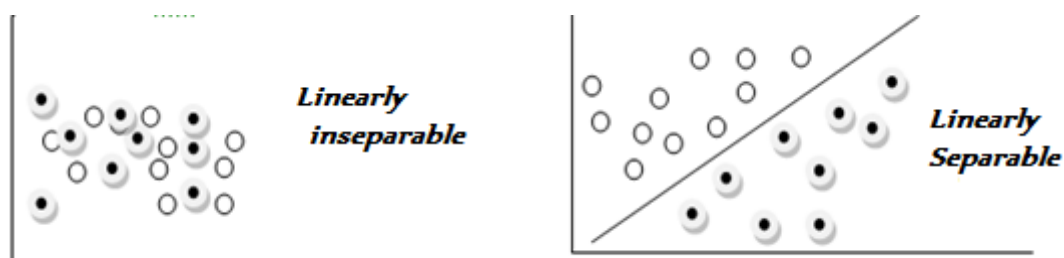


Figure 7: importance of using kernel?

This mapping is defined by $K(x, y) = \phi(x)^T \phi(y)$

when w and b is obtained the problem is solved for a simple linear scenario in which data is separated by a hyper plane. The Kernel trick allows SVM's to form nonlinear boundaries. Steps involved in kernel trick are given below (Mitchell, 1997.), (V. Vapnik, 1997).

- a) The algorithm is expressed using only the inner products of data sets. This is also called as dual problem.
- b) Original data are passed through non linear maps to form new data with respect to new dimensions by adding a pair wise product of some of the original data dimension to each data vector.
- c) Rather than an inner product on these new, larger vectors, and store in tables and later do a table lookup, can be represented a dot product of the data after doing non linear mapping on them. This function is the kernel function.

The kernel represents a legitimate inner product in feature space. The training set is not linearly separable in an input space. The training set is linearly separable in the feature space. This is called the "Kernel trick" (Mitchell, 1997.), (V. Vapnik, 1997).

The different kernel functions are listed below: More explanation on kernel functions can be found in the book. (Vapnik, 1995) The below mentioned ones are extracted from there and just for mentioning purposes are listed below.

Kernel Trick: Dual Problem

First we convert the problem with optimization to the dual form in which we try to eliminate w , and a Lagrangian now is only a function of λ_i . There is a mathematical solution for it but this can be avoided here as this paper has instructions to minimize the mathematical equations, It would describe it instead. To solve the problem we should maximize the L_D with respect to λ_i . The dual form simplifies the optimization and we see that the major achievement is the dot product obtained from this (science, 2010).

Kernel Trick: Inner Product summarization

Here we see that we need to represent the dot product of the data vectors used. The dot product of nonlinearly mapped data can be expensive. The kernel trick just picks a suitable function that corresponds to dot product of some nonlinear mapping instead (science, 2010). Some of the most commonly chosen kernel functions are given below in later part of this section. A particular kernel is only chosen by trial and error on the test set, choosing the right kernel based on the problem or application would enhance SVM's performance.

Kernel Functions

The idea of the kernel function is to enable operations to be performed in the input space rather than the potentially high dimensional feature space. Hence the inner product does not need to be evaluated in the feature space. It wants the function to perform mapping of the attributes of the input space to the feature space. The kernel function plays a critical role in SVM and its performance.

1] *Polynomial*: A polynomial mapping is a popular method for non-linear modeling. The second kernel is usually preferable as it avoids problems with the hessian becoming Zero.

$$K(x, x') = \langle x, x' \rangle^d.$$

$$K(x, x') = (\langle x, x' \rangle + 1)^d.$$

2] *Gaussian Radial Basis Function*: Radial basis functions most commonly with a Gaussian form

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

3] *Exponential Radial Basis Function*: A radial basis function produces a piecewise linear solution which can be attractive when discontinuities are acceptable.

$$K(x, x') = \exp\left(-\frac{\|x - x'\|}{2\sigma^2}\right)$$

4] *Multi-Layer Perceptron*: The long established MLP, with a single hidden layer, also has a valid kernel representation.

$$K(x, x') = \tanh(\rho(x, x') + c)$$

There are many more including Fourier, splines, B-splines, additive kernels and tensor products (B.-Scholkopf, 1998). For more on kernel functions refer the book (Shawe-Taylor, 2000).

A particular kernel is only chosen by trial and error on the test set, choosing the right kernel based on the problem or application would enhance SVM's performance.

SVM for Classification

SVM is a useful technique for data classification. Even though it's considered that Neural Networks are easier to use than this, however, sometimes unsatisfactory results are obtained. A classification task usually involves with training and testing data which consist of some data instances (J.P.Lewis, 2004). Each instance in the training set contains one target values and several attributes. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes (P, 1973)

Multi class classification using SVM

Normally the SVM in Matlab tool is a two class classifiers but for a real world problem like pattern recognition it is needed to have multiple class classifiers. For this case uses SVM-KM model of SVM. For Amharic optical character recognition the number of classes might be as per the number of characters. For instance the language has about 310 letters or characters and needs 310 different classes. Currently used multiclass learning methods based on SVM are one-against- all, or one-against-one. and the multiclass learning task is described as follows.

Let $X = \{x_1, x_2, x_3, \dots, x_m\}$

And assign $K > 2 \Rightarrow$ so that each element in X belongs to exactly one class. The goal is to find a dictation function $F: X \Rightarrow \{1, \dots, K\}$, to get a pair wise $(x_i, f(x_i))$ for all $i=1, \dots, m$ and assigned class $f(x_i) = r \in \{1, \dots, k\}$ this is called class label or target. They use kernel function to avoid mapping and duplication.

Standard Multiclass Algorithm

ONE-AGAINST-ALL:- well known simple approach for the assignment of instances to several classes to separate each class from all other classes. And this method is called one- versus-the-rest or one-against-all.

ONE-AGAINST-ONE:- the idea of this method is to describe to extract all pairs of classes and accomplish a binary classification between the two classes in each pair. The training set contains only elements of two classes. The other training instances are eliminated from the set. This results in a smaller complexity in comparison to the one-against-all method, but the number of classes is $O(k^2)$ instead of $O(K)$. Assignment of a class to a test point occurs by voting. An advantage of the pair wise classification is that in general, the margin is larger than in the One-against-All.

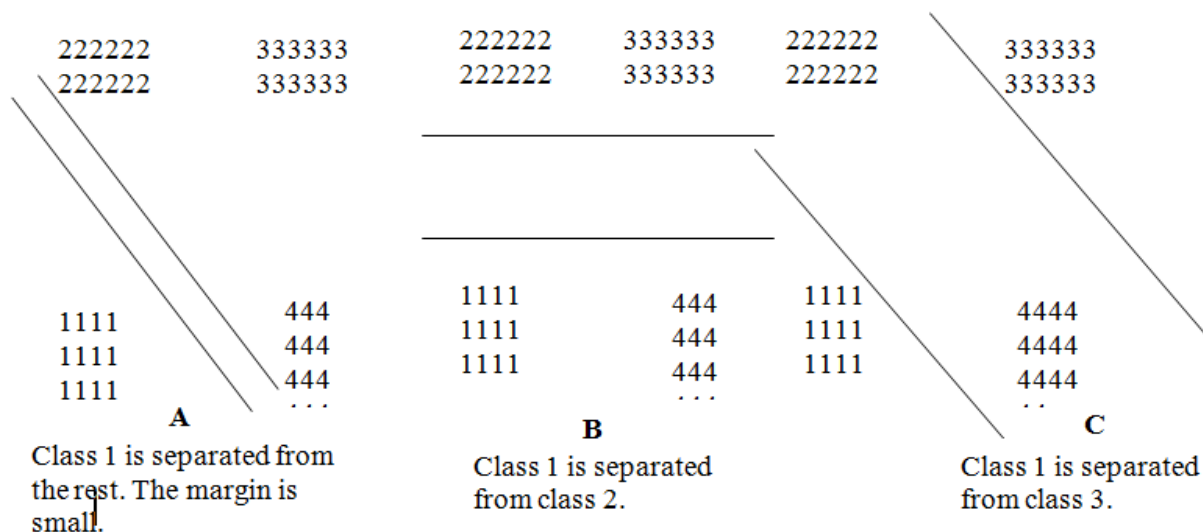


Figure 8: multi class classification for soft margin

From the diagram it is possible to understand that the margins differ more in the one-against-one. Hence, from the [Figure 9(C)] shows a very large margin, because of this the paper uses one-against-one.

Chapter 4

Ethiopic Optical Character Recognition System Design

In the previous chapter the methodology that used in the project explained. This chapter examines the practical aspects and experimental setup that cover the algorithm which embedded in this research. The chapter is divided into two parts and a summery. The first part describes about Ethiopic character set, natures with their Unicode value, and arrangement and in the second part all technical aspects and design issues of the system described.

Part I

The character set

It is written using Amharic Fidel, ፊደል, which grew out of the Ge'ez abugida—called, in Ethiopian Semitic languages, ፊደል *fidel* ("alphabet", "letter," or "character") and አቡጊዳ *abugida* (from the first four Ethiopic letters which gave rise to the modern linguistic term abugida).

The abugida character set has 33 major characters and a combination of consonants and vowels with 6 order of each major character. In general there is $33 \times 7 = 231$ character, in addition to these there are also some labeled and special characters around 40 the so called extended characters. About 20 punctuations and 20 alphanumeric characters. Totally the characters are about 310.

The characters are related in three dimensions.

- I. The first order characters have simple or the basic character shapes.
- II. The other order characters have dashes, circles, etc. additions in almost a uniform pattern for orders.
- III. The pattern of the sound of the characters is the same within an order.

ሀ He - as in hurt

ሁ Hu - as in hood

ሂ Hi - as in hit

- ʏ Ha - as in hat
- ɥ Hie - as in hen
- ʊ H - the "h" sound as in dahlia
- ʊ Ho - as in hot
- ʊ Hwo - as in whole

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1200	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ	ሇ	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሐ
1210	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ሗ	መ	ሙ	ሚ	ማ	ሚ	ም	ሞ	ሟ
1220	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ሧ	ረ	ሩ	ሪ	ራ	ረ	ር	ሮ	ሯ
1230	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ሷ	ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ	ሿ
1240	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቇ	ቈ	቉	ቊ	ቋ	ቌ	ቍ	቎	
1250	ቐ	ቑ	ቒ	ቓ	ቄ	ቅ	ቆ	ቇ	ቈ	቉	ቊ	ቋ	ቌ	ቍ	቎	
1260	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ገ
1270	ተ	ቱ	ቲ	ታ	ቲ	ቶ	ቱ	ታ	ቶ	ቱ	ቲ	ታ	ቲ	ቶ	ቱ	ታ
1280	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ
1290	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኇ	ኈ	኉	ኊ	ኋ	ኌ	ኍ	኎	ነ
12A0	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ
12B0	ከ		ከ	ከ	ከ	ከ			ከ	ከ	ከ	ከ	ከ	ከ	ከ	ከ
12C0	ከ		ከ	ከ	ከ	ከ			ከ	ከ	ከ	ከ	ከ	ከ	ከ	ከ
12D0	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ
12E0	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ
12F0	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ
1300	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ
1310	ገ		ገ	ገ	ገ	ገ			ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ
1320	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ
1330	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ	አ
1340	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ
1350	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ
1360	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳	፳
1370	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲	፲
1380	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ	ሙ
1390	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ	ሠ
2D80	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ
2D90	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ
2DA0	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ
2DB0	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ
2DC0	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ
2DD0	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ	ሐ

Ethiopic 2002 ግዕዝ <http://www.ethiopic.com>

Figure 9 Ethiopic characters with their Unicode value (Unicode, 1991-2011)

Some Ethiopic users have drifted from these basic dimensions and problems have crept in to the usage of two groups of first order characters. Characters involved in this error are "ሀ", "ሐ", "ኀ", "አ" and "ዐ" as these glyphs erroneously share the sound with their respective fourth order form varieties. Ethiopic characters do not share sound across orders and thus "ሀ" and "ኀ" as well as "አ" and "ኣ" do not share the same sound. The true sound of the Geez "አ" is "ea" as in "earth". It is most likely confusion with the wrong usage of "ሀ" as "ha" (instead of "HE" or "ኸ") that forced Unicode to refer to "ኀ" as "HAA" to differentiate it from "ኀ" or "HA". It is because "ሀ" and "ኸ" share the same "HE" sound that "ኀ" and "ኸ" share the "HA" sound or "ሀ" and "ኸ" share the "HO" sound.

Another example is the wrong spelling of "Abeba" as "Ababa" probably on the assumption that "በ" should be spelt as "ba" if the spelling of "ሀ" is "ha". The right spelling of "Addis Abeba" is "አዲስ አበባ" in Amharic and "Addis Abeba" in English. This is also the way I knew it in my geography lessons. This misspelling has also metastasized and an instance is "Asmera" spelt as "Asmara". In the near future Ethiopic will take advantage of speech and character recognition.

The Ethiopic Glyphs:

The glyphs are very close to the Latin alphabet in shape and size. The set consists of syllables, numerals, symbols and notation marks. There is a typeface, but no capitalization.

Syllables:- Syllables have their own names.

Symbols:- have their own symbols.

Numerals:- The digits have their own names and unique symbol.

Pronunciation:- Ethiopic is a syllabic alphabet and each character represents a separate sound. However, there are a few characters that represent the same series of sound.

Part II

System Design

A system to recognize and classify Ethiopic character should be able to perform in two stages. The training stage in which the training set can be built by collecting a set of Ethiopic character from different font combination and different shapes for training and validation, and a classification and mapping stage in which the system can recognize a character and classify to the appropriate class and to map to the lookup table. A system to recognize Ethiopic character is shown in [Figure 10]. It consists of modules which work together to perform the recognition.

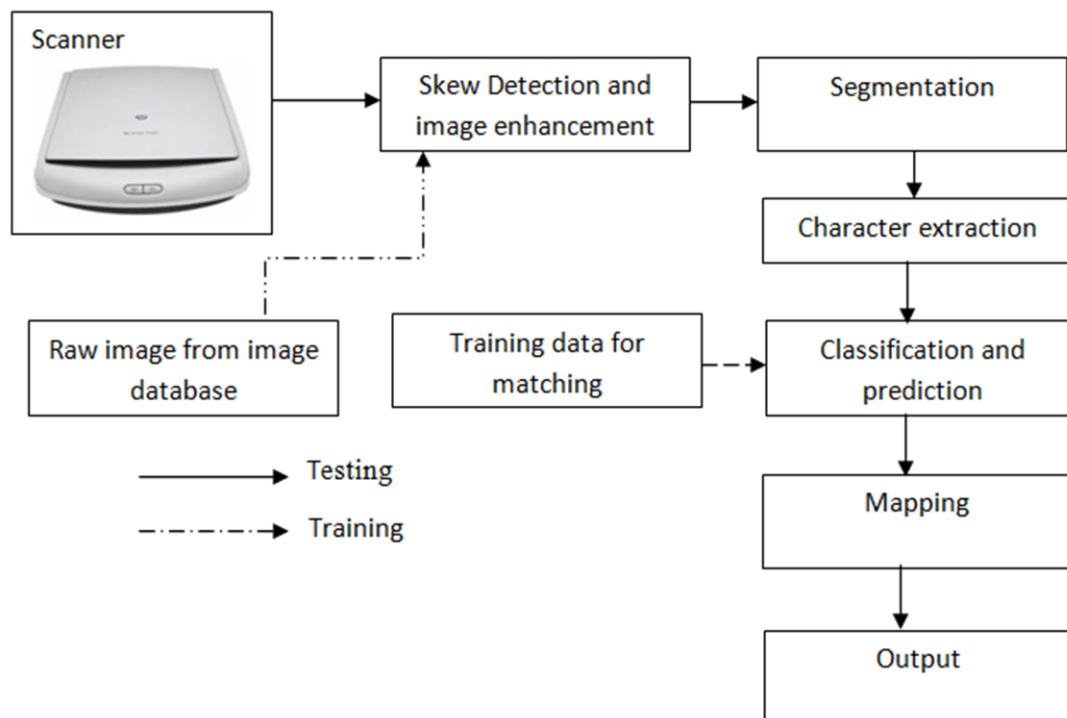


Figure 30 System design for Ethiopic optical character recognition.

a. The scanner

A good quality scanner which gives good quality image is important. No special equipment is needed for this purpose. Any scanner can work for it. But these days most of the scanner has good image enhancement and skew detection features, this is good to choose scanner.

b. Skew detection and Image enhancement

An efficient and accurate method for determining document image skew is an essential need, which can simplify layout analysis and improve character recognition. Most document analysis systems require a prior skew detection before the images are forwarded for processing by the subsequent layout analysis and character recognition stages. The aim of Image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide 'better' input for other automated image processing techniques or for the recognition phase.

c. Segmentation

The document can be scanned by any types of color and picture resolution but for the recognition process it should be converted to black and white, and extracting the document zone.

d. Character Extraction

After the document zone is identified the next step is identifying horizontal line and extracting every connected character from the horizontal line. And this extracted character will be sent to the SVM for classification and prediction.

e. Training set

The training set consists of binary image of a normalized size such as 30*30 pixels. The training set is created and updated in the training stages in such a way that binary images of the desired Ethiopic character are selected from a set of images.

f. Classification

Classification is carried out using a Support Vector Machine (SVM).

g. Mapping

The desired output will be mapped to the lookup table. The lookup table contains the hexadecimal value of the character in the standard of Unicode. Then the exact character will be displayed in office application or to save it for future use.

The scanner

The purpose of greater scanning resolution is to create more pixels, to create a larger image size.

35 mm film is relatively tiny, requiring greater scanning resolution than photo prints to create an enlarged image for printing. The ratio of (scanning resolution/printing resolution) is the enlargement factor. For example, scanning film at 2700 dpi and printing scaled to 300 dpi gives $2700/300 = 9$ times enlargement of the original film size. 9X is about 8x12 inches (near A4 size) from full frame 35 mm. This enlargement requirement is why film needs high resolution.

Most flatbed choices are 600 or 1200 dpi now, and some are 2400 dpi. You won't need more than 300 dpi for scanning photo prints, or 600 dpi for line art documents, assuming printing at original size. 1200 and 2400 dpi would be used for scanning film.

Flatbed scanner specifications are stated **with two numbers, like 1200x2400 dpi**. Flatbeds also usually specify a maximum resolution, like perhaps 9600 dpi. So what does all of this mean?

A scanner scans **one horizontal row of pixels** at a time, moving that scan line down the page with a carriage motor. The **smaller** dpi number is the **optical resolution** of the CCD sensor cells. A 1200 dpi scanner takes 1200 color samples per inch (creates 1200 pixels per inch) **horizontally** from the width being scanned. A 1200 dpi CCD sensor really cannot do anything else but scan at 1200 dpi. This rating does not mean that it can resolve 1200 lpi in a test target, but instead, the CCD simply reads 1200 samples per inch. Nyquist sampling theory

says the image can never resolve more than 1/2 of that detail level, and in the real world, a little less.

The **larger** dpi number is the possible positioning of the carriage **stepping motor**. A stepping motor doesn't rotate continuously like regular motors. Instead it is pulsed to move in precise steps, rotating only a few degrees with each input power pulse. A 1200x2400 dpi scanner is geared so that each pulse of the carriage motor moves in $1/2400$ inch steps **vertically**. If we scan at 300 dpi, the carriage moves eight motor steps at a time vertically, then stops and samples, and resample's the scan line to 1/4 size horizontally, to create the 300x300 dpi image requested. If scanning at say 250 dpi, it should move $2400/250 = 9.6$ steps per row, but it can only move 10 steps on some rows, and 9 steps on others. Any location error will be less than half a CCD cell height, even in worst case. This is the purpose of the 2X rating of the motor. The purpose is NOT to scan at 2400 dpi. The motor does not contribute to optical resolution. A 1200x2400 dpi unit is a 1200 dpi scanner.

Most flatbed scanners also advertise a "maximum" resolution, 9600 dpi, or even more, but this is a meaningless number. It is simply interpolated resolution, and you can do the same thing blowing up the image later in a photo program (except you won't, the quality is blurred, not improved). Resolution greater than the CCD optical rating is simply interpolated resolution, done in software after the 600 or 1200 dpi optical scan. Interpolated resolution is the least important scanner specification. It was **useful for line art mode, and only for line art**, to reduce jaggies when we had 300 dpi scanners and needed 600 dpi line art.



Figure 11 a scanner machine

The flatbed scanner bed is 8.5 inches wide, so a 1200 dpi CCD sensor is an array of (1200 dpi x 8.5 inches) = 10200 pixels in one horizontal line. A wide-angle optical lens focuses the 8.5 inch image width onto a much smaller CCD chip, using mirrors to fold the long optical path inside the scanner. A typical flatbed CCD array is perhaps 72 mm wide, with 7x7 micron cells (3628 per inch in this example) being popular today. The carriage motor moves the CCD scan line vertically down the length of the bed, taking a 10200x1 pixel scan line sampled periodically from the 8.5 inch image, at each image row location where the carriage motor stops. We call this 1200 dpi, and for all purposes it is, because at the glass bed, 10200 pixels / 8.5 inches = 1200 dpi.

A 35 mm **film scanner** uses a different optical lens which covers only the 0.9 inch film width instead of 8.5 inches. 4000 dpi over 0.9 inches is 3600 pixels, instead of 10200 pixels. That's a big deal, and the narrow width allows higher resolution, and in particular allows larger CCD cells, which means more CCD quality with less CCD noise. This larger sensor size is a big advantage for film dynamic range.

Digital photo images have "square" resolution, the same in both directions like 300x300 dpi, simply called 300 dpi. If we did try to scan film at 2400 dpi using a 1200x2400 dpi scanner, the carriage motor can indeed step at 2400 dpi vertically. However, all samples will overlap each other vertically by 50% because the 1200 dpi CCD cells are twice taller than $\frac{1}{2400}$ inch in size. Horizontally, the CCD can only sample at 1200 dpi but our images must be square resolution, so the software interpolates larger horizontally to create a 2400x2400 dpi image.

This will not be the same quality as a "true" 2400 dpi CCD can do, either horizontally or vertically.

For this research and full featured system any types of scanner can be applicable and no need of special features.

Skew Detection and Image Enhancement

An efficient and accurate method for determining document image skew is an essential need, which can simplify layout analysis and improve character recognition. Most document analysis systems require a prior skew detection before the images are forwarded for processing by the subsequent layout analysis and character recognition stages. Document skew is a distortion that often occurs during scanning or copying of a document or as a design feature in the document's layout. This mainly concerns the orientation of text lines, where a zero skew occurs when the lines are horizontal or vertical, depending on the language and page layout. Skew estimation and correction are therefore significant preprocessing document restoration stages before the actual document analysis.

In general, there can be three types of skew within a page: a global skew, when all text areas have the same orientation; a multiple skew, when certain text areas have a different slant than the others; and a non uniform text line skew, when the orientation fluctuates within a line, e.g. a line is bent at one or both of its ends, or a line has a wave-like shape.

Currently there are a lot of on shelf skew detector and corrector software, but this paper uses and implements technique for detecting and correcting the skew of text areas in a document and see (P. SARAGIOTIS, 2008).

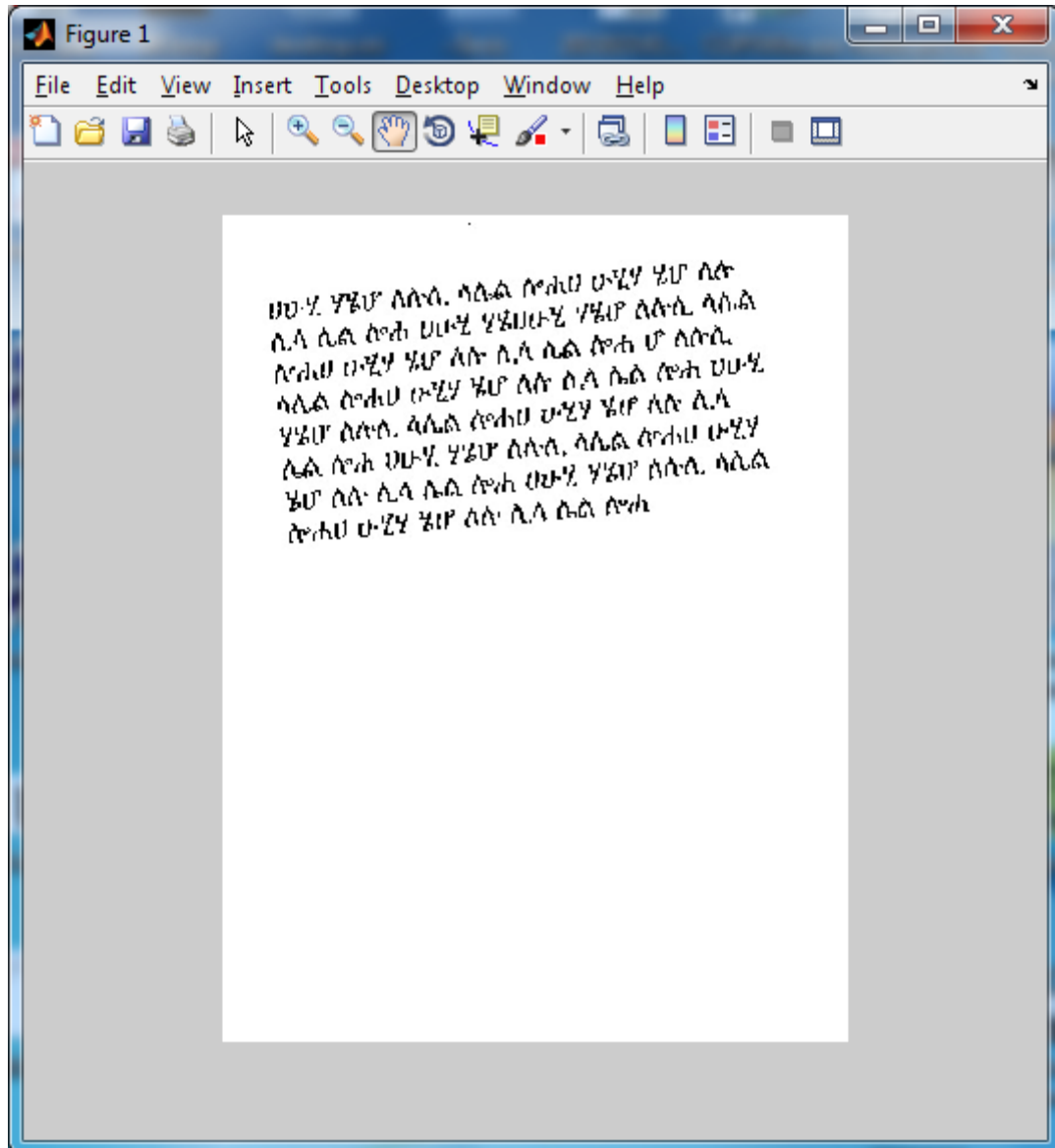


Figure 12 Skew Detected Images

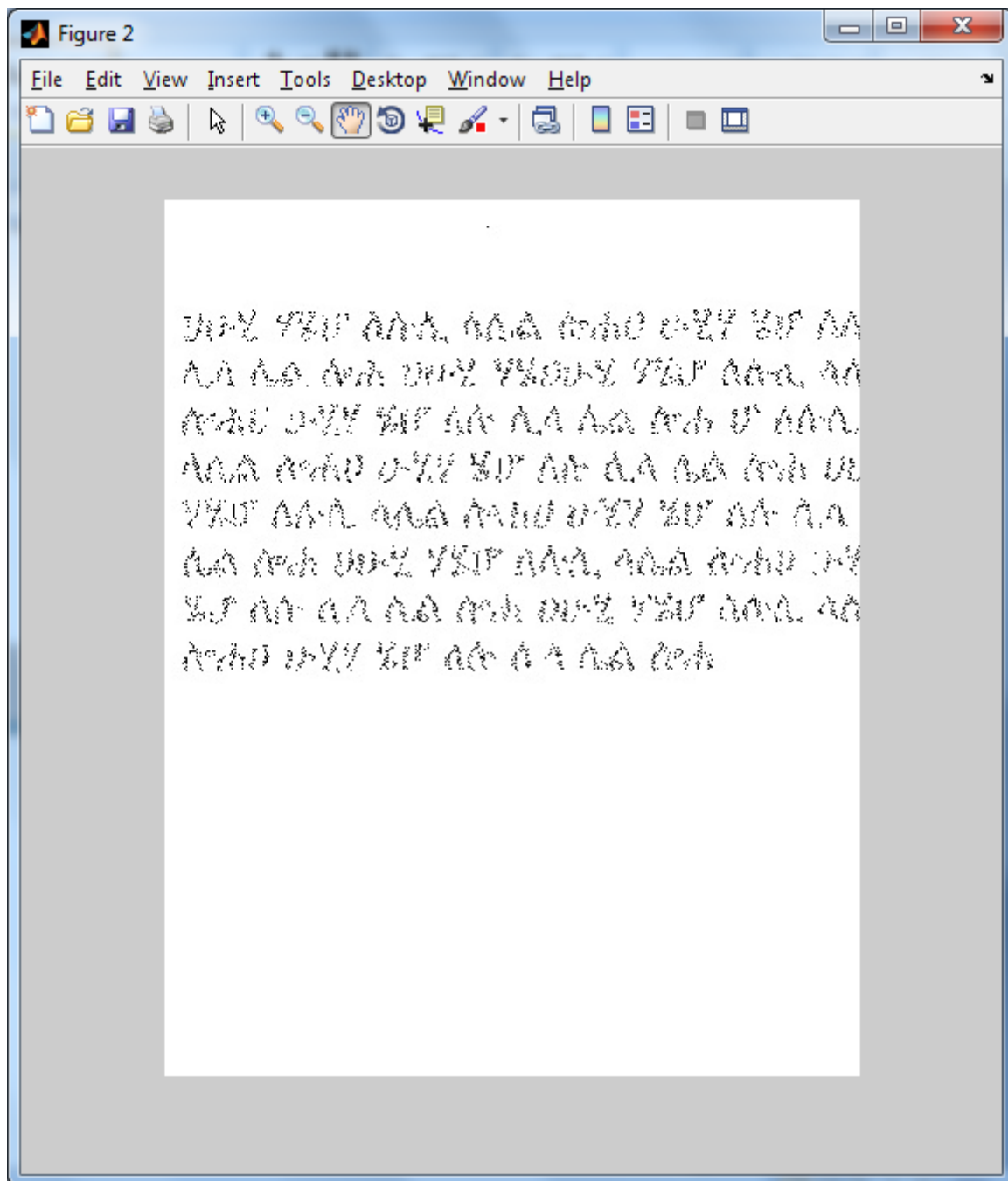


Figure 13 after skew correction

Image enhancement for old documents

The quality of the image is the important factor for the good character recognition software, this section presents the image enhancement that applied in a very old documents but it is possible that this types of old documents needs more image enhancement technique.

Image enhancement, since the methods take an input image create another image as an output and will be input again for the Ethiopic recognition system. Other appropriate terms often used are filtering enhance, or conditioning. The major notion is that the image contains some signal or structure, which we want to extract, along with uninteresting or unwanted variations, which we want to suppress.

A paper document needs to be scanned converted into a text file. Before applying character recognition, noise pixels need to be cleaned from the background and dropouts in the characters need to be filled.

The most important methods are noise filtering, contrast adjustment and image smoothing and thinning .All these are implemented in this system. Here are sample old and very noisy pictures and its enhanced picture next.



Figure 14 this document is very old and noisy.



Figure 15 after applying different image enhancement Technique

Segmentation

The document can be scanned by any types of color and picture resolution but for the recognition process it should be converted to black and white, and extracting the document zone. The algorithm to segment the document is first calculating the threshold value then using the Matlab built-in function `rgb2bw` the image will be converted to black and white binary images. Then secondly from the scanned image the document zone will be identified. Then by applying horizontal projection the horizontal line will be identified, in the horizontal line the connected characters will be identified and extracted.

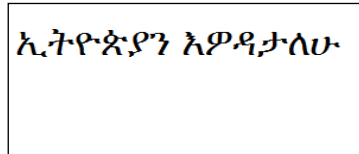


Figure 16 scanned document in image form

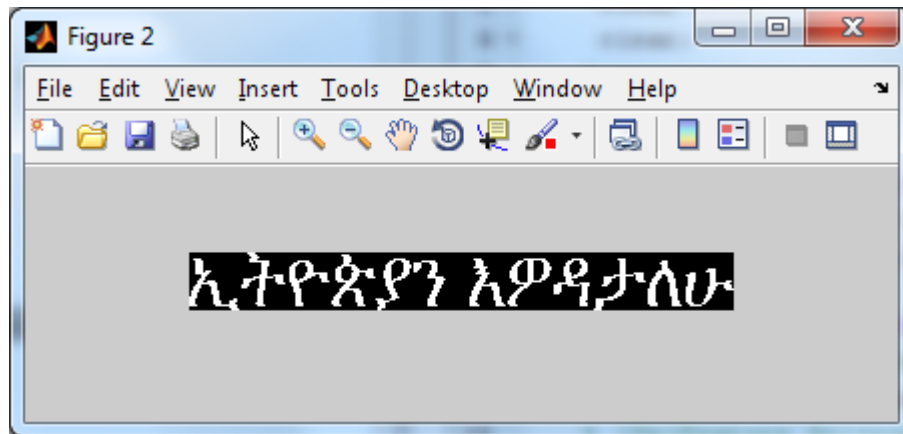


Figure 17 loaded and segmented image

Character Extraction

After the document zone is identified the next step is identifying horizontal line and extracting every connected character from the horizontal line. And this extracted character will be sent to the SVM for classification and prediction. The algorithm works, after the document zone is identified it tries to calculate and count the horizontal line from the first sentence to the last, then the main loop iterates the number of lines times, within each iteration it takes one line then it counts connected images from the extracted line, then another loop inside the main loop will extract each connected objects one by one, this iteration continues up to the number of connected objects in that specific extracted line. [Figure 18] shows that a sample extracted connected object that is loaded to the system.

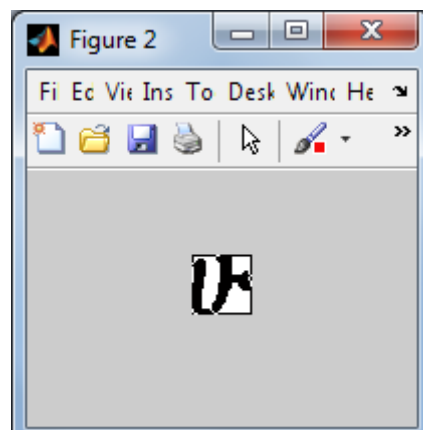


Figure 18 sample character extracted from the above segmented image

Training Set

The training set consists of binary image of a normalized size such as 30x30 pixels. The training set is created and updated in the training stages in such a way that binary images of the desired Ethiopic character are selected from a set of images. The Ethiopic character has about 310 different characters, for this research purpose this paper chooses only 15 different characters. And for each 15 different characters 100 different samples is prepared, the samples provided from different fonts that are used in the country currently and most of the samples are duplicated. Total training set contains 1500 training samples. From this sample data 1200 samples used for training and the remain used for testing. This means 80 samples per character for training and 20 samples for testing. In the following table the training image, their class label and desired output and actual character is displayed.

Table 3 Training image desired output and actual character

Training Images	Class label and Desired output	Actual character
	1	ሀ
	2	ሁ
	3	ለ
	4	ታ
	5	ት
	6	ኢ
	7	እ
	8	ዎ
	9	ያ
	10	ዮ
	11	ዳ
	12	ሠ
	13	ጵ
	14	ን
	15	ሐ

To implement the training and testing activity in the system the following sequences of steps applied. First [Figure 19] depicts that loading the collected training image then segment and convert to binary then normalize and create a vector image then the by collecting all the vector image a single matrix will be created. In this big single matrix dimension reduction will be applied then the SVM will take and make training.

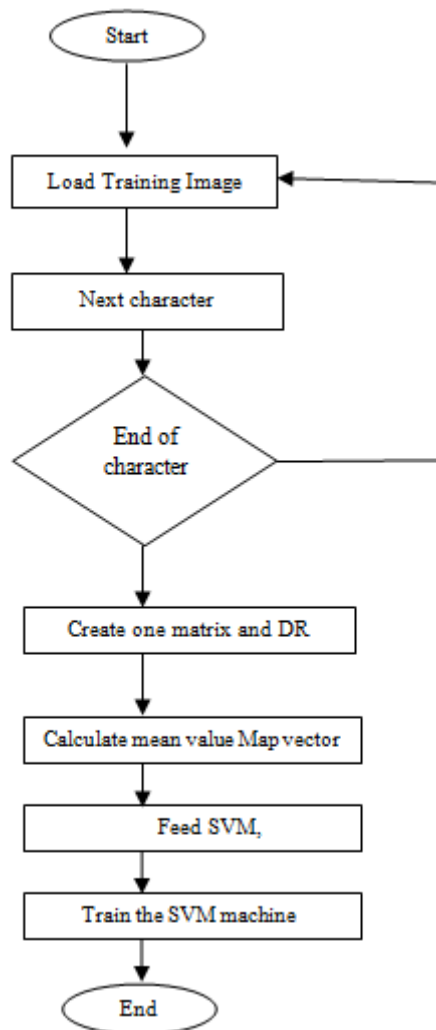


Figure 19 Flow chart for the Training Activity of the System.

For the testing stage the training will be loaded and the new scanned document will be loaded then a certain image enhancement and skew detection and correction will be applied, if it requires. Then the document zone will be detected from the image and the lines and characters will be identified and extracted, then the extracted characters will be feeded to the SVM and the SVM will return the desired output as a class label. [Figure 20] describes this sequence of activities clearly.



Figure 20 flow chart for testing and prediction part of the system

Mapping

In the above sections of this chapter the SVM classifies and predict the class or the target values. Using the target value the system will map to the lookup table to get the hexadecimal values of the target output. The lookup table contains hexadecimal values of the Ethiopic character. Then the extracted hexadecimal values will be displayed using browser or office application software. For example for class 1 its corresponding hexadecimal value is 'ሀ' and this value will be passed to the browser and the browser will interpreted to its equivalent character **ሀ**.

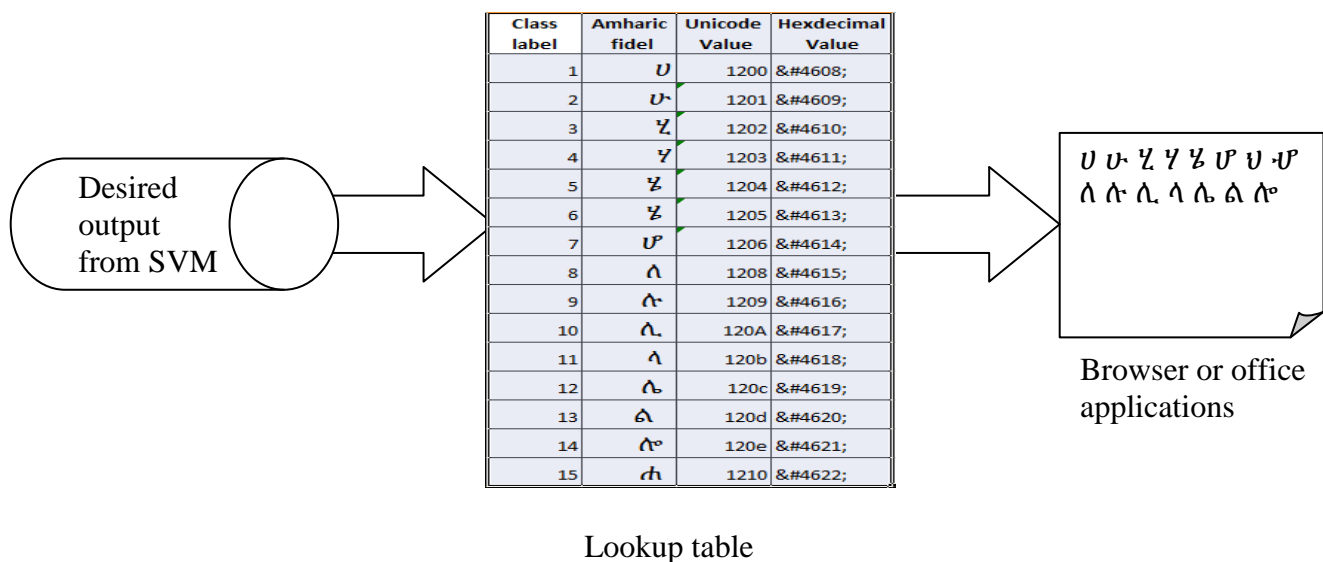


Figure 21 from the classifiers up to the office application

The following three pictures [Figure 22] , [Figure 23] and [Figure 24] are a screen snapshot that displays the actual input output follow of the above all explanations.

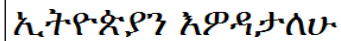


Figure 22 scanned document

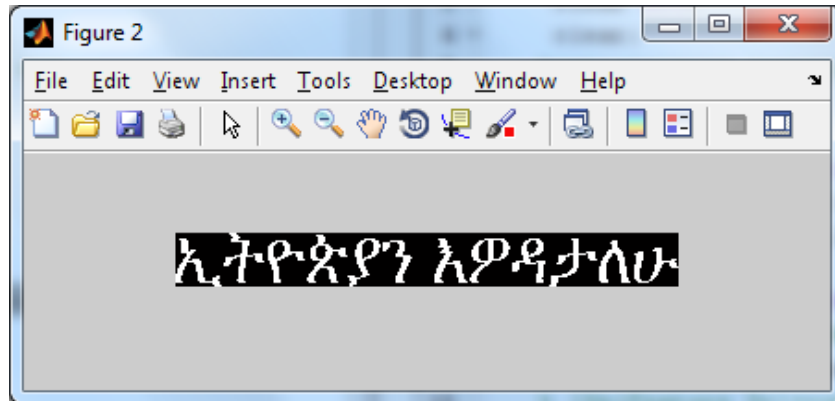


Figure 23 loaded to the system

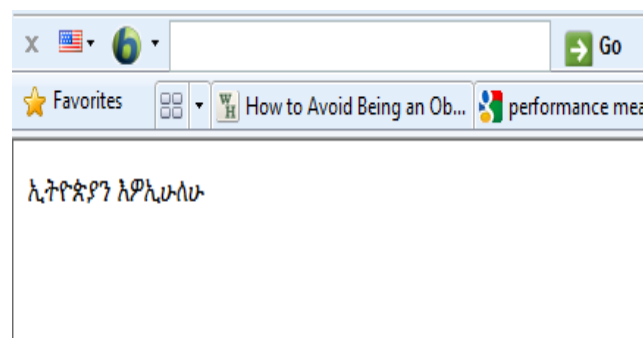


Figure 24 system result

Binary Representation

Binary representation is the most straightforward and simplest method to present a binary image; 0 denotes black pixels while 1 denotes white pixels. In this project since each binary image is saved in 30×30 pixels, so totally there are 900 attributes for one input vector. Figure 26 shows an example of **ሁ** character.



Figure 25 Binary Image

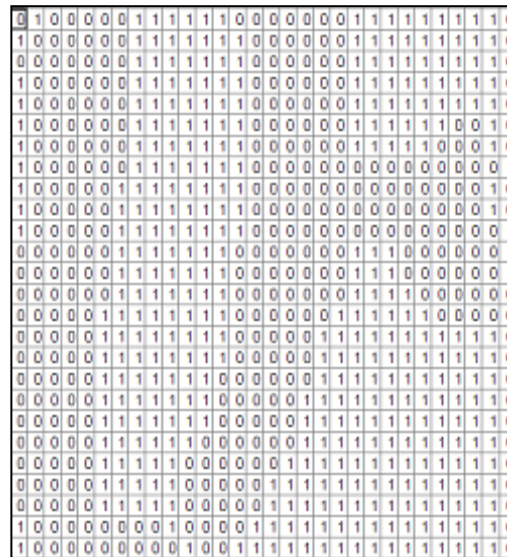


Figure 26 Binary images in pixel

Data Normalization and Dimension Reduction

Before the data are presented into SVM for training and test, normalization is an important process that scales and evenly distributes the data into an acceptable and meaningful range. The above binary is presented to the SVM as it is the 900 features directly but this makes the system very slow to train as well the classification rate is about 35% is correctly classified. To avoid this problem it should have a mechanism to reduce the data into more meaningful and acceptable range of data. The mechanism is called dimensions reduction; to reducing the dimension of the input vector Principal Component Analysis (PCA) is used. The toolbox that provides this PCA is called DRtoolbox. The toolbox provides 37 different dimension reduction tools.

Data Representation

The following notation is used for data representation in to SVM

$y \ d1:x1 \ d2:x2 \ d3:x3 \ d4:x4 \ d5:x5 \dots \dots \dots \ d_n: X_n$

Where, y is the desired output of the data sample, which identifies a class, for each character

classes classification $y = 1$ to 15 since there are 15 categories, $i x$ is the i^{th} attribute of a input vector \mathbf{x} and $i = 0$ to n , $i d$ is the index of attribute $i x$ starting from 1 in an ascending order. In training data set the desired output y is used to supervise the SVM learning while in test data set y is used to verify the output of SVM. If the desired output of test data set is unknown, y can be any number. In that case, the test result cannot be verified whether the actual output is correct or not.

Features and facilities of Microsoft Windows vista / 7

In the earlier version of Microsoft operating system like windows ME, XP and 2000 there is no any way to use Amharic scripts in the computer. Now in the new version windows vista and 7 a new font called **NYALA** and **Unicode** that supports Ethiopic scripts is included. Then now it is possible to use a standard font and the office applications easily without need of any additional font software. This features and facilities make simple the development process.

Unicode

Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, JustSystems, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others. Unicode is required by modern standards such as XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc., and is the official way to implement ISO/IEC 10646. It is supported in many operating systems, all modern browsers, and many other products. The emergence of the Unicode Standard, and the availability of tools supporting it, is among the most significant recent global software technology trends. **Unicode provides a unique number for every character, No matter what the platform, No matter what the program, No matter what the language.**(Unicode, 1991-2011)

Summery

In this chapter, the building block of the Ethiopic character recognition system has been presented. To develop an Ethiopic character recognition system, a collection of images which were collected from different fonts are prepared and a scanned image is prepared to test and validate the system. As stated above the visual similarity of the character and the largeness of the numbers of character is the problem that is challenged this research too. However, the support vector machine is used to classify. This classifier was trained and tested by either a set of normalized images.

Note that the size of each training samples and the newly extracted characters is resized in to 30*30 pixels. This size is determined by trial and error, the trial and error chooses because of, Amharic characters can differ in size. There are short characters like መ, ሠ, and ፈ and there are very long characters such as ቅ, ኸ, and ሚ there is also noticeable variance in width, for instance between ኀ, ሚ, and ም. For the implementation case every training and testing images should have the same size and to compensate the variety in size such as width and length this paper chooses a 30*30 size finally.

The performance of the whole system in general and every individual step in particular together with failure analysis and reason for failure are presented in the next chapter.

Table [Table 5] shows that the chosen parameter to do the experiment for the recognition model.

Table 4 chosen and applicable parameter for the experiments

Machine learning and classifier tool	Support Vector Machine (C-SVM)
Kernel Function	Gaussian Radial Base Function
Dimension reduction tool	Principle Component Analysis (PCA)
File format Support	bmp, jpeg
Character encoding technique	Unicode
Destination Supported Font	Nyala
Output file format	HTML
Recognized fonts	Possibly all available Ethiopic fonts
Character Mapping	Using a lookup table.
Sample character	15 characters
Samples for each Character	100 Samples
Total Training set per character	80 for each
Total Testing set per character	20 for each
Execution control parameter(c)	100000 (trial and error)
Initial value for lambda	1E-8 (trial and error)
Number of classes	15(equal with number of characters)
Class labels	1, 2, 3, 4, 5, 6, ..., 15
Size of Images for training and or testing	30*30
SVM toolbox type	SVM-KM
Dimension Reduction Tool	DRtoolbox

Except the number of classes and the kernel, the value of the remains parameters assigned by using trial and error by looking the classification result while running the training.

Chapter 5

Results Analysis

This chapter focuses on the evaluation of the performance of the Ethiopic Character recognition System. The performance evaluation is done first for very noisy and old images. In the country there are different old documents written by different types of type writer machines with different styles. Secondly skewed images, for different dimension reduction tool and different kernel types of the SVM with different parameters. The last part of this chapter is an analysis of the classifier performance and the parameters which could affect the classification rate. Finally from the overall performance evaluation Gaussian Radial Base Function is chosen as good kernel trick, PCA is used for dimension reduction tool and 100000 is used for execution control parameters and $1E-8$ is lambda value.

Noisy and old images

Image noise is the random variation of brightness or color information in images produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is generally regarded as an undesirable by-product of image capture.

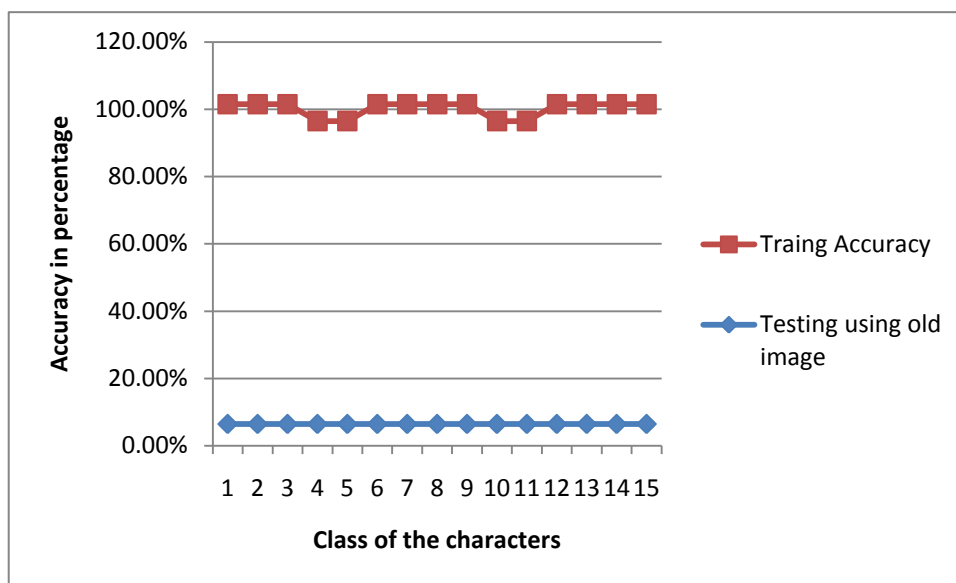
Old images are images that losses its quality through long time and old the styles the character and every related things. [Figure 14] and [Figure 15] are examples of Old image and enhanced image.

In this experiment the noisy image is scanned and feeded to the system. The system applies different image enhancement technique to remove the noise and to filter the exact images from the noise. Figure 14 shows that the original old and noisy Ethiopian documents. [Figure 15] shows that the new enhanced binary image. And as you can see it is difficult to recognize the character from the image. For the image like this with all good kernel trick dimension reduction and parameters, the performance result is 6.5%. Note that ‘good’ is from the point of view of this research.

Table 5 old image testing failure result analysis

Total character Set	Training Set	Testing Set
	Training Set(80)	Testing set(20)
15	98 %	6.5 %

The quality of the image is the important factor for the good character recognition software, from the result analysis it is possible to see recognition is difficult for very noisy and old images and it requires too much image enhancement technique.

**Figure 27 experiment analysis for testing with noisy and poor quality image**

Skew Detection and Correction

Inaccurate deskew will significantly deteriorate the subsequent processing stages and may lead to incorrect layout analysis, erroneous word or character segmentation and misrecognition. The overall performance of a document analysis system will thereby be severely decreased due to the skew.

Basic steps

1. *Word grouping*
2. *Text line identification*
3. *Text line skew calculation*
4. *Text area region growing.*
5. *Text area rotation.*

Table 6 Skew Result Analysis

Rotation in Degree	Detection Error
0 ⁰	0.154
5 ⁰	0.158
15 ⁰	0.147
30 ⁰	0.179
40 ⁰	0.339

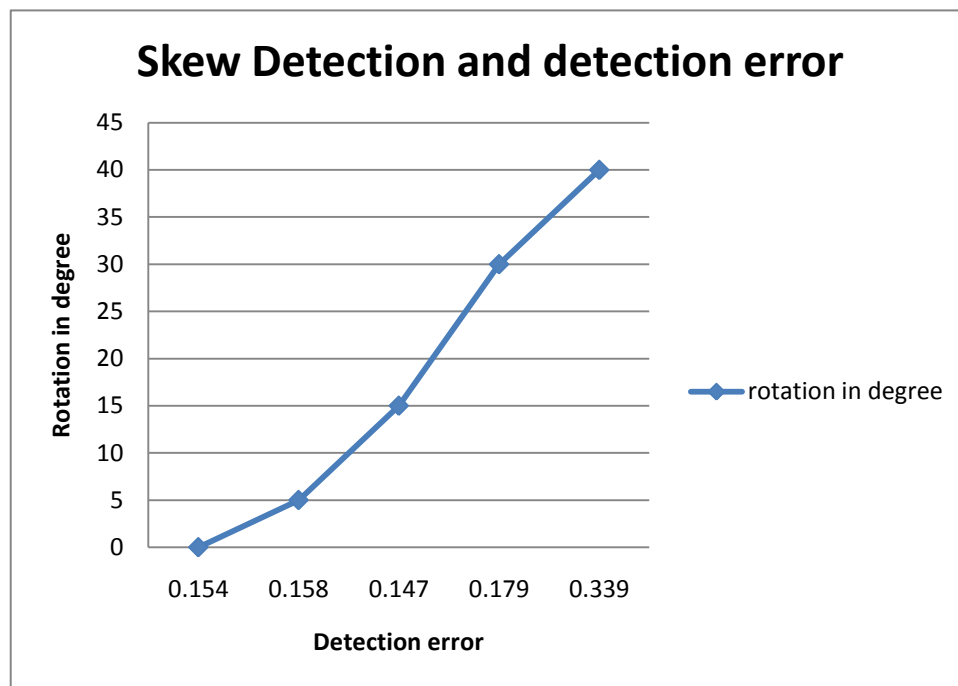


Figure 28 experiment analysis for skewed documents with different degree of rotation

This paper presents for documents skewed by 5° , 15° , 30° and 40° and estimated the skew angle using a technique for detecting and correcting the skew of text areas in a document. Then the absolute detection error is calculated, which is defined as the absolute difference between the detected skew angle and the given ground-truth and its standard deviation. There is also calculated the correlation between the detected skew angle and the given ground-truth. From the results, which are presented in [Table 4], it is possible to conclude that the proposed technique is robust in handling all possible rotation angles without any variation on its accuracy. Also, the technique is proven able to handle all the different types of documents. Furthermore, the estimated skew angle is strongly correlated with the given ground-truth. For further refer [21].

Documents in different fonts

In the country there is no standard font to use Ethiopic scripts in the computer. The font developed by some developers has good beginning for the Ethiopian computer user using their native language, but the problem is peoples are using different fonts in different machine different fonts, no standard and no compatibility between those fonts. And now to narrow this

difference and incompatibility problem, this research uses the training samples from all different available and most popularly used fonts then it will be possible to convert from different documents written by different fonts to common standard fonts the so called Unicode. This achievement is a good beginning for developers to develop the full scaled software. This performance evaluation is done by the chosen good kernel trick dimension reduction and parameters.

Table 7 General result analysis for different fonts with PCA , Gaussian, 10000 control parameter and 1E-8 lambda value.

Image number	Document font	Number of character	Correctly Classified	False Negative	Time
1	Nyala	15	14	1	9s
2	Agafari	15	13	2	11.85s
3	Power geez	15	14	1	10s

Classification using different dimension reduction tool

It is possible to present directly the binary image with the pixel value to SVM but the problem is when the number of samples increases the total data size will; increase and that will be a problem for the speed and error free classification as stated above in the previous chapter, the importance of the dimension reduction is to get rid of this problem and to prepare more meaningful and acceptable small sized data. A dimension reduction tool that the project cited is DRToolbox and it provides about 37 different tools out of these different tools NPE, PCA, and MDS is chosen. Performs a technique for dimensionality reduction on the data specified in A , reducing data with a lower dimensionality in $mappedA$. The data on which dimensionality reduction is performed is given in A (rows correspond to observations, columns to dimensions). A may also be a (labeled or unlabeled) PRTools dataset.

- a. NPE is good for large amount of features and the total size of the samples should be greater than the features of single characters. For example for my selected 15 characters, each character is $30 * 30$ then the total features will be 900 then the total

amount of the training samples should be greater than 900 to use NPE, in this case the total character is 1500 and applied and the following result is achieved.

- b. PCA is a principal component analysis it is a traditional types of dimension reduction and classification tool. PCA create the mean value of all the samples and reduces the size to the desired setup like 2 columns, and it generates the mapping value for classification and testing. And PCA is chosen for this project due to its good value to reduce the dimension of the features.
- c. MDS is the same as PCA in all its behavior except in some cases.

Note that in DRtoolbox there are about 37 different dimension reduction tools each of them has their own characters and parameters to apply in a given problem. For this pattern recognition this paper chooses the above three and in future for large full scale software researchers and developers can see each in detail and choose their best.

Classification using Different kernel trick with the selected dimension reduction tool.

This part focuses on the analysis of performance of the SVM using different kernels. The SVM recognition model is trained using for basic kernels: linear, polynomial and Gaussian Radial base Function (GRBF). Each kernel experiment is carried out using the same pair of training/test dataset. And other parameters like $c=10000$, $n=15$, $v=1$, $\lambda=1E-8$, where n is the number of classes. the following tables contains performance evaluation for a combinations of each of chosen 3 fonts namely Nyala, Power Geez and Agafari with the kernel types namely linear, polynomial and RGBF, with PCA, MDS and NPE types of dimension reduction tool. The total combinations become 27. Their speed in Second is also specified.

Table 8 Experiment result for a combination of Nyala font and PCA DR with the three kernel tricks.

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	95	12	92	8
Polynomial	95	13.5	93	11
GRBF	98%	11	95	9

Table 9 Experiment results for a combination of Power Geez font and PCA DR with the three kernel tricks.

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	95	13	92	12
Polynomial	85	15.85	80	14
GRBF	98%	12	95	11

Table 10 Experiment result for a combination of Nyala font and MDS DR with the three kernel tricks.

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	95	12	92	8
Polynomial	95	13.5	93	11
GRBF	98%	11	95	9

Table 11 Experiment result for a combination of Nyala font and NPE DR with the three kernel tricks.

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	95	12	92	10
Polynomial	95	12.85	93	11
GRBF	98	11	95	11

Table 12 Experiment results for a combination of Power Geez font and MDS DR with the three kernel tricks

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	95	13	92	12
Polynomial	85	15.85	80	14
GRBF	98%	10	95	9

Table 13 Experiment results for a combination of Power Geez font and NPE DR with the three kernel tricks

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	90	12.5	92	12
Polynomial	80	15.85	93	14
GRBF	95	13	95	11

Table 14 Experiment results for a combination of Agafari font and PCA DR with the three kernel tricks

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	87	12.87	80	11
Polynomial	80	13.85	75	15
GRBF	93%	12	84	11

Table 15 Experiment results for a combination of Agafari font and MDS DR with the three kernel tricks

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	87	12.87	80	11
Polynomial	80	13.85	75	15
GRBF	93%	12	84	11

Table 16 Experiment results for a combination of Agafari font and NPE DR with the three kernel tricks

Kernel	Training %	Time for training(s)	Testing %	Time for testing(s)
Linear	90	14	80	12.89
Polynomial	87	12.85	75	13.5
GRBF	95	12	90	12

Different documents with different fonts are tested and [Table 7 - Table 15] results analysis are conducted with the time duration for processing the document. This result shows that the proposed OCR application can avoid incompatibility problem among the differences of different documents.

The confusion matrix is commonly used to report results of classification experiments. The following two tables [Table 16] and [Table 17] is confusion matrix for training classification and testing classification respectively. For example the entry in row I, column J records the number of times that an object labeled to be truly of class I was classified as class J.

Table 17 confusion matrix for training

Desired Output	Classified as															Error Classified
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	78	2	0	0	0	0	0	0	0	0	0	0	2
5	0	0	0	1	79	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	63	17	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	2	78	0	0	0	0	0	2
11	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0

Table 18 confusion matrix for Testing

Desired Output	Classified as															Error Classified
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	18	2	0	0	0	0	0	0	0	0	0	0	2
5	0	0	0	1	19	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	3	17	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	2	18	0	0	0	0	0	2
11	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0

The confusion matrix diagonal, where $i = j$, indicates the success: with perfect classification results.

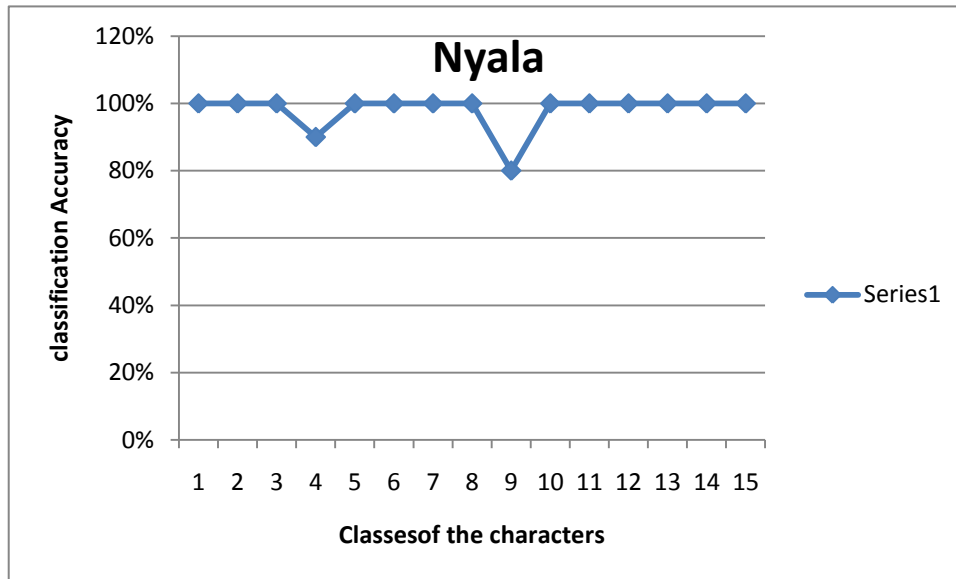


Figure 29 classification accuracy for Nyala font

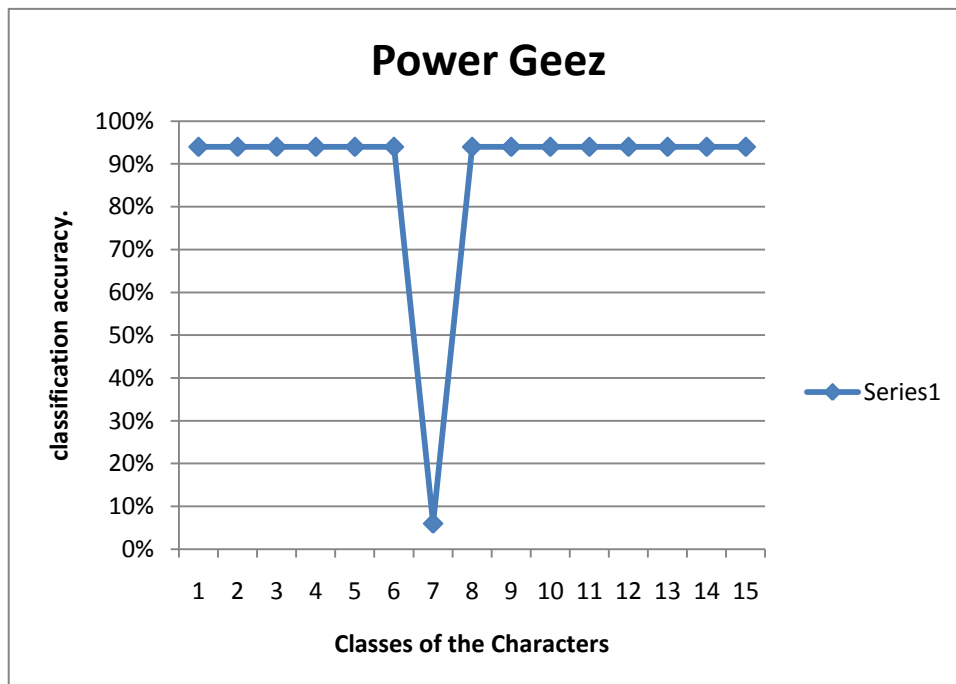


Figure 30 Classification accuracy for Power geez Font

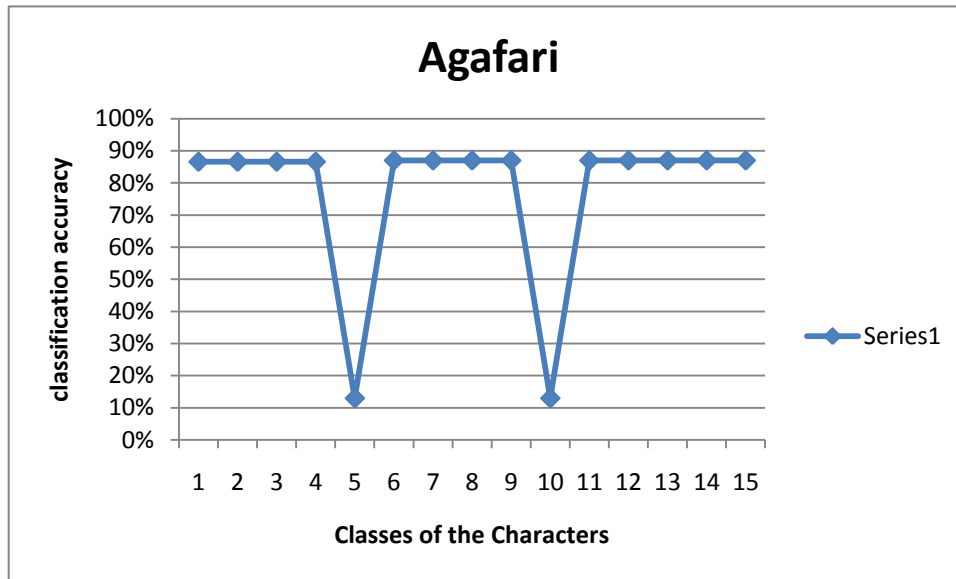


Figure 34 Classification Accuracy for Agafari font

This system is expected to digitized scanned documents written in different fonts which are incompatible and which have different font and shapes of characters to one common Unicode based font the so called Nyala, and this is the success and very beginning objective of the project.

Reason for Failure Analysis

From the above result analysis in the previous section it is possible to see that the system is successful, but there are possibilities of failure and the major three reasons are described under this section. First the quality of the image is the important factor for the good character recognition software, an image with low quality is difficult to recognize the character and every noises and scratched will be recognized as characters and the prediction will be wrong totally, again this paper recommends the quality of the image is an important issue in character recognition system. From the result analysis in [Figure 27] and [Table 3] it is possible to see recognition result for very noisy and old images and quality images respectively. Second issue for failure is disconnected objects. Basically this system works with connected objects and a character which is made from two or three disconnected object is classified as wrong and each of them classified as a single object. Third and the final one is, the shape of many Amharic characters shows similarities with few distinctions among them, for example ሀ and ቢ, ሠ and ጠ, ይ and ደ, ተ and ቸ, ኀ and ኘ and ነ, ዘ and ዠ. Sometimes the classifiers classified these characters in to the reverse and wrong types of class because of their similarity. These similarities will be also a problem for Amharic OCR that works using features.

Result Analysis Summery

The Number and Type of OCR Engines Available

Modern OCR software uses multiple engines to achieve a high level of accuracy. My research focuses on machine print OCR (OCR-A, OCR-B, etc) recognition engines.

Recognition Speed

The speed at which the OCR software recognizes a given scanned document is summarized by the table [Table 7] above.

Supported Output Formats

The number of possible output formats such as XML, HTML, PDF, and DOC in which the output of the scanned documents can be saved is also an important issue and here uses HTML.

Support for Unicode Fonts

The OCR software under consideration should also support Unicode fonts and this is the main theme of this research.

File Enhancement Features

The quality and conditions of the original documents affects the OCR processing process. The OCR software must have facilities for removing discolorations and improving contrast. Noise filtering and image enhancement is implemented and tested.

Availability of advanced features

The availability of advanced features such as spell-checkers and WYSIWYG editors should also be considered. But this is future work and not included in this research.

The functionality of the program is just from different font and types of document to convert to editable and searchable form of document with a normal font, font style, and font size. This means that the destination font is Nyala. The SVM classification diagrammatical representation looks like as follows. The different colors represent different classes. In this instance there are 15 different class and these classes in the diagram represented by 15 different colors. Just to illustrate the classification.

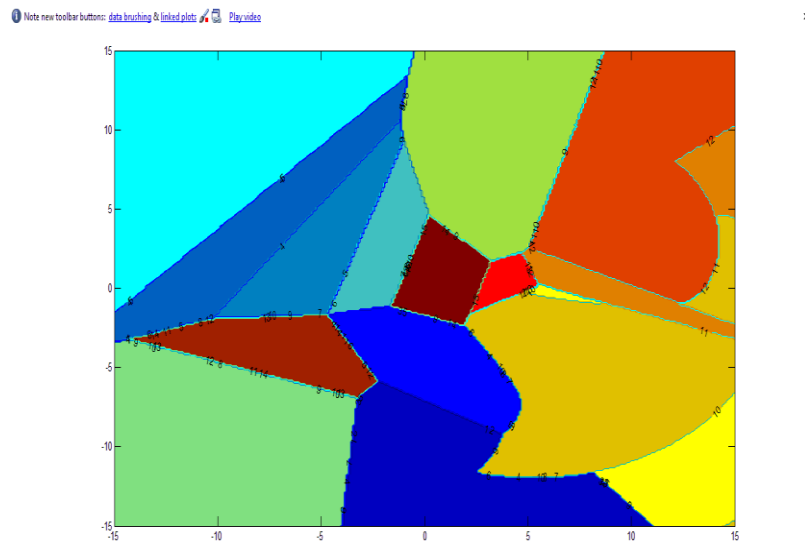


Figure 32 diagrammatical illustrations for classifications

The training set of the program is collected from the entire available font in Ethiopia and this make the research power full to support conversion documents from all previously developed fonts to a Unicode standard fonts.

Chapter 6

Conclusion

The Amharic language is the Official language of over 70 million people mainly in Ethiopia. The Amharic script has 33 basic characters each with seven orders giving 310 distinct characters, including numbers and punctuation symbols. The characters are visually similar; there is a typeface, but no capitalization. Beside this there is no any standard font to use the language in the computer but they use different fonts developed by different stakeholders without keeping a standard, on their own way and interest, and this create a problem of incompatibility between different fonts and documents.

An extensive literature survey and the government report reveal no single Amharic character recognition is found in the Ethiopia, and also Ethiopic script optical character recognition software is not addressed by local researcher and international software developers. It is a very important application for the country but it is the least developed ICT discipline. By the fact that stated above a two stage activity is designed to solve the problem.

First to study the reason out *why* Ethiopic script optical character recognition software is not addressed by local researcher and international developers. And the major reason that identified by this work are

1. The total number of the characters are large,
2. The availability of visually similar characters,
3. The absence of responsible person who promote the language in the world for the international software developers and
4. Luck of cooperation between developers and local researchers are the major reason identified in the literature.
5. And the most one is luck of a standard font for the language.

The second activity is to implement and test the applications by challenging the problem and to put the base line for local and international software developers.

To solve this problem 3 alternative solution is proposed and the best solution is selected and implemented. The best solution is feasible economically, schedualy, operationally, and technically. To implement the newly proposed solution a two stage system is designed first training stage the second one is testing and prediction stage. For training machine learning tool is chosen that is SVM. SVM was introduced by Vapnik in 1992, which has been quickly gained attention due to a great number of theoretical and computational merits. SVM roots in statistical learning theory and follows the principle of structural risk minimization to control the generalization ability of a learning machine. To solve a classification problem, SVM constructs a feature space by using a kernel function, and separates the data into categories in the feature space. For the training activity a database of training image is prepared and for this thesis 15 character only selected. And for each character 100 different samples and total 1500 training dataset is prepared, 1200 for training and 300 for testing. A PCA dimension reduction tool is applied and feeded the data to the SVM for training. The SVM uses Gaussian radial base Function for the kernel trick and fixes important parameter for classification and prediction. The size of each image is 30 * 30 and in bmp file format. The system is trained and training result analysis is found under the result analysis chapter in [Table 7].

The second stage is prediction and testing, for this stage a scanned document is prepared written by different local fonts and image enhancement and skew detection is applied if it is important then the document will be loaded to the system and the system segmented and changed to a binary image and the binary image will be converted to a vector image then will be inputted to the SVM for testing and prediction then the class label will be returned as an output or prediction then the class label or the desired output will be mapped to the lookup table, from the lookup table the corresponding hexadecimal or Unicode value returned and the browser will interpreted the input hexadecimal value to actual character. And finally the result will be displayed by browser or saved permanently in the secondary storage media. This described in Chapter 4 under the mapping section in [Figure 21].

The work is successful, the paper identifies major reason and tries to solve the problems and presents technical specification and experiment and result analysis. For this work the feature and facilities of Microsoft windows vista and 7 were very important. The application developed using a Unicode standard. The presences of Unicode standard for the country is also another important benefit to develop different application like OCR by getting rid of the problems of the lack of standard fonts. Before the release of Microsoft windows vista and windows 7 there were no such facility for Ethiopian scripts but nowadays the feature and facilities of those operating system decreases the development cost.

As stated in the previous chapter under result analysis the performance evaluation shows that the system is successful.

This project has implemented an Ethiopic Script character recognition system based on Unicode standards using SVM classification. It focuses on recognizing character images.

In this work skew detection and correction as well some image enhancement technique is implemented but for full feature software it is not important to develop all this kinds of application because of the current scanner software has such facility and also it is possible to embed third party software easily.

In general, using the feature and facility of the Microsoft windows vista/7 operating system based on Unicode standard using SVM classification the Ethiopic script recognition system is developed and tested successfully.

References

- B.~Scholkopf, B. (1998) *Advances in Kernel Methods--Support Vector Learning*, Cambridge, MA: MIT press.
- Burges, C. (1998) *A tutorial on support vector machines for pattern recognition*, Boston: Kluwer Academic.
- Campbell, C. (2009) *An Investigtion into Novelty Detection available*, [Online], Available: HYPERLINK "http://www.enm.bris.ac.uk/teaching/projects/2004_05/dm1654/kernel.htm" [2011 February].
- D - L I B, M.A.G.A.Z.I.N.E. (2009) *Analyzing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs*, March, [Online], Available: HYPERLINK "http://www.dlib.org/dlib/march09/holley/03holley.html" [January 2011].
- Ethiopia (2007) *Africa's second most-populous country*, 03 may, [Online], Available: HYPERLINK "http://www.nctimes.com/news/national/article_06b9f13d-293d-5e47-9452-ab8eda278f0e.html" [January 2011].
- IEEE (2007) *Recognition of Modification-based Scripts Using Direction Tensors*, [Online], Available: HYPERLINK "http://www.citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.3845" . [January 2011].
- IEEE, (.D.I. (2008) *Amharic Character Recognition using a Fast Signature Based Algorithm* , [Online], Available: HYPERLINK "http://www.computer.org/portal/web/csdl/doi/10.1109/ICDAR.2005.198" [january 2011].
- IEEE (2008) *Recognition of Printed Amharic Documents*, [Online], Available: HYPERLINK "http://www.computer.org/portal/web/csdl/doi/10.1109/IV.2003.1218014" [January 2011].
- J.P.Lewis (2004) *Tutorial on SVM*, CGIT Lab.
- Leadtools (2010) *Leadtools OCR module API help*, [Online], Available: HYPERLINK "http://www.leadtools.com/help/leadtools/v15/OCR/API/whnjs.htm" [January 2011].

Microsoft (2003) *Office 2003: Microsoft Office Document Imaging Visual Basic Reference*,
[Online], Available: HYPERLINK

"<http://www.microsoft.com/downloads/en/details.aspx?familyid=8F93E445-B1CF-4477-A373-E17417D616BC&displaylang=en>" [January 2011].

Mitchell, T. (1997.) *Machine Learning*, McGraw-Hill Computer science series.

Nuance Communications, I. (2010) *Omnipage Captures Software developers kit*, [Online],
Available: HYPERLINK "<http://www.nuance.com/for-business/by-product/omnipage/csdk/index.htm>" [January 2011].

OCROPUS (2009) *OCROPUS Project Site*, [Online], Available: HYPERLINK
"<http://sites.google.com/site/ocropus/>" [January 2011].

P, D.R.a.H. (1973) *Pattern Classification and Scene Analysis*, New York: Wiley.

P. SARAGIOTIS, N.P. (2008) 'LOCAL SKEW CORRECTION IN DOCUMENTS',
International Journal of Pattern Recognition, vol. 22, no. 4, pp. 691-710.

science, C.m.s.o.c. (2010) *Tutorial slides by Andrew Moore*, [Online], Available:
HYPERLINK "[Http://www.cs.cmu.edu/~awm](http://www.cs.cmu.edu/~awm)" [February 2011].

Shawe-Taylor, N.C.a.J. (2000) *An Introduction to Support Vector Machines and Other
Kernel-based Learning Methods*, Cambridge : Cambridge University Press.

Unicode, I. (1991-2011) *Unicode 6.0 Character Code Charts*, [Online], Available:
HYPERLINK "<http://www.unicode.org/charts/PDF/U1200.pdf>" [January 2011].

V. Vapnik, S.G.a.A.S. (1997) *Support vector method for function approximation, regression
estimation, and signal processing*. In M. Mozer, M. Jordan, and T. Petsche, editors,
Advances in Neural Information Processing Systems 9, Cambridge, MA: MIT Press.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. Springer, N.Y.: ISBN.