

**Jimma University
Jimma Institute of Technology
Faculty of Computing and Informatics
Information Science Department**



**MSc. In Information Science
(Information and Knowledge Management)**

**MSc. Thesis Research
On**

**Offline Handwritten Text Recognition of Historical Ge'ez
Manuscripts Using Deep Learning Techniques**

**By
Mesfin Geresu Gurmu**

**Principal Advisor
Dr. Million Meshesha (PhD)**

**Co-Advisor
Ms. Elsabet Wodajo (MSc.)**

**January, 2021
Jimma, Ethiopia**

DECLARATION

I, Mesfin Geresu, declare that this thesis is my original work and has not been submitted for a degree in any other university, and that all source of materials used for the thesis have been fully acknowledged.

Mesfin Geresu Gurmu

This thesis has been submitted for examination with my approval as university advisor.

**Dr. Million Meshesha (PhD)
(Principal Advisor)**

**Ms. Elsabet Wodajo (MSc)
(Co-Advisor)**

**January, 2021
Jimma, Ethiopia**

**Jimma University
Jimma Institute of Technology
Faculty of Computing and Informatics
Information Science Department**

**OFFLINE HANDWRITTEN TEXT RECOGNITION OF HISTORICAL GE'EZ MANUSCRIPTS
USING DEEP LEARNING TECHNIQUES**

By

Mesfin Geresu Gurmu

Name and Signature of the Examining Board

Chairman, Examining Board

External Examiner

Internal Examiner

Dr. Million Meshesha (PhD)

Principal Advisor

Ms. Elsabet Wodajo (MSc)

Co-Advisor

million

ACKNOWLEDGEMENT

I would like first to express my heartfelt gratitude to my principal advisor Dr. Million Meshesha for inspiring me to focus on machine learning as well as for his guidance. I would like also to thank Ms. Elsabet Wodajo, co-advisor, for her consistent follow-up and encouragement.

Moreover, heartfelt thanks to Shiferaw Tegen (University of Gondar) for providing me a collection of digitized historical Ge'ez manuscripts. Special thanks to Mesay Samuel (Arba Minch University) and Mohammed Adem (senior lecturer of linguistics in Jimma University) for their provision of reading materials. Birhanemeskel Tarekegn and Lidet Elias also deserve my special thanks for allowing me to use their office as a Computer Vision lab (really sensed like that).

It also gives me a great pleasure to extend my heartfelt thanks to Lemlem Endrias (*Fiyorina*) and Samuel Tigistu (*Sami Worq*) who supported me financially during the course of this research.

Finally, I would like to thank everyone who assisted me in different forms and put something in this research in one way or another.

DEDICATION

To

All innocent migrants of Ethiopians and Eritreans who have lost their life in the Sahara desert and Mediterranean Sea.

Rest All in Peace!

Table of Contents

ACKNOWLEDGEMENT	IV
LIST OF TABLES	IX
LIST OF FIGURES	X
LIST OF ABBREVIATIONS/ACRONYMS	XI
ABSTRACT	XII
CHAPTER ONE	1
INTRODUCTION	1
1.1. Background of the Study	1
1.2. Statement of the Problem	4
1.3. Research Questions	5
1.4. Objective of the Study	5
1.4.1. General Objective	5
1.4.2. Specific Objectives	5
1.5. Significance of the Study	6
1.6. Scope and Limitation of the Study	7
1.7. Methodology of the Study	8
1.7.1. Document Image Collection	8
1.7.2. Research Design	8
1.7.3. Document Image Pre-processing	8
1.7.4. Document Page Layout Analysis	9
1.7.5. Training OCR engine	9
1.7.6. Performance Evaluation	9
1.8. Operational Definitions	9
1.9. Organization of the Thesis	10
CHAPTER TWO	11
LITERATURE REVIEW	11
2.1. Overview	11
2.2. Stages in Developing an OCR system	11
2.3. Document Image Analysis and Recognition	13
2.3.1. Binarization	13
2.3.2. Skew Estimation	20

2.3.3.	Page Layout Analysis	23
2.3.4.	Handwritten Text Recognition	25
2.3.5.	Summary	28
2.3.	Deep Neural Networks	29
2.3.1.	The Perceptron.....	30
2.3.2.	Multi-Layer Perceptrons (MLP).....	31
2.3.3.	Convolutional Neural Networks (CNN)	33
2.3.4.	Recurrent Neural Networks (RNN).....	35
2.3.5.	Bidirectional Recurrent Neural Networks (BRNN)	35
2.3.6.	Long Short-Term Memory Units (LSTM)	36
2.3.7.	Bidirectional Long Short-Term Memory Units (BLSTM).....	39
2.3.8.	Connectionist Temporal Classification (CTC)	39
2.4.	Training Deep Neural Networks.....	40
2.4.1.	Backpropagation Algorithm	41
2.4.2.	Backpropagation Through Time (BPTT)	42
2.4.3.	Real-Time Recurrent Learning (RTRL)	43
2.4.4.	Other Training Approaches to Deep Neural Networks.....	44
2.4.5.	Generalization and Regularization.....	45
2.4.6.	Summary	47
2.5.	Ge'ez Writing System and Historical Ge'ez Manuscripts	48
2.5.1.	Origin of the Ge'ez Writing System.....	50
2.5.2.	Symbols of the Ge'ez Writing System (<i>Fidälat</i>).....	51
2.5.3.	Order Formation in the Ge'ez Writing System.....	55
2.5.4.	Challenges of OCR for the Ge'ez Writing System.....	56
2.5.5.	Historical Ge'ez Manuscript Collections	59
2.5.6.	Digitization of Historical Ge'ez Manuscripts	61
2.5.7.	Challenges of OCR for Historical Ge'ez Manuscripts.....	62
2.6.	Related Works	66
2.6.1.	Historical Developments of OCR for Ge'ez and Amharic Scripts	67
2.6.2.	List of Research Works on OCR for Ge'ez and Amharic Scripts	68
2.6.3.	Challenges of OCR for Ge'ez and Amharic Scripts in the Era of Deep Learning.....	84
2.6.4.	Future Directions of OCR for Ge'ez and Amharic Scripts.....	85

2.6.5. Summary	87
CHAPTER THREE	89
METHODS AND APPROACHES	89
3.1. Architecture of the Proposed Handwriting Recognition System	89
3.2. Experimental Setup for Pre-processing	90
3.2.1. Document Image Binarization.....	90
3.2.2. Document Image Skew Estimation	91
3.3. Experimental Setup for Page Layout Analysis.....	92
3.4. Experimental Setup for Training OCR Engine.....	93
3.4.1. Training Data Preparation	94
3.4.2. Running the Training Process.....	94
3.4.3. Performance Evaluation.....	94
3.5. Hardware and Software Employed for Implementation	95
CHAPTER FOUR	96
EXPERIMENTAL RESULTS AND DISCUSSION.....	96
4.1. Results and Discussion of the Pre-processing	96
4.1.1. Document Image Binarization.....	96
4.1.2. Document Image Skew Estimation	100
4.1.3. Comparison	102
4.1.4. Discussion of Results	103
4.2. Results and Discussion of the Page Layout Analysis.....	104
4.3. Results and Discussion of Training the OCR Engine.....	113
4.3.1. Training Data Preparation	113
4.3.2. Running the Training Process.....	114
4.3.3. Performance Evaluation.....	116
4.3.4. Comparison	116
4.3.5. Discussion of Results	117
CHAPTER FIVE.....	118
CONCLUSIONS AND FUTURE WORKS	118
5.1. Conclusions	118
5.2. Future Works.....	118
References	120

LIST OF TABLES

Table 2. 1: The core syllables in the Ge'ez writing system (Scelta, 2001).....	52
Table 2. 2: Labialized symbols (Labiovelars) in the Ge'ez writing system (አቤሲሌም, 2012 ዓ.ም).....	53
Table 2. 3: Punctuation marks in the Ge'ez writing system (ተግባሩ, 2008 ዓ.ም)	54
Table 2. 4: Numerals in Ge'ez writing system (አቤሲሌም, 2012 ዓ.ም)	54
Table 2. 5: Composition and total number of symbols in the Ge'ez writing system (ተግባሩ, 2008 ዓ.ም)...	55
Table 2. 6: High inter-class similarity of the Ge'ez script	57
Table 2. 7: The writing convention of words in Ge'ez writing system (ተግባሩ, 2008 ዓ.ም).....	58
Table 2. 8: Research works on OCR for Ge'ez and Amharic scripts [17 th May 1997 – 30 th Sep. 2020].....	70
Table 4. 1: Overall evaluation results and the final ranking of the binarization methods	98
Table 4. 2: Evaluation results of binarization for each test image with respect to the metrics used	98
Table 4. 3: Frequency table of the ground truth skew angle distribution	101
Table 4. 4: Frequency table of estimated skew angles using Hough transform.....	102
Table 4. 5: Evaluation results and comparison of skew estimation methods	103
Table 4. 6: Page layout evaluation results of success rate for each document images.....	112

LIST OF FIGURES

Figure 2. 1: Multi-Layer Perceptron: x_1, \dots, x_N are the inputs, $\mathbf{W}(i)$, $\mathbf{b}(i)$ are the weight matrix and bias vector i (Bluche, 2015).....	32
Figure 2. 2: The simple form of Recurrent Neural Networks (Bluche, 2015)	35
Figure 2. 3: Neurons for RNNs: (a) Simple Neuron (b) LSTM unit (Bluche, 2015)	37
Figure 2. 4: Multi-Layer Perceptron training by backpropagation of the error (Bluche, 2015).....	42
Figure 2. 5: Backpropagation Through Time (Bluche, 2015)	42
Figure 2. 6: The language family of the Ge'ez language (አብዚኦታዊ, 2012 ዓ.ም)	51
Figure 2. 7: Degradations and decoration in the Four Gospels of Endä Abbä Garimä, 4th - 6th century manuscript (courtesy to Bausi <i>et al.</i> 2015, photograph by EBW).....	63
Figure 2. 8: Drawing and ornaments in the 14th Century historical Ge'ez manuscript (courtesy to FDRE, Ministry of Tourism and Culture).....	64
Figure 2. 9: Marginal notes in the 19th Century historical Ge'ez manuscript (courtesy to FDRE, Ministry of Tourism and Culture)	65
Figure 2.10: Research works of OCR for Ge'ez and Amharic scripts.....	67
Figure 2.11: Research works of OCR for Ge'ez and Amharic scripts based on the writing mode	70
Figure 2.12: Research works of OCR for Ge'ez and Amharic scripts based on the recognition level	82
Figure 2.13: Research works of OCR for Ge'ez and Amharic scripts based on the type of dataset	83
Figure 3. 1: Architecture of the proposed handwriting recognition system	89
Figure 4. 1: Example of binarization results of Sauvola's method.....	100
Figure 4. 2: The Histogram of angle values for the DISEC'13 experimental dataset	101
Figure 4. 3: The Histogram of the estimated skew angle values using Hough transform	102
Figure 4. 4: Page layout description of each document image in the testing set.....	106
Figure 4. 5: Region classification and reading order of each image in the testing set	108
Figure 4. 6: Sample of text line detection in Image1	109
Figure 4. 7: Sample of word detection in Image2	110
Figure 4. 8: Ground truth of Image3 created manually using Aletheia tool.....	111
Figure 4. 9: Snapshot of the Ground truth XML file of Image1.....	112
Figure 4. 10: Sample of scanned pages.....	114
Figure 4. 11: Sample of text line images with their corresponding ground truth	114
Figure 4. 12: The output on the terminal after the training is finished	115
Figure 4. 13: The output on the terminal that shows the lstmtraining	115
Figure 4. 14: Comparison of OCR performance by the base and new models	117

LIST OF ABBREVIATIONS/ACRONYMS

AED	Average Error Deviation
ANN	Artificial Neural Network
BLSTM	Bidirectional Long Short-Term Memory
BPTT	Backpropagation Through Time
BRNN	Bidirectional Recurrent Neural Networks
CE	Correct Estimation
CER	Character Error Rate
CNN	Convolutional Neural Networks
CTC	Connectionist Temporal Classification
DIBCO	Document Image Binarization Contest
DISEC	Document Image Skew Estimation Contest
DRD	Distance Reciprocal Distortion
FM	F-Measure
HMM	Hidden Markov Model
ICA	Independent Component Analysis
LSTM	Long Short-Term Memory
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PSNR	Peak Signal-to-Noise Ratio
RNN	Recurrent Neural Network
RTRL	Real-Time Recurrent Learning
SVM	Support Vector Machines
TBPTT	Truncated Backpropagation Through Time

ABSTRACT

Handwriting recognition of historical documents is still largely unsolved problem in the field of pattern recognition. This thesis investigates how the-state-of-the-art deep learning techniques perform handwriting recognition in the context of historical Ge'ez manuscripts. Though Ge'ez was the language of literature in Ethiopia until the middle of the 19th century, it is underrepresented in the research areas of document image analysis and recognition. Thus handwriting recognition system is proposed based on real-world large scale digitization scenarios. Its architecture is comprised of tasks, namely: pre-processing (binarization and skew estimation), page layout analysis, recognition model, and post-processing. For each task, experimental setup is designed. In the task of binarization, four binarization methods (Otsu's global method, Otsu's local method, Sauvola's method and Gato's adaptive method) were investigated using FM, p-FM, PSNR and DRD evaluation metrics. Sauvola's method outperforms all other methods on all the metrics. In the document image skew estimation task, Hough transform based method was investigated by experimenting and examining the results over a dataset. Evaluation criterion AED, TOP80, and CE were used and obtained values equal to 0.3115, 0.058, and 76.00 respectively. In the page layout analysis task, the performance of Leptonica which is open source C library was investigated and achieved results with high success rate on region and text line level over a wide variety of page layouts of actual historical Ge'ez manuscripts. The final experimental setup was designed for building a recognition model using Tesseract OCR engine. Due to a difficulty to prepare large training data with ground truth from actual historical documents, fine tuning approach was proposed and applied in the context of historical Ge'ez manuscripts. A total of 257 text line images collected from 15 different pages were prepared and able to build a recognition model with character error rate of 2.632%. Overall, the performed experiments with the prototyping approach have produced encouraging results so that a complete OCR system development for historical Ge'ez manuscripts is applicable. The major weakness of the study is optimization. Therefore, further optimization technique with large training sample is required. Furthermore, as a future work, investigation needs to consider incorporating post-processing into the recognition process.

CHAPTER ONE

INTRODUCTION

1.1. Background of the Study

Handwriting is a concatenation of graphical symbols drawn using a hand to represent linguistic constructs for communication and knowledge storage. These graphical marks or writing symbols have deep orthographic relation to the phonology of a spoken language. However, to a machine or computer, handwriting is nothing but a pattern (Adak, 2019). The patterns can be observed as internal relationships within the pixels of a document image. Therefore, recognition of this pattern is performed in order to read a manuscript by a computer. The process of automatic pattern recognition of characters from an optically scanned document image is known as Optical Character Recognition (OCR) (Plamondon & Srihari, 2000). OCR works by involving the extraction of features and discrimination or classification of these features based on patterns. These patterns are highly based on the nature of the input data.

Based on the writing generation strategy and data processing, the handwritten input data for handwriting recognition can be broadly categorized into two modes, i.e., *offline and online* (Plamondon & Srihari, 2000; Adak, 2019). The offline handwritten input data for handwriting recognition is a static data and generated from scanned images while the online handwritten input data is dynamic and its generation is based on the movement of pen tip having certain velocity, projection angle, position and locus point. In the online mode, the data is captured at the time of writing using a digital device, such as PDAs. In this study, however, we mainly deal with offline handwriting. For a comprehensive survey on online and offline handwriting recognition, refer to (Plamondon & Srihari, 2000).

Offline handwriting recognition to historical documents, in particular, is a complex task mainly due to low document quality and various complex page layouts. It can be defined as a task to recognize handwritten text in order to generate a transcript of a given document (Fischer, 2012). Manual transcribing is a laborious job, and requires expertise and a fair amount of time with keen attention. Therefore, there is a growing interest in

pattern recognition for automatic handwriting recognition in order to ease this transcript generation task. However, handwriting patterns are complex due to the challenges of their multifold variations. Since the early time, hence, automatic handwriting recognition has become an important research topic in the areas of image and pattern recognition (Adak, 2019). It has also been a major research problem for several decades and has gained attention in recent times due to the potential value that can be unlocked from extracting the information stored in historical documents.

Handwriting recognition uses various pattern recognition approaches which are known as template matching, statistical, structural and syntactical for feature extraction and classification tasks (Fischer, 2012). More recently, however, deep learning techniques and methods derived from statistical learning theory have been receiving increasing attention in pattern representation. Unlike simple artificial neural networks, deep learning is not only used for the mapping from representation to output but also to learn the representation itself. This approach is known as representation learning (Goodfellow, Bengio, & Courville, 2016). Learned representations often result in much better performance than can be obtained with hand-designed representations. Thus, the powerful automatic feature extraction ability of deep learning reduces the need for a separate handcrafted feature extraction process. The recently released multilingual Tesseract¹ OCR engine by Google, for instance, is also purely implemented using deep learning techniques. Some other successful application areas of deep learning include image classification, object detection, video processing, natural language processing (NLP), and speech recognition (Goodfellow, Bengio, & Courville, 2016).

Generally, deep learning techniques give a computer the ability to acquire its own knowledge by extracting patterns from raw data. The application of deep learning in the handwriting recognition problem allows the learning algorithm to learn the underlying relationships in the raw pixels of the input image data. In this thesis, handwriting recognition is formulated as a sequence of pattern classification problem where an optically scanned 2D (two-dimensional) digital handwriting document images is used as a raw data.

¹ <https://tesseract-ocr.github.io/>

There are three (3) key challenges in this research, namely: (a) handwriting variability, (b) processing of low quality degraded documents, and (c) need of training samples.

The first challenge, i.e. handwriting variability is the major challenge in handwriting recognition. Handwriting varies intrinsically due to time, space (geographic location), culture, etc. Such variation among individuals can be referred as inter-variability. However, the handwriting of a particular individual may also vary due to various factors, namely: mechanical (e.g., writing instrument, writing surface, etc.), physical (e.g., illness, aging, etc.), psychological (e.g., excitement, anger, mood, etc.). This type of variation can be referred as intra-variability. The factors of variation are major sources of difficulty in the recognition process.

The second challenge relates with the writing medium and writing mode. The writing medium is a joint venture of tools and materials. The tools are the utensils used for writing, e.g., quill, pencil, pen, etc. Writing is performed on a sheet made of e.g., parchment, papyrus, paper, etc. The inks used for writing can be made from iron salts, tannic acids, pigments of plant's leaves, etc. In addition to this, bad image condition due to damaged parchments, faded ink and ink bleed-through may also occur. Such variation in the quality of the writing medium poses a challenge in handwriting recognition. Similarly, the type of the writing mode, i.e. the writing generation, has also its own impact. Due to the limitations in the capabilities of different digitizers and noises introduced by the digitizers, representations, and etc., the study of handwriting recognition has become a challenging problem.

The third challenge is the need of training samples. We need representative training samples of handwriting in order to use deep learning techniques. The creation of training samples for handwriting recognition from actual historical documents requires much human effort. There are very few publicly made available datasets, but only for the world's major languages, such as for Latin, English, Greek, Arabic, Chinese, Devanagari, etc. However, to the researchers' knowledge, there is no publicly made freely available dataset in the context of historical Ge'ez manuscripts. From in-depth exploration of the literature, in fact, it is well observed that historical Ge'ez manuscripts are underrepresented in the research areas of document image analysis and recognition.

1.2. Statement of the Problem

Handwriting recognition of historical documents is still largely unsolved problem in the field of pattern recognition. This thesis investigates how the-state-of-the-art deep learning techniques perform handwriting recognition in the context of historical Ge'ez manuscripts. The inspiration for doing this research came after learned about the DATECH² (Digital Access to Textual Cultural Heritage) conferences held in different times. The recent significant digitization projects in Europe working to improve the creation, transformation and exploitation of historical documents in digital form have been the theme of the events. Thus this research is proposed with a motivation to implement the good practices of these research projects towards large scale digitization of historical Ge'ez manuscripts. Though Ge'ez was the language of literature in Ethiopia until the middle of the 19th century (Worku & Fuchs, 2003), it is underrepresented in the research areas of document image analysis and recognition.

Extensive study of the previous related research works reveals that only four studies i.e., (Yaregal & Bigun, 2008), (Siranesh, 2016), (Shiferaw, 2017) and (Fitehalew, 2019) attempted to apply OCR to historical Ge'ez manuscripts. In light of the pattern recognition techniques, the studies applied hybrid (structural/syntactical approach), deep Multilayer perceptron (MLP), statistical approach, and deep convolutional neural networks (CNN), respectively. It can be observed that two of the above mentioned research works, i.e. (Siranesh, 2016) and (Fitehalew, 2019), purely implemented deep learning techniques. However, all of them focused towards character level recognition, i.e. none of them formulated the handwriting recognition problem as a sequence of pattern classification problem on text line level. In terms of the pre-processing step, none of them applied objective evaluation method which accounts for the performance of the binarization and skew estimation tasks. In addition, page layout analysis was not considered in all previous research works of OCR for historical Ge'ez manuscripts. Because it is often not sufficient to simply segment the scanned pages into text and non-text areas to proceed with OCR.

² <https://www.digitisation.eu/event/datech-2019-international-conference/>

Hence a detailed page layout analysis, i.e. page segmentation and region classification is required.

1.3. Research Questions

The study investigates how deep learning techniques perform handwriting recognition in the context of historical Ge'ez manuscripts.

- What suitable pre-processing techniques to use for improving the quality of historical documents in order to improve recognition performance?
- What suitable page layout analysis methods to apply in order to extract the region of interest to proceed with OCR?
- How to tune and enhance generalization and representational capacity of deep neural networks?
- To what extent the prototype handwriting recognition performs?

1.4. Objective of the Study

1.4.1. General Objective

The main objective of the study is to design offline handwritten text recognition of historical Ge'ez manuscripts using deep learning techniques.

1.4.2. Specific Objectives

The study attempted to achieve the following specific objectives, which are directed towards answering the research questions and achieving the general purpose of the study.

- To perform pre-processing tasks (binarization and skew estimation) on degraded historical documents in order to improve the recognition process.
- To perform page layout analysis and study the effectiveness of page segmentation and region classification.
- To create ground truth training samples for deep learning.
- To construct deep neural network model for the recognition of offline handwritten historical Ge'ez manuscripts.

- To evaluate the performance of the constructed model for handwritten historical Ge'ez manuscripts recognition.

1.5. Significance of the Study

Handwriting recognition is an important research problem primarily for organizations attempting to digitize large volumes of handwritten scanned documents. Hence the study has several significances. There are four (4) major beneficiaries of the study, namely: (a) Digital libraries and archives, (b) Cultural heritage preservation advocates, (c) Scholars in the humanities, computing and informatics, (d) The Ge'ez language

Digital libraries and archives can use handwriting recognition as it allows unrestricted indexing, searching and querying from digitized document image collection by making e-archiving of the manuscripts. This contributes immensely to the advancement of automation process of information storage and retrieval, to enhance retrieval of information through the Internet and other applications, along with text mining and translation into another language.

Cultural heritage preservation advocates can use handwriting recognition in the context of historical document collection since it plays an important role to preserve historical documents so that they can be transferred to the next generation in a well-organized manner. Furthermore, it saves the cultural heritage represented in the handwritten documents from being lost due to degradation of parchments.

The study can also support scholars in the humanities in doing research concerning what has been written in the historical documents. It contributes also something to other scholars such as historians, anthropologists, archeologists, politicians, lawyers and social workers by enhancing easy access to the historical documents. On other words, handwriting recognition system allows making the textual content of large number of document images readily accessible to researchers and the public. In addition, scholars in computing and informatics can also use the study in order to benchmark results and uncover specific problems of the methods used in the study and help to improve them.

The Ge'ez language itself also benefits from the study for its development. OCR technology has a profound effect on a language in which it is developed for. Ge'ez

language is one of among the few languages in the world that has survived for more than a century, though its current role is limited to the liturgy service of the ancient Ethiopian and Eritrean orthodox churches. Nowadays, however, many scholars believed that the fate of a language is more likely to be determined by the support it gets from technology. Any language that does not grow in line with the technology likely will perish. In this regard, the current study plays a vital role for the renaissance of Ge'ez language by allowing the historical Ge'ez manuscripts to be available in textual forms.

Overall, when we look at the recent advancement in the field of deep learning, we have no choice but to be involved and committed to study and conduct research in the field. Whatever the nature and the degree of our involvement, we cannot afford not to take advantage and enrich ourselves as well as others who will benefit from the research output or utilize it for social good.

1.6. Scope and Limitation of the Study

The study investigated pre-processing tasks, i.e. binarization and skew estimation, along with page layout analysis before proceeding with OCR. Four binarization methods, namely Otsu's global method, Otsu's local method, Sauvola's method and Gato's adaptive method were investigated using objective evaluation method over a testing set. The testing set with Ground truth was collected from the DIBCO contest held in 2019. It consists of nine (9) images which have representative degradations. In the document image skew estimation task, Hough transform based method was investigated by experimenting and examining the results over a dataset. The dataset consists of 200 samples of different skew angles with Ground truth is employed from the DISEC'13 competition.

In the page layout analysis task, the performance of Leptonica which is an open source C library for efficient document image analysis operations was investigated using a testing set consists of five (5) document pages of historical Ge'ez manuscripts with various complex layouts. In building a recognition model, training samples of 257 text lines from 15 different pages of actual historical Ge'ez manuscripts were prepared for the deep learning application. The manuscripts were collected from parishes and

monasteries found in the North Gondar town, Ethiopia. Due to the novel Coronavirus pandemic, several cities across the world including in Ethiopia were under lockdown. Thus other organizations were not communicated to collect additional manuscripts.

The other limitation of the study is that it does not address the issue on how to choose the architecture of deep neural network for the handwriting recognition problem. In other words, it doesn't provide analysis and guidance for choosing which architecture to use in which circumstances of the handwriting. In addition, it doesn't consider post-processing task such as a language model in the recognition process.

1.7. Methodology of the Study

1.7.1. Document Image Collection

The major tasks that were performed in this step are the selection and collection of the representative historical Ge'ez manuscripts. The manuscripts were collected from parishes and monasteries found in the North Gondar town, Ethiopia. The collected manuscripts were digitized using two methods; cam scanner software that is installed in the Samsung Note 5 mobile with 16MP camera and iPhone 4s with 8MP camera.

1.7.2. Research Design

In this study experimental research is followed since it is very helpful to improve the proposed system through experiment. Experimental research involves a collection of research designs which use manipulation and controlled testing to understand causal processes. Generally, one or more variables are manipulated to determine their effect on a dependent variable. Datasets and algorithms used in the study are independent variables, and the parameters and the performance are dependent variables.

1.7.3. Document Image Pre-processing

This experimental setup is designed with a goal mainly to select the best method for binarization and to investigate skew estimation. Four binarization methods, namely Otsu's global method, Otsu's local method, Sauvola's method and Gato's adaptive method were investigated using objective evaluation method over a testing set. The testing set with Ground truth was collected from the DIBCO contest held in 2019. In the document image skew estimation task, Hough transform based method was investigated by experimenting

and examining the results over a dataset. The dataset is employed from the DISEC'13 competition.

1.7.4. Document Page Layout Analysis

This experimental setup aims to investigate the performance of Leptonica which is open source C library over a testing set. The testing set consists of document images with a wide variety of physical formats and page layouts, such as pages with text regions of multiple columns, marginal notes, decoration and diagrams with degradations.

1.7.5. Training OCR engine

This experimental setup is designed for building a recognition model using OCR engine. Tesseract was selected as OCR engine primarily due to its support to Ethiopic script as well as its open source ethos and popularity with large scale digitization. Moreover, Tesseract is still under active development by Google and its latest version uses LSTM based deep neural networks. The challenge at hand can be formulated as optimization problem, which hypothesized to maximize Tesseract's model accuracy over a set of training samples from actual historical Ge'ez manuscripts. Training Tesseract OCR engine involves three main steps: training data preparation, running the training process, and performance evaluation of the recognition model.

1.7.6. Performance Evaluation

Performance evaluation metric known as Character Error Rate (CER) was used to evaluate the recognition model.

1.8. Operational Definitions

Handwriting recognition: is a task to recognize handwritten text in order to generate a transcript of a given manuscript.

Manuscript: Character formation style broadly categorized into the *manuscript* and *cursive* writing style. In the manuscript writing, each character is distinctly written and typically more legible than cursive which is continuous.

Script: refers to the visual appearance of a writing system. The writing system is used to refer to the principles that guide how symbols are mapped to the language.

Writing: is basically a nexus of graphical symbols, used to *record* the phonology of a spoken language. This *record* is generally called as a “*document*”, which stores and conveys information.

Recognition: is to label/assign unknown handwritten sample into a pre-defined class.

Machine learning: Machine learning is the branch of artificial intelligence that involves creating algorithms that can learn from data.

Deep learning: Deep learning is a type of machine learning that uses deep (or many layered) artificial neural networks.

Deep learning techniques: are speculative ideas that are widely believed to be important for current and future research in deep learning.

Document image: refers to a digital copy of a document gained by scanning or photographing.

OCR: is electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text.

1.9. Organization of the Thesis

This thesis report begins with Chapter 1, introduces the background of the study, and subsequently followed by statement of the problem, objective, significance of the study, scope of the study, methodology, operational definitions, and organization of the thesis. In Chapter 2, extensive literature reviews and related works explore different scientific papers on document image analysis and OCR research works of Ge’ez and Amharic scripts, along with some theoretical backgrounds on deep neural networks. Chapter 3 describes thoroughly the designed experimental setups and approach followed to achieve the goal. Chapter 4 provides results and discussion of the experimental findings. Finally Chapter 5 draws conclusion and presents future extension works.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview

Handwriting recognition is an important research problem for organizations attempting to digitize large volumes of documents, particularly in case of historical documents. In order to preserve and explore historical documents (e.g. in the field of digital humanities), there is a growing need for document image analysis and recognition techniques. However, approaches and techniques selection for building a general purpose OCR system or handwriting recognition system is not a trivial task but requires extensive study of the nature of the script, quality and layout of the document, along with other factors that affect the recognition process directly or indirectly.

OCR can be defined as electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text (Plamondon & Srihari, 2000). OCR works by involving the extraction of features and discrimination or classification of these features based on patterns. These patterns are highly based on the nature of the input data. Based on the writing generation strategy and data processing, the handwritten input data for handwriting recognition can be broadly categorized into two modes, i.e., *offline and online* (Plamondon & Srihari, 2000; Adak, 2019). The offline handwritten input data for handwriting recognition is a static data and generated from scanned images while the online handwritten input data is dynamic and its generation is based on the movement of pen tip having certain velocity, projection angle, position and locus point. In the online mode, the data is captured at the time of writing using a digital device, such as PDAs. In this study, however, we mainly deal with offline handwriting.

2.2. Stages in Developing an OCR system

The first step towards digitization of handwritten document is image acquisition using a device, such as flatbed scanner, hand-held scanner, digital camera, and smartphone which have sensor in optical wavelengths. The captured two dimensional signals are sampled and quantized to yield digital document images. The human perception has the

capability to analyze, integrate, read and understand all the existing information in the digital document images, including, text, drawings, etc. (Acharya & Ray, 2005; Burger & Burge, 2016). However, the idea to impart such capabilities to a machine (computer) in order to interpret the information embedded in the document images requires intelligent optical character recognition (OCR) system.

Various researchers proposed different stages of processing in an OCR system. Kim *et al.* (1999) presented integrated functional modules for handwritten text recognition system. The first step is pre-processing, which concerns introducing an image representation. The second is line separation, concerning text line detection and extracting images of lines of text from document image. This is followed by word segmentation, which concerns isolating words from text line image. The third step is feature extraction and word recognition, concerning handwritten word recognition algorithms; and finally linguistic post-processing, which concerns the use of linguistic constraints to intelligently parse and recognize text.

Shafii (2014) describes the classical work flow of OCR systems as having two major phases: pre-processing and recognition in a linear sequence. The pre-processing phase consists of noise removal, binarization, skew detection, page analysis, and segmentation, while the recognition phase includes feature extraction and classification.

In the past, various algorithms were proposed generally following the above steps for the development of OCR system. In recent years, however, new systems based on deep learning techniques merged the various stages of processing and reflects a change in the architecture of OCR system. The new development in OCR technology is an application of various deep neural networks, particularly Long Short-Term Memory (LSTM) networks which is a type of recurrent neural network (RNN), led to major breakthroughs in handwriting recognition (Goodfellow, Bengio, & Courville, 2016). The recently released Tesseract OCR engine by Google, for instance, is also purely implemented using these ideas. However, OCR system to historical documents is still a problem of science. In fact, the extensive literature review reveals no standard procedure for building handwriting recognition system that targets low quality historical documents with complex page layout.

Therefore, in this thesis, the preferred workflow for designing handwriting recognition system is the one that allows modular and sequential processing of information. After extensive study of the literature, the following modular process in the architecture of handwriting recognition system is proposed for the problem at hand. The proposed system is made based on real-world large scale digitization scenarios. Its architecture is comprised of tasks, namely: pre-processing (binarization and skew estimation), page layout analysis (page segmentation and region classification), recognition model, and post-processing. The subsequent sections of this chapter revolve around these tasks with particular emphasis on the methods and techniques used in the study.

2.3. Document Image Analysis and Recognition

When it comes to handwriting recognition for historical documents, the goal to realize machine reading systems that can match or exceed the capability of the human perception to read, integrate and understand handwritten text is still far from being reached (Shafii, 2014). The artifacts/degradations existing in the original historical documents along with the complexity of the page layout make it difficult for document image analysis and advanced OCR engines to recognize the text accurately. In the past, researchers dealt with the degradation problem during pre-processing using noise removal, document image enhancement, and binarization methods and techniques. In recent years, however, the trend is changed towards a unified process to deal with these problems during the binarization process. Nowadays, different document image binarization techniques have been developed to enhance the quality and be more robust against different types of degradations in historical documents. The subsequent section discusses binarization in detail.

2.3.1. Binarization

Image binarization is the process that converts a given input gray level or color image into a binary representation (Acharya & Ray, 2005). In other words, it is the process of converting a multi-tonal image into a bi-tonal image. Thus, the pixels in a binary image assume only two values, 0 or 1. For instance, in the case of document image binarization, pixels that belong to foreground are assigned a value of 1 and pixels of background have

a value of 0. Mathematical properties of a binary image such as connectivity, projection, area, and perimeter are important components in binary image processing (Acharya & Ray, 2005; Tensmeyer & Martinez, 2020).

For OCR engine, we only need to keep the textual content of the document, so it is sufficient to represent document images in binary format which will be more efficient to process instead of the original grayscale or color images. Many advanced binarization method segments the grayscale document image into text and background by removing any existing degradations, such as contrast, ink bleed-through, large ink stains and non-uniform illumination. It is an important pre-processing step of the document image analysis that can affect subsequent tasks, such as page layout analysis and handwritten text recognition (Tensmeyer & Martinez, 2020). Thus, it is essential to have a robust binarization technique that can correctly keep all essential textual information.

Binarization is also referred as *thresholding* in the document image analysis literature (Sezgin & Sankur, 2004), mainly because a gray level image may be converted to a binary image by the classic thresholding method. The various thresholding methods are commonly categorized in three groups: *global thresholding methods* (e.g. Otsu's method, Kittler and Illingworth's method, and Kapur's method), *local thresholding methods* (e.g. Niblack's method, Bernsen's method, and Sauvola's method) and *hybrid methods* which is a combination of thresholding techniques (Acharya and Ray, 2005; Sezgin & Sankur, 2004).

In global methods of thresholding, threshold selection results in a single threshold value for the entire image (Acharya & Ray, 2005). Global thresholding has a good performance in the case that there is a good separation between the foreground and the background. However, very often, it is not sufficient on low quality and degraded historical document images. In local thresholding methods, unlike global thresholding, a threshold value is assigned for each pixel or a small region of the document image in adaptive manner. These techniques have been widely used in document image analysis because they have a better performance on low quality or degraded document images. The other category is a combination of different techniques for the purpose of a better thresholding result.

According to Lins *et al.* (2017), there is no single algorithm that can perform binarization best for all document image types. Thus, various binarization methods that target a particular degradation type are designed so far. In the last decade, the binarization field has progressed much and new techniques other than thresholding (e.g. conditional random fields and machine learning methods) also have been introduced. In recent times, pixel classification approaches using deep learning models have become the state of the art in historical document binarization (Pratikakis *et al.*, 2017; Tensmeyer and Martinez, 2020). Deep learning models, however, require huge amount of labeled data for training. This makes it difficult for application to limited amount of historical document.

In the subsequent paragraphs, the most popular methods which are also implemented in this study for investigation of the problem at hand are discussed.

A. Otsu's Method

It is a global thresholding algorithm proposed by (Otsu, 1979). The method remains popular, likely because it effectively handles images with uniform background and has no parameters to tune (Tensmeyer & Martinez, 2020). The Otsu threshold, T_{Otsu} , is derived from the histogram of grayscale image intensity values, h , which typically has $L = 256$ bins for 8-bit images. Any chosen threshold $0 \leq T \leq L$ partitions the histogram into two clusters. The number of pixels (Equation 2.1), mean intensity (Equation 2.2), and variance of both clusters (Equation 2.3) are, respectively, given below.

$$w_0(T) = \sum_{i=0}^{T-1} h(i) \quad w_1(T) = \sum_{i=T}^{L-1} h(i) \quad (2.1)$$

$$\mu_0(T) = \frac{1}{w_0} \sum_{i=0}^{T-1} ih(i) \quad \mu_1(T) = \frac{1}{w_1} \sum_{i=T}^{L-1} ih(i) \quad (2.2)$$

$$\sigma_0^2(T) = \frac{1}{w_0} \sum_{i=0}^{T-1} h(i)(i - \mu_0(T))^2 \quad (2.3)$$

$$\sigma_1^2(T) = \frac{1}{w_1} \sum_{i=T}^{L-1} h(i)(i - \mu_1(T))^2$$

T_{Otsu} is then defined as the threshold that minimizes within-cluster variance:

$$T_{Otsu} = \underset{T}{\operatorname{argmin}} w_0(T)\sigma_0^2(T) + w_1(T)\sigma_1^2(T) \quad (2.4)$$

or equivalently maximizes the between-cluster variance, which reduces to:

$$T_{Otsu} = \underset{T}{\operatorname{argmax}} w_0(T)w_1(T)(\mu_1(T) - \mu_0(T))^2 \quad (2.5)$$

Finding T_{Otsu} is done by trying all values of T and seeing which one minimizes Equation 2.4 or maximizes Equation 2.5. Afterwards, binarization is performed globally:

$$B(i, j) = \begin{cases} 0, & I(i, j) < T_{Otsu} \\ 255, & I(i, j) \geq T_{Otsu} \end{cases} \quad (2.6)$$

One disadvantage of Otsu or any other global thresholding method is that the background may be non-uniform, leading to some background pixels being darker than some foreground pixels (Tensmeyer & Martinez, 2020). In fact, Otsu's method can also be performed locally when we do Otsu's method in a local window or some other variant of Otsu's method.

B. Niblack's Method

Niblack (1985) proposed a simple local adaptive threshold, where a threshold is determined for each pixel based on statistics computed from a local window centered on the pixel of interest. Because the threshold is adaptive, it can potentially handle cases of foreground and background intensity distribution overlap. Specifically, Niblack thresholding uses the local mean and local standard deviation:

$$\mu(i, j) = \frac{1}{w^2} \sum_{i'=i-w}^{i+w} \sum_{j'=j-w}^{j+w} I(i', j') \quad (2.7)$$

$$\sigma(i, j) = \sqrt{\frac{\sum_{i'=i-w}^{i+w} \sum_{j'=j-w}^{j+w} (I(i', j') - \mu(i, j))^2}{w^2}} \quad (2.8)$$

where w is called the window size and controls how much context is used to compute these statistics. The per-pixel Niblack threshold is then:

$$T_N(i, j) = \mu(i, j) + k\sigma(i, j) \quad (2.9)$$

where k is a user-set parameter that controls the trade-off between foreground detection precision and recall. The recommended parameter setting is $k = -0.2$, though the optimal k depends on the image and chosen window size.

Binarization is then accomplished with:

$$B(i, j) = \begin{cases} 0, & I(i, j) < T_N(i, j) \\ 255, & I(i, j) \geq T_N(i, j) \end{cases} \quad (2.10)$$

One issue with Niblack is when the window covers only background pixels, it causes the darkest background pixels to be set to foreground. While this noise is often large, the background immediately around the text is correctly identified, which makes Niblack thresholding useful in combination with other binarization techniques (Tensmeyer & Martinez, 2020).

C. Sauvola's Method

Sauvola and Pietikäinen (2000) proposed a variant of Niblack to solve the problem with background-only windows.

$$T_S(i, j) = \mu(i, j) \left[1 + k \left(\frac{\sigma(i, j)}{R} - 1 \right) \right] \quad (2.11)$$

where $\mu(i, j)$ and $\sigma(i, j)$ are computed as in Niblack (Equation 2.7 and Equation 2.8), $k = 0.5$ is the recommended value for the user-set parameter, and R is a constant set to the maximum possible standard deviation, i.e., $R = 128$ for 256 gray levels. While Niblack takes $\mu(i, j)$ and adjusts downward based only on the $\sigma(i, j)$, Sauvola adjusts downward based on (i, j) $\sigma(i, j)$. In windows of only background, $\mu(i, j)$ is relatively large, so $T_S < T_N$, which means fewer of these background pixels are set to foreground.

Though there are various binarization methods that have good records in the document image analysis literature, selection of an appropriate binarization method for the problem

at hand is very difficult. The best method needs to be chosen by experimenting and examining the results in the context of the problem at hand. Therefore, it is very essential to have objective evaluation method which accounts for the performance of the binarization process. However, this requires ground truth construction and automatic evaluation metrics for comparison the performance between algorithms.

In the last decade, a tremendous amount of progress has been made in terms of performance evaluation for binarization. In 2009, the first Document Image Binarization Contest (DIBCO) introduced the first dataset of real degraded images that have ground truth annotations at the pixel level (Gatos, Ntirogiannis, & Pratikakis, 2009). This enabled a standardized performance evaluation that allowed for comparison between binarization algorithms. This encouraged researches in the field and the creation of more publicly available datasets (Tensmeyer & Martinez, 2020). The standard performance evaluation metrics include: F-Measure (FM), pseudo F-Measure (*ps*-FM), Peak Signal-to-Noise Ratio (PSNR), and Distance Reciprocal Distortion (DRD). They are also widely used for evaluation purpose by H-DIBCO which is an international Document Image Binarization Contest dedicated to handwritten document image and organized in the context of ICFHR conference since 2010.

In its latest publication, DIBCO used the following evaluation metrics for comparison between binarization algorithms (Pratikakis *et al.*, 2019).

a) F-Measure (FM)

F-measure (FM) is the harmonic mean of Precision (P) and Recall (R), which in turn are defined by the number of True Positives (TP), False Positives (FP), and False Negatives (FN).

$$FM = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.12)$$

$$where \quad Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP}$$

b) pseudo F-Measure (*ps*-FM)

As introduced by Ntirogiannis *et al.*, (2013) pseudo F-Measure (ps-FM) uses pseudo-Recall $ps-R$ and pseudo-Precision $ps-P$ (following the same formula as F-Measure shown in Equation 2.12). The pseudo Recall/Precision metrics use distance weights with respect to the contour of the ground-truth (GT) characters. In the case of pseudo-Recall, the weights of the GT foreground are normalized according to the local stroke width. Generally, those weights are delimited between [0, 1]. In the case of pseudo-Precision, the weights are constrained within an area that expands to the GT background taking into account the stroke width of the nearest GT component. Inside this area, the weights are greater than one (generally delimited between (1, 2]) while outside this area they are equal to one.

c) Peak Signal-to-Noise Ratio (PSNR)

$PSNR$ is a measure of how close is an image to another. The higher the value of $PSNR$ means the higher the similarity of the two images. Note that the difference between foreground and background equals to C .

$$PSNR = 10 \log \left(\frac{C^2}{MSE} \right) \quad (2.13)$$

$$\text{where } MSE = \frac{\sum_{x=1}^M \sum_{y=1}^N (I(x, y) - I'(x, y))^2}{MN}$$

d) Distance Reciprocal Distortion Metric (DRD)

The Distance Reciprocal Distortion Metric (DRD) has been used to measure the visual distortion in binary document images (Lu *et al.*, 2004). It properly correlates with the human visual perception and it measures the distortion for all the \mathbf{S} flipped pixels as follows:

$$DRD = \frac{\sum_{k=1}^S DRD_K}{NUBN} \quad (2.14)$$

where $NUBN$ is the number of the non-uniform (not all black or white pixels) 8x8 blocks in the GT image, and DRD_K is the distortion of the k^{th} flipped pixel that is calculated using a 5x5 normalized weight matrix \mathbf{W}_{Nm} as defined in (Lu *et al.*, 2004). DRD_K equals to the weighted sum of the pixels in the 5x5 block of the GT that differ from the centered k^{th} flipped pixel at (x, y) in the binarization result image \mathbf{B} (Equation 2.15).

$$DRD_k = \sum_{i=-2}^2 \sum_{j=-2}^2 |GT_k(i, j) - \mathbf{B}_k(x, y)| \times \mathbf{W}_{Nm}(i, j) \quad (2.15)$$

The above standard binarization metrics, however, require annotated pixel based ground truth to evaluate the performance of the binarization algorithm. While the DIBCO framework for ground truth creation allows for efficient user interaction, the heavy use of automation can introduce bias toward the algorithms used for automation. In recent time, hence, there have been some issues and arguments on whether this is a useful form of evaluation. Though the debate on the proper way to evaluate the performance of binarization algorithms continues, the majority of published research continues to evaluate at the pixel level using the DIBCO provided annotations and metrics (Tensmeyer & Martinez, 2020).

2.3.2. Skew Estimation

Skew estimation refers to the detection and correction of document skew which is one of the most important tasks in the document image analysis step of OCR system (Papandreou *et al.*, 2013). In order to proceed with OCR, document image skew correction is essential as a pre-processing step since some degree of skew is unavoidable to be introduced when a document is scanned or captured using digital camera. The skew angle of a document image is defined as the deviation of the dominant orientation of the text lines from the horizontal axis. Skew angle greater than 0.1° may be visible to a human observer. The existence of skew may seriously affect the performance of subsequent processing i.e. page layout analysis and text recognition (Papandreou *et al.*, 2013; Boudraa *et al.*, 2020).

Skew detection and correction is still complex and challenging issue especially for documents with graphics, charts, figures or various font sizes (Boudraa *et al.*, 2020). The extensive study of the literature reveals that various skew detection techniques are available and fall broadly into multiple categories based on the basic approach they adopt. According to Shafii (2014), there are four categories namely: *Projection profile based methods*, *Hough transform methods*, *Nearest-neighbor clustering methods* and *Interline cross correlation methods*. Boudraa *et al.*, (2020), further adds more categories such as

Morphological transform based methods, Analysis of the background of documents images based methods, Statistical mixture model based methods, Principal component analysis based methods, Radon transform based methods, and Fourier transform based methods.

In the subsequent paragraphs, the most popular methods which are also implemented in this study for investigation of the problem at hand are discussed.

A. Projection profile based method (PP)

This method is one of the most popular skew estimation techniques and it was initially proposed by Postl (1986). In this method, histograms of the number of black pixels along horizontal parallel sample lines through the document for a range of angles are calculated. For a non-skewed document, horizontal projection profile will have peaks whose width are equal to the characters' height with maximum peak heights at the text lines and valleys whose width are equal to the between-the-line spacing. Therefore, for each angle a measure of the variation in the bin heights (such as variance) along the projection profile is tracked and the angle with the most variation gives the skew angle.

B. Hough transform method (HT)

Duda and Hart (1972) introduced the Hough transform as a simple linear transform to detect a straight line. In this method, each point in Cartesian space (x, y) is mapped to a sinusoidal curve in $\rho - \theta$ Hough space using the following transform function:

$$\rho = x \cos \theta + y \sin \theta \quad (2.16)$$

When multiple points are on the same line, their transformation will intersect at the same point on the transform plane. Therefore, an accumulator is defined to track number of intersections that sinusoidal curves have at various ρ and θ values. As the number of intersections increases at a particular value of θ so does the possibility of having a line in the original image corresponding to that θ value. Finally, peaks at each ρ value give the angles at which straight lines can be fit through the original pixels. The skew angle of the documents is found by averaging the θ s with highest accumulator peaks.

C. Nearest-neighbor clustering method (NN)

Nearest-neighbor method is based on finding the connected components of document, then finds the histogram of the direction vectors for all nearest neighbors of all components and computes the first nearest neighbor of each component. The angle between centroids of nearest neighbor components is obtained and accumulated in the histogram. To find document skewed angle, the dominant peak is computed (Hashizume *et al.*, 1986).

There are several factors (such as unknown page layout, the range of skew angles, etc.) that restrict the effectiveness of skew estimation methods. The best method needs to be chosen by experimenting and examining the results in the context of the problem at hand. Therefore, it is very essential to have objective evaluation method which accounts for the performance of the skew estimation.

In 2013, the first international Document Image Skew Estimation Contest (DISEC'13) was organized in conjunction with ICDAR conference and provides a dataset which could be considered as a generic benchmarking set (Papandreou *et al.*, 2013). It was also accompanied with performance evaluation metrics which enabled comparison between skew estimation algorithms. In order to measure the performance of the different skew estimation algorithms, according to Papandreou *et al.*, (2013), the following criteria were used: (a) the Average Error Deviation (AED), (b) the Average Error Deviation of the Top 80%, and (c) the percentage of Correct Estimation (CE).

a) The Average Error Deviation (AED)

For every document image j from the given benchmarking dataset the distance $E(j)$ between the ground-truth and the estimation of the algorithm is calculated. The AED criterion is described as:

$$AED = \frac{\sum_{j=1}^N E(j)}{N} \quad (2.17)$$

where N denotes the number of images of the benchmarking dataset.

b) The Average Error Deviation of the Top 80% (TOP80)

For the calculation of the *TOP80* criterion, the distances $E(j)$ were sorted, resulting in an ascending sE list, and the average error deviation is now calculated taking into account only the first 80% values of the images according to:

$$TOP80 = \frac{\sum_{j=1}^M sE(j)}{M} \quad (2.18)$$

where M denotes the number of images.

This criterion imprints the performance of each method excluding cases which we assume that the algorithm can't handle efficiently. In that way, the accuracy of the method is tested in its desired operation status (Papandreou *et al.*, 2013).

c) The percentage of Correct Estimation (CE)

The *CE* criterion is determined as:

$$CE = \frac{\sum_{j=1}^N K(j)}{N} \quad \text{where } K(j) = \begin{cases} 1 & \text{if } E(j) \leq 0.1 \\ 0 & \text{otherwise} \end{cases} \quad (2.19)$$

The threshold of 0.1^0 was chosen due to the fact that a skew angle greater than this threshold may be visible to a human observer (Papandreou *et al.*, 2013).

2.3.3. Page Layout Analysis

A document image is composed of not just pure text but a variety of segments such as text, pictures/drawings, tables, background, etc. For automatic text recognition, segments of the document image need to be separated and then analyzed. This process is commonly known as *page layout analysis* (Shafii, 2014). Binary image representation is essential format for document page segmentation process in machine reading system. In fact, most of the page segmentation algorithms are designed for binary and deskewed document images. Thus, the quality of binarization and skew correction process significantly matters the page layout analysis process.

According to Shafii (2014), the objective of page layout analysis is primarily to carry out page segmentation and region classification, i.e. to group image pixels according to constituent regions or objects. The different methods for page layout analysis are commonly categorized into three groups based on the approach they followed. Top-down,

bottom-up, and hybrid approaches (O’Gorman and Kasturi, 2009; Khurshid, 2009; Shafii, 2014).

Top-down techniques start with the complete document image and divide it repeatedly to form smaller and smaller regions. Top-down techniques include Projection profile methods, histogram analysis, X-Y cut algorithm, space transforms, etc.. These techniques are often fast, but the efficiency depends on a prior knowledge about the class of documents to be processed. In contrast, bottom-up techniques (like connected component analysis, region growing method, run-length smoothing, Voronoi-diagram, the docstrum method, and etc.) start with the smallest components of a document (pixels or connected components) and merging them recursively to form larger, homogenous, regions. They are more flexible but may suffer from accumulation of errors. In addition, one could employ a hybrid approach that uses a combination of top-down and bottom-up strategies. A hybrid approach includes Gabor filter method, wavelet and fractal analysis, and etc.. Each method has its own pros and cons, hence selection of a method highly depends on the type of task for which the segmentation is required (Khurshid, 2009; Shafii, 2014).

In the past various algorithms were proposed generally following the above strategies to perform page layout analysis. In recent years, advanced open source and freeware page layout analysis tools that target document complexities and large scale digitization have been developed. Some of the tools to mention include: Aletheia, LAREX, OCRopus, and OCRFeeder.

In this thesis, Aletheia is employed for the page layout analysis task and ground truth construction. Aletheia is known for its robustness and script independent in the process of document image analysis. Moreover, it can be applied across multiple platforms such as Linux, Windows, and macOS. Its core function is to create and view page segmentation and OCR ground truth. The native storage format is PAGE/pagecontent (XML). Thus it can also be used as a viewer for segmentation and OCR results produced by third party software supporting PAGE (Page Analysis and Groundtruth Elements) format. The recently released Aletheia version 4.1 supports PAGE features including: Page elements

on four levels (regions, text lines, words and glyphs), reading order (page structure), region layers, page collections and performance evaluation (PRImA Research Lab, 2019).

The performance evaluation for page segmentation and region classification is used to benchmark results of layout segmentation methods and uncover specific problems of the algorithms to help developers to improve them. As input: the ground truth XML file, the segmentation result XML file and the black-and-white document image are required. For the evaluation, the ground truth regions are compared to the segmentation result regions. Differences are logged as evaluation errors (such as merge, split, miss, partial miss, misclassification, false detection and overall error). Weights and settings for a specific scenario can be specified using an evaluation profile in Aletheia.

2.3.4. Handwritten Text Recognition

The primary goal of designing pattern recognition system (e.g. handwriting recognition system) is supervised or unsupervised classification (Jain *et al.*, 2000). It requires careful attention to the issues such as, definition of pattern classes, pattern representation, feature extraction and selection, classifier design and learning, selection of training and test samples, and performance evaluation (Jain *et al.*, 2000; Fischer, 2012). Handwriting recognition problem, for instance, can be formulated either as a multiclass pattern classification problem or a sequence of pattern classification problem.

The extensive study of the literature reveals there are various approaches in which pattern recognition techniques categorized traditionally, namely: Template matching approach, Statistical approach, Structural approach, and Syntactic approach (Jain *et al.*, 2000; Fischer, 2012).

Template matching is the simplest and oldest approach to pattern recognition. In template matching, a template (typically, a 2D shape) or a prototype of the pattern to be recognized is available. The pattern to be recognized is matched against the stored template while taking into account all allowable pose (translation and rotation) and scale changes. The similarity measure, often a correlation, may be optimized based on the available training set. Often, the template itself is learned from the training set. Template matching is computationally demanding, but the availability of faster processors has now made this

approach more feasible. The rigid template matching mentioned above, while effective in some application domains, has a number of disadvantages. For instance, it would fail if the patterns are distorted due to the imaging process, viewpoint change, or large intra-class variations among the patterns (Jain et al., 2000).

In the statistical approach, each pattern is represented in terms of d features or measurements and is viewed as a point in a d -dimensional space. For images of handwritten text, typical features include contour position, stroke direction, and center of mass. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d -dimensional feature space. The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated. Given a set of training patterns from each class, the objective is to establish decision boundaries in the feature space which separate patterns belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions of the patterns belonging to each class, which must either be specified or learned (Jain *et al.*, 2000; Fischer, 2012).

In the structural approach, patterns of an object are described symbolically with strings, trees, and graphs. Graphs are the most general models that consist of nodes which are linked with edges. The nodes describe subparts of an object while the edges capture binary relationships of the subparts. Both nodes and edges may be labeled with additional information such as symbols and feature vectors. For handwritten text, a natural graph model could represent start and end positions of individual strokes with nodes and link them with edges labeled with the length and curvature of the stroke. Because of its high representational power, a large number of algorithms have been developed for graph-based object representation (Fischer, 2012).

More recently, however, deep learning techniques and methods derived from statistical learning theory have been receiving increasing attention in pattern representation. Unlike simple artificial neural networks, deep learning is not only used for the mapping from representation to output but also to learn the representation itself. This approach is known as representation learning. Learned representations often result in much better

performance than can be obtained with hand-designed representations (Goodfellow *et al.*, 2016). Thus, the powerful automatic feature extraction ability of deep learning reduces the need for a separate handcrafted feature extraction process. The recently released multilingual Tesseract³ OCR engine by Google, for instance, is also purely implemented using deep learning techniques.

The other important issue needs to be considered in building pattern recognition system is performance evaluation. Particularly in the case of handwritten text recognition, Levenshtein⁴ distance algorithm is widely used for the performance evaluation. Levenshtein distance algorithm, also commonly known as string edit distance algorithm, utilizes dynamic programming for its edit operation. Levenshtein distance is defined as the minimum distance required changing one string into another. The edit distance between two strings is obtained using the edit operations known as insertion (due to errors of spurious symbols), substitution (due to errors of misspelled characters), and deletion (due to errors of lost or missing text). Based on the Levenshtein distance algorithm, Character Error Rate (CER) is commonly used as evaluation metric in handwritten text recognition (Drobac & Lindén, 2020).

CER is computed with the minimum number of operations required to transform the reference text (ground truth) into the output. The larger the number, the more different both texts are. The recognition accuracy is measured in terms of Levenshtein Distance as follows:

$$CER = (i + s + d)/n \quad (2.20)$$

where i denotes the insertion, s denotes the substitution and d denotes the deletion errors. Using the total number n of characters and the minimal number of character insertions i , substitutions s and deletions d required transforming the reference text into the OCR output.

³ <https://github.com/tesseract-ocr/tesseract>

⁴ https://en.wikipedia.org/wiki/Levenshtein_distance

2.3.5. Summary

Handwriting recognition is an important research problem for organizations attempting to digitize large volumes of documents, particularly in case of historical documents. The first step towards digitization of handwritten document is image acquisition using a device which has sensor in optical wavelengths. In this thesis, after extensive study of the literature, modular handwriting recognition system is proposed for historical Ge'ez manuscripts. The proposed system is made based on real-world large scale digitization scenarios. Its architecture is comprised of tasks, namely: pre-processing (binarization and skew estimation), page layout analysis (page segmentation and region classification), recognition (trained model), and post-processing.

Image binarization is a process that converts a given input gray level or color image into a binary representation. Binarization is also referred as thresholding in the document image analysis literature. The various thresholding methods are commonly categorized in three groups: global thresholding methods (e.g. Otsu's method, Kittler and Illingworth's method, and Kapur's method), local thresholding methods (e.g. Niblack's method, Bernsen's method, and Sauvola's method) and hybrid method which is a combination of thresholding techniques. In global methods of thresholding, threshold selection results in a single threshold value for the entire image. In local thresholding methods, unlike global thresholding, a threshold value is assigned for each pixel or a small region of the document image in adaptive manner.

A tremendous amount of progress has been also made in terms of performance evaluation for binarization in the last decade. In 2009, the first Document Image Binarization Contest (DIBCO) introduced the first dataset of real degraded images that have ground truth annotations at the pixel level. This enabled a standardized performance evaluation that allowed for comparison between binarization algorithms. This encouraged researches in the field and the creation of more publicly available datasets. Some of the common standard performance evaluation metrics include: F-Measure (FM), pseudo F-Measure (ps-FM), Peak Signal-to-Noise Ratio (PSNR), and Distance Reciprocal Distortion (DRD).

In order to proceed with OCR, document image skew correction is also essential as a pre-processing step since some degree of skew is unavoidable to be introduced when a document is scanned. The existence of skew may seriously affect the performance of subsequent processing i.e. page layout analysis and text recognition. The extensive study of the literature reveals that various skew detection techniques are available and fall broadly into multiple categories based on the basic approach they adopt. The four common categories include: Projection profile based methods, Hough transform methods, Nearest-neighbor clustering methods and Interline cross correlation methods. In 2013, the first international Document Image Skew Estimation Contest (DISEC'13) was organized in conjunction with ICDAR conference and provides a dataset which could be considered as a generic benchmarking set. It was also accompanied with performance evaluation metrics, namely: (a) the Average Error Deviation (AED), (b) the Average Error Deviation of the Top 80%, and (c) the percentage of Correct Estimation (CE).

The next task is page layout analysis. The objective of page layout analysis is primarily to carry out page segmentation and region classification. The different methods for page layout analysis are commonly categorized in three groups based on the approach they followed. These are Top-down, bottom-up, and hybrid approaches. In the past, various algorithms were proposed generally following these strategies to perform page layout analysis. In recent years, however, advanced open source and freeware libraries for page layout analysis that target document complexities and large scale digitization have been developed.

2.3. Deep Neural Networks

Artificial Neural Networks (ANNs) are popular computational systems for pattern classification problem. They are made from basic processing units, linked to each other with weighted and directed connections, in such a way that the outputs of some units are inputs to the others. The appellation "Artificial" to Neural Networks comes from the similarity between the units of these models and biological neurons. It resembles the brain in two aspects. The first resemblance is that knowledge is acquired by the network through training on representative samples. The other is interneuron connection strengths,

known as synaptic weights, are used to store the acquired knowledge (Haykin, 2009; Goodfellow *et al.*, 2016).

Artificial neural networks (ANNs) grew from research on *artificial neurons*, which were first proposed in 1943 by McCulloch and Pitts (McCulloch & Pitts, 1943). A neuron is an information processing unit that is fundamental to the operation of a neural network, each possibly having a small amount of local memory. The units are connected by communication channels or *connections* which usually carry numeric (as opposed to symbolic) data, encoded by any of various means. Most ANNs have some sort of *training* rule whereby the weights of connections are adjusted on the basis of data. In other words, ANNs *learn* from training samples. If trained carefully, ANNs may exhibit some capability for generalization beyond the training data, that is, to produce approximately correct results for new cases that were not used for training.

Algorithms to adjust the weights of the connections lead to the perceptron which is similar to a typical photo-perceptron that responds to optical patterns as stimuli (Rosenblatt, 1958). Following the perceptron model, various neural network topologies were introduced. Feedforward neural networks (Rumelhart *et al.*, 1986), Recurrent neural Networks (Hopfield, 1982), Convolutional neural networks (LeCun *et al.*, 1989), and Long short-term memory units (Hochreiter & Schmidhuber, 1997) are some of the fundamentals worth mentioning. The subsequent sections discuss each of them in detail.

2.3.1. The Perceptron

Rosenblatt's perceptron, commonly known as *the perceptron*, is the simplest form of a neural network used for binary classification of patterns said to be linearly separable (i.e., patterns that lie on opposite sides of a hyperplane). Basically, it consists of a single neuron with adjustable synaptic weights and bias. The algorithm used to adjust the free parameters of this neural network first appeared in a learning procedure developed by Rosenblatt in 1958 for his perceptron brain model (Rosenblatt, 1958). Indeed, Rosenblatt proved that if the patterns used to train the perceptron are drawn from two linearly separable classes, then the perceptron algorithm converges and positions the decision surface in the form of a hyperplane between the two classes. Therefore, the perceptron

is optimal only when the classification problem can be linearly separated in that space. The synaptic weights of the perceptron can be adapted on an iteration-by-iteration basis. For the adaptation, we may use an *error-correction* rule known as *the perceptron convergence algorithm* (Haykin, 2009).

The perceptron built around a single neuron is limited to performing pattern classification with only two classes. By expanding the output (computation) layer of the perceptron to include more than one neuron, we may correspondingly perform classification with more than two classes. In a layered neural network, the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects directly onto an output layer of computation neurons, but not vice versa. In single-layer perceptrons, 'single-layer' refers to the output layer of computation neurons. We do not count the input layer of source nodes because no computation is performed there.

2.3.2. Multi-Layer Perceptrons (MLP)

Multi-Layer Perceptrons are an example of *feedforward neural networks*, where single layer perceptrons are connected to each other. An MLP contains neurons organized in layers (see figure 2.1). Instead of the single perceptron, several neurons are connected to the same inputs x_1, \dots, x_n , with a different set of weights. The outputs of all these neurons are inputs for a new layer of neurons. Considered altogether, the weights of each neuron k ($\mathbf{W}_{(k)}^{(i)}$), define a weight matrix from layer L_{i-1} to layer L_i : $\mathbf{W}^{(i)}$. Thus the output (vector) of a given layer L_i can be computed as the multiplication of the input vector $y^{(i-1)}$ by the weight matrix $\mathbf{W}^{(i)}$, the addition of a bias vector $\mathbf{b}^{(i)}$, and the element-wise application of a non-linear function f_i (Bluche, 2015).

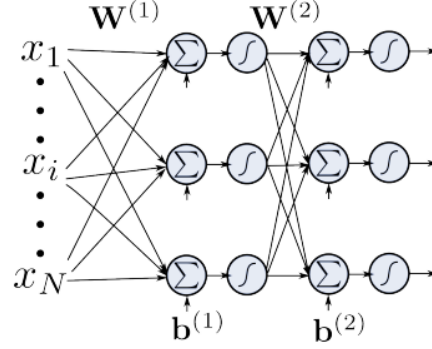


Figure 2. 1: Multi-Layer Perceptron: x_1, \dots, x_N are the inputs, $\mathbf{W}^{(l)}, \mathbf{b}^{(l)}$ are the weight matrix and bias vector l (Bluche, 2015).

$$\begin{aligned}
 y^{(1)} &= f_1(\mathbf{W}^{(1)} \cdot x + \mathbf{b}^{(1)}) \\
 &\vdots \\
 y^{(l)} &= f_l(\mathbf{W}^{(l)} \cdot y^{(l-1)} + \mathbf{b}^{(l)}) \quad (2.21) \\
 &\vdots \\
 y^{(N)} &= f_N(\mathbf{W}^{(N)} \cdot y^{(N-1)} + \mathbf{b}^{(N)})
 \end{aligned}$$

This organization in successive layers has some advantages. Since a given layer only receives inputs from previous layers and provides inputs for next layers, the output of the network can be computed in a single *feed-forward pass*, by sequentially determining the output of each layer. This is also interesting for the backpropagation algorithm, and it is discussed in Section 2.4.1.

An MLP with several outputs is a multi-class classifier. It was shown that the outputs of the network can be interpreted as posterior probabilities (Bourlard & Wellekens, 1988). The Softmax function is often applied as an output layer. For n neurons with activations a_1, \dots, a_n , the Softmax function is defined as follows:

$$z_i = \text{softmax}(a_i) = \frac{e^{a_i}}{\sum_{k=1}^n e^{a_k}} \quad (2.22)$$

With this function, the outputs $z_i \in]0,1]$ sum up to one, i.e., $\sum_{i=1}^n z_i = 1$, and define probability distribution over the different classes, conditioned on the inputs of the network. The advantages of such a property are that the classification is associated with a confidence score with a meaningful interpretation on the one hand, and the network can be a component of a larger system, where the posterior probabilities are important on the other hand, as is the case in hybrid NN/HMM systems (Graves, 2008; Bluche, 2015).

The early recorded attempt of a neural network approach to Amharic OCR was recorded in (Berhanu, 1999). Following it, research works such as (Nigussie, 2000), (Yaregal, 2002), (Mesay, 2003), (Wondwossen, 2004), (Yaregal and Bigun, 2007a), (Yaregal and Bigun, 2007b) and (Abay, 2010) applied neural networks for the pattern recognition task in the development of OCR system for Amharic script. But, all of them applied shallow networks. In 2016, a thesis work by Siranesh (2016) attempted to adopt MLP with deep neural networks (three hidden layers) to OCR for historical Ge'ez manuscripts. The study is considered as the early recorded attempt of adopting deep learning techniques to OCR for Ge'ez script.

2.3.3. Convolutional Neural Networks (CNN)

Convolutional neural networks (CNNs), also known as convolutional networks, are a specialized kind of neural networks for processing data that has a grid-like topology. Examples include image data which can be thought of as a two dimensional grid of pixels. CNNs can be simply defined as neural networks that use a mathematical operation, termed as convolution, in at least one of their layers (Goodfellow *et al.*, 2016). Typical architecture or building blocks of CNNs have two components: *Feature extraction part and classification part*. In the feature extraction part, the network performs a series of *convolutions, pooling (subsampling) and non-linear transformations* during which the features are detected.

The convolution is performed on the input data with the use of a *kernel* to then produce a feature map. The kernel slides across the input feature map. At each location, the product between each element of the kernel and the input element it overlaps is computed and the results are summed up to obtain the output in the current location. The procedure can

be repeated using different kernels to form as many output feature maps as desired. The final outputs of this procedure are called *output feature maps*. The convolution extracts different features of the input. The first convolution layer extracts low-level features like edges, lines, and corners. Higher-level layers extract higher-level features. In contrast, pooling shrinks the dimension of input by an integer factor. Pooling is also known as subsampling and is widely used in deep learning. Pooling layers reduce the dimension and resolution of input while preserving the most important information (Goodfellow *et al.*, 2016).

The CNNs also include other architectural features, namely: stride and zero padding. Stride is the distance between two consecutive positions of the kernel along an axis, whereas zero padding is the number of zeroes concatenated at the beginning and at the end of the axis. A convolutional layer's output shape is affected by the shape of its input as well as the choice of kernel shape, strides and zero padding. Moreover, the relationship between these properties is not trivial to infer. This contrasts with the fully connected layers, whose output size is independent of the input size (Dumoulin & Visin, 2018).

In the recent years, following the breakthroughs that have been gained using deep learning techniques, deep CNNs have become the de facto standard for complex computer vision tasks. Some other successful application areas of deep CNN include image classification, object detection, video processing, natural language processing (NLP), handwriting recognition and speech recognition. The powerful automatic feature extraction ability of deep CNN reduces the need for a separate handcrafted feature extraction process. This ability is primarily owing to the use of multiple feature extraction stages that can automatically learn representations from raw data. The significant improvement in the representational capacity of the deep CNNs is achieved through architectural innovations. For a survey on the different architectures of deep CNNs, refer to (Khan *et al.*, 2020).

The early recorded attempt of adopting deep CNNs to Amharic OCR was recorded in (Birhanu *et al.*, 2018). Following it, research works such as (Abeto, 2018), (Birhanu *et al.*, 2019a), (Fetulhak, 2019), (Mesay *et al.*, 2019), (Fitehalew, 2019), (Halefom *et al.*, 2019)

and (Birhanu *et al.*, 2020) applied deep CNNs for the feature extraction process in the development of OCR system for Amharic and Ge'ez scripts. The thesis work by Fitehalew (2019), in particular, attempted to apply OCR system with deep CNN for historical Ge'ez manuscripts.

2.3.4. Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are networks with a notion of internal state, evolving through time, achieved by cycles or loops in the network. In other words, RNN is a form of neural networks that deals with sequential data (e.g. speech recognition, handwriting recognition, etc.) for modelling. Hopfield networks (Hopfield, 1982) are an early form of recurrent neural network, though the recurrence was used to achieve a stable state rather than process time sequences. In its simplest form, an RNN is an MLP with a recurrent layer. But, the recurrent layer does not only receive inputs from the previous layers, but also from itself (Bluche, 2015). The figure 2.2 below depicted the simple form of RNNs.

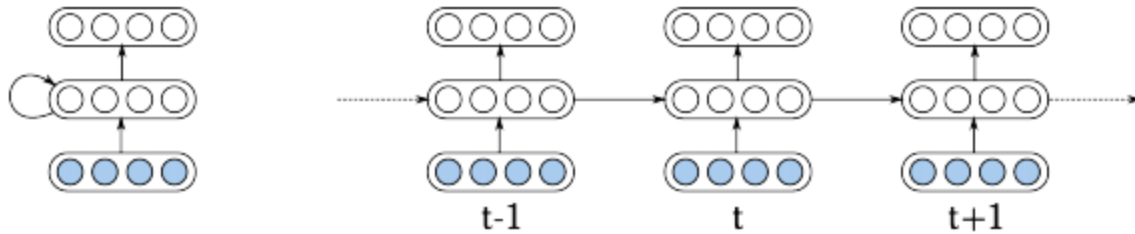


Figure 2. 2: The simple form of Recurrent Neural Networks (Bluche, 2015)

The activations a_k^t of such a layer evolve through time with the following recurrence

$$a_k^t = \sum_{i=1}^I w_{ki}^{in} x_i^t + \sum_{h=1}^H w_{kh}^{rec} z_h^{t-1} \quad (2.23)$$

where x_i s are the inputs and w_{ki}^{in} the corresponding weights, and z_h^{t-1} the layer's outputs at the previous timestemp and w_{kh}^{rec} the corresponding weights.

2.3.5. Bidirectional Recurrent Neural Networks (BRNN)

RNN in its simplest form has a causal structure only. In a standard RNNs, in other words, the state at time t only captures information from the past, $x(1), \dots, x(t-1)$, and the present input $x(t)$. However, in many applications we want to output a prediction of $y(t)$

which may depend on the *whole input sequence*. For example, in speech recognition, we may have to look far into the future and the past for the correct interpretation of the current sound as a phoneme. This is also required in many other sequence-to-sequence learning tasks, such as handwriting recognition (Goodfellow *et al.*, 2016).

Bidirectional recurrent neural networks (BRNNs) were invented by Schuster and Paliwal (1997) so as to process the sequence in both directions. In these networks, there are two recurrent layers: a forward layer, which take inputs from the previous timestep, and a backward layer, connected to the next timestep. Both layers are connected to the same input and output layers (Bluche, 2015). The standard RNN architectures including BRNNs are explicitly one dimensional, meaning that they are suitable for one spatio-temporal dimension. Graves *et al.*, in 2007, introduced multi-dimensional recurrent neural networks (MDRNNs) which process an input image with four directions in recurrent layers, thereby extending the potential applicability of RNNs to vision, video processing, medical imaging and many other areas (Graves *et al.*, 2007).

The extensive literature survey reveals no record of a research work on OCR that applied architectures of simple RNN, BRNNs or MDRNNs for Ge'ez or Amharic scripts so far.

2.3.6. Long Short-Term Memory Units (LSTM)

The idea behind the standard RNNs and the various architectures of it discussed so far is to deal with sequential data for modelling. But in practice (i.e. during training), RNN suffers from problems known as *vanishing gradient* (i.e. *the gradient tends to zero*) or *exploding gradient* (i.e. *the gradient tends to infinity*), because of the computations involved in the process which was first identified by Hochreiter and Schmidhuber (1997). The vanishing gradient issue prevents the network to learn long time dependencies. To address the issue, Hochreiter and Schmidhuber introduced Long Short-Term Memory units.

There are various architectures of LSTM units. A common architecture is composed of a cell (the memory part of the LSTM unit) and three gates (*an input gate, an output gate, and a forget gate*) for the flow of information, which is controlled by a gating system inside

the LSTM unit. The cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell, and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. Some variations of the LSTM unit do not have one or more of these gates or maybe have other gates. For instance, gated recurrent units (GRUs) do not have an output gate.

Bluche (2015) compared an LSTM cell and a basic recurrent neuron as depicted in the figure 2.3 below. The cell input and all gates receive the activation of the lower layer and of the layer at the previous timestep. The following equations define the behavior of the LSTM unit as described by Bluche (2015).

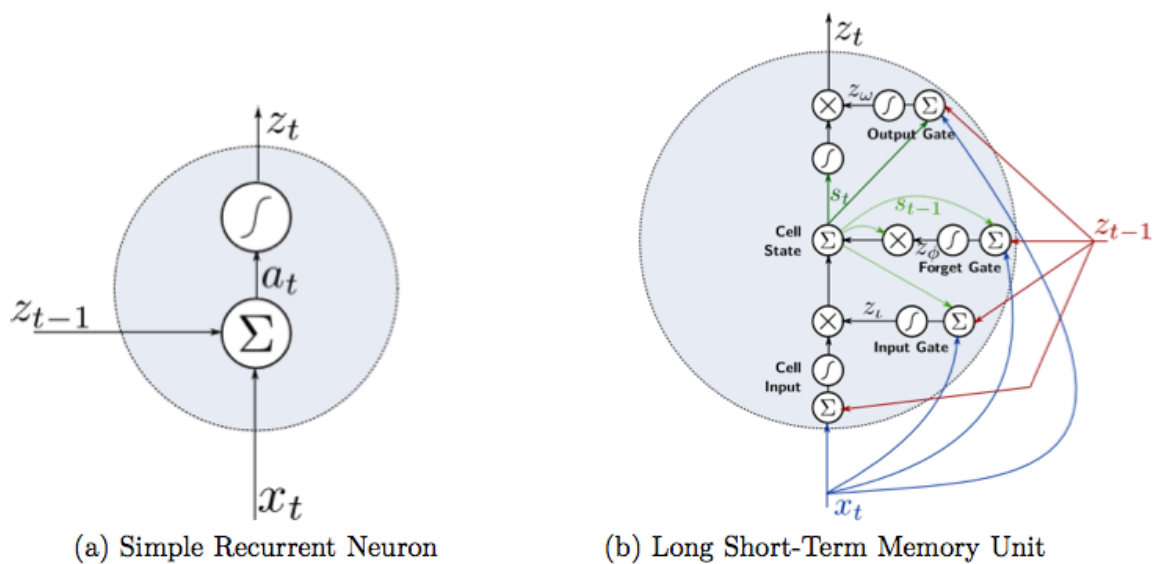


Figure 2. 3: Neurons for RNNs: (a) Simple Neuron (b) LSTM unit (Bluche, 2015)

The Input Gate controls whether the input of the cell is integrated in the cell state

$$a_i^t = \sum_{i=1}^I w_{il}x_i^t + \sum_{h=1}^H w_{hl}z_h^{t-1} + \sum_{c=1}^C w_{cl}S_c^{t-1} \quad (2.24)$$

$$z_i^t = f(a_i^t)$$

The Forget Gate controls whether the previous state is integrated in the cell state, or if it is forgotten.

$$a_{\phi}^t = \sum_{i=1}^I w_{i\phi} x_i^t + \sum_{h=1}^H w_{h\phi} z_h^{t-1} + \sum_{c=1}^C w_{c\phi} S_c^{t-1} \quad (2.25)$$

The Cell state is the sum of the previous state, scaled by the forget gate, and of the cell input, scaled by the input gate.

$$a_c^t = \sum_{i=1}^I w_{ic} x_i^t + \sum_{h=1}^H w_{hc} z_h^{t-1} \quad (2.26)$$

$$S_c^t = z_{\phi}^t S_c^{t-1} + z_i^t g(a_c^t)$$

The Output Gate controls whether the LSTM unit emits the activation $h(S_c^t)$.

$$a_{\omega}^t = \sum_{i=1}^I w_{i\omega} x_i^t + \sum_{h=1}^H w_{h\omega} z_h^{t-1} + \sum_{c=1}^C w_{c\omega} S_c^t \quad (2.27)$$

$$z_{\omega}^t = f(a_{\omega}^t)$$

The Cell output is computed by applying the activation function h to the cell state, scaled by the output gate.

$$z_c^t = z_{\omega}^t h(S_c^t) \quad (2.28)$$

Recent studies show excellent results and great potential of LSTM units in recurrent layers, with BRNN and MDRNN in handwriting recognition, and constitute the state-of-the-art in the handwriting recognition problem (Graves, 2008; Bluche, 2015). LSTM networks have been applied to the newly released open source Tesseract OCR engine that offers support for Unicode with the ability to recognize more than 100 languages including Amharic.

From the previous research works, Direselign *et al.*, (2018) applied LSTM network to OCR system for Ethiopic script. But, the network was trained over on a dataset consisted of synthetic text-line images written in Amharic, Ge'ez and Tigrigna languages in various publications. The extensive literature survey reveals no record of a research work on OCR that applied LSTM to historical Ge'ez manuscripts.

2.3.7. Bidirectional Long Short-Term Memory Units (BLSTM)

The idea behind the bidirectional recurrent neural network (BRNN) is straightforward and discussed in Section 2.3.5. It involves the network to be trained using all available input information in the past and future of a specific time steps. This idea has also been applied to LSTM networks and outperforms unidirectional network architectures in phoneme classification (Graves and Schmidhuber, 2005). In fact, BLSTM networks offer some benefit in terms of better performance in the domains where it is appropriate (e.g. speech recognition). However, it may not make sense to use it for all sequential data modelling.

In more recent time, research works of Birhanu *et al.*, (2019b) and Birhanu *et al.*, (2020) applied BLSTM network to OCR system for Amharic script. In both cases, the network was trained over ADOCR⁵ dataset curated using OCRopus which is open source OCR system. However, the dataset is consisted of printed and synthetic text-line images. The extensive literature survey reveals no record of a research work on OCR that applied BLSTM to historical Ge'ez manuscripts.

2.3.8. Connectionist Temporal Classification (CTC)

Another major breakthrough that has been recorded so far comes from end-to-end deep learning in sequential data modelling, well-known examples include speech recognition and handwriting recognition. According to Graves *et al.*, (2006), RNNs are very powerful dynamic network for sequence learning tasks; but its applicability to address real world problem that needs the prediction of sequences of labels from unsegmented, noisy input data (e.g. handwriting recognition) has been limited. Because such tasks require pre-segmented training data, and post-processing to transform their outputs into label sequences. To address the problem, Connectionist Temporal Classification (CTC) framework was proposed by Graves *et al.* (2006), and corresponds to the task of sequence labelling from unsegmented data with neural networks.

The basic idea behind the CTC framework is that the output of the neural network, when applied to an input sequence, is directly the sequence of symbols of interest (i.e.

⁵ <http://www.dfki.uni-kl.de/~belay/>.

sequence of characters in the case of handwriting recognition). This is different from the previous methods of applying sequence learning tasks, such as in RNNs, where there is one target at each timestep. According to Graves *et al.* (2006), this offers two main advantages: (i) the training data does not need to be pre-segmented, and (ii) the output do not require any post-processing, because it is already the sequence of characters.

Graves (2008) described CTC framework in detail and applied it in speech and handwriting recognition. CTC has also been subsequently applied in several open source OCR systems, such as in Kraken and Tesseract.

In more recent time, research works (Direselign *et al.*, 2018), (Birhanu *et al.*, 2019b) and (Birhanu *et al.*, 2020) applied CTC for sequence labeling with different architectures of deep neural networks to OCR system for Amharic script. All the works were focused towards a recognition level of text-line images. However, the datasets were consisted of primarily synthetic text-line images. The extensive literature survey reveals no record of a research work on OCR that applied CTC to historical Ge'ez manuscripts.

2.4. Training Deep Neural Networks

In deep neural networks, knowledge is acquired by the network through training on representative samples. The procedure used to perform the learning process, i.e. the training, is known as a *learning algorithm*, the function of which is to modify the connection weights of the network in an orderly fashion to attain a desired design objective. Training a deep neural network involves adjusting its parameters, the connection weights, so that the model is able to perform the task at hand. The goal is to optimize a criterion that reflects the quality of the network (Haykin, 2009).

Given such a criterion, one may apply mathematical optimization methods such as gradient descent. The backpropagation algorithm takes advantage of the structure of MLPs to apply these methods. Other algorithms, such as the Backpropagation Through Time (BPTT) algorithm and Real-Time Recurrent Learning (RTRL) were also designed to handle the temporal aspect in RNNs (Bluche, 2015; Haykin, 2009; Sutskever, 2013). Following them, along with other training approaches such as unsupervised and

supervised pre-training to deep neural networks are briefly discussed in the subsequent sections.

2.4.1. Backpropagation Algorithm

Backpropagation is a widely used algorithm in training feedforward neural networks for supervised learning tasks. Though its origin goes back to 1960s, the first documented description of the use of backpropagation in neural nets is attributed to Paul Werbos (1974). Backpropagation was applied to Multi-Layer neural networks by (Rumelhart *et al.*, 1986), and this work showed through experiments that such networks in its hidden layers can learn useful internal representation of data.

Backpropagation learning algorithm involves computing the derivatives of the error with respect to each weight by the chain rule, computing the gradient of one layer at a time, and iterating backward from the last layer to the input layer based on the following formulae:

$$\frac{\partial E}{\partial in_i} = \frac{\partial E}{\partial out_i} \frac{\partial out_i}{\partial in_i} = \frac{\partial E}{\partial out_{i-1}} \quad (2.29)$$

$$\frac{\partial E}{\partial \theta_k} = \frac{\partial E}{\partial out_i} \frac{\partial out_i}{\partial \theta_k} \quad (2.30)$$

The algorithm can be applied when the connections between layers form a directed acyclic graph, and the error is propagated from the outputs to the inputs, as illustrated on the figure 2.4 below.

For in-depth understanding and implementation of the backpropagation algorithm in deep neural networks, refer to (Nielsen, 2019).

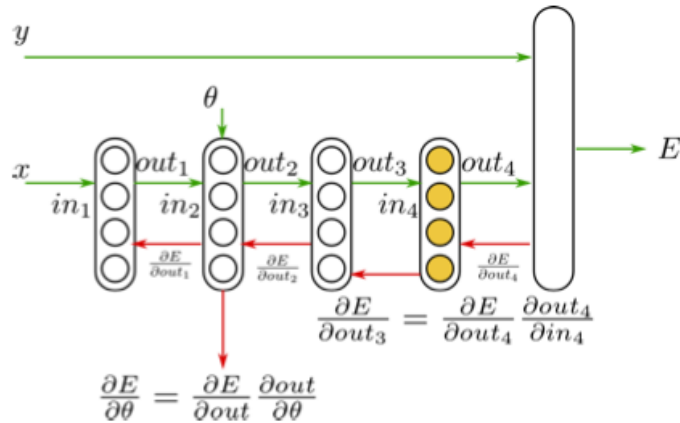


Figure 2. 4: Multi-Layer Perceptron training by backpropagation of the error (Bluche, 2015)

2.4.2. Backpropagation Through Time (BPTT)

Backpropagation Through Time (BPTT) is the application of the Backpropagation training algorithm to recurrent neural networks. In RNNs, BPTT works by unrolling all input time-steps. Thus, a sequential aspect is added to the layered structure of the network. BPTT involves in propagating the error both from the output to the input layer and to the previous time-steps (Werbos, 1990).

When the network is unrolled in time, there are inputs and output layers at every timestep t , so the error for each t should be incorporated in the gradients computations, as shown on the figure 2.5 below.

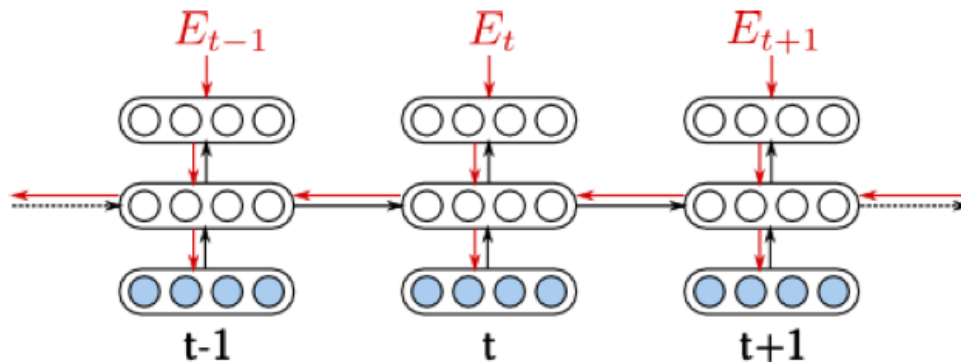


Figure 2. 5: Backpropagation Through Time (Bluche, 2015)

Sutskever (2013) addressed the difficulty of training RNNs and the problems pertaining to BPTT. According to him, one of the main problems of BPTT is the high cost of a single parameter update, which makes it impossible to use a large number of iterations. For instance, if input sequences are comprised of thousands of time-steps, then the same number of derivatives is required for a single weight update. Though the gradients of the RNN are easy to compute, in practice, RNNs also suffers from problems known as vanishing gradient (i.e. the gradient tends to zero) or exploding gradient (i.e. the gradient tends to infinity), because of the computations involved in the process. The vanishing gradient issue prevents the network to learn long time dependencies. To address the issue, Hochreiter and Schmidhuber (1997) introduced Long Short-Term Memory units.

According to Sutskever (2013), the vanishing gradient is undesirable, because it turns BPTT into truncated-BPTT, which is likely, incapable of training RNNs to exploit long-term temporal structure. The vanishing and the exploding gradient problems make it difficult to optimize RNNs on sequences with long-range temporal dependencies, and are possible causes for the abandonment of RNNs by machine learning researchers.

This shortcoming of BPTT is often overcome heuristically, by arbitrarily splitting the initial sequence into subsequences, and only back propagating on the subsequences. The resulting algorithm is often referred to as Truncated Backpropagation Through Time (TBPTT) (Haykin, 2009; Tallec and Ollivier, 2017). In other words, TBPTT is a modified version of the BPTT training algorithm for recurrent neural networks where the sequence is processed one timestep at a time and periodically the BPTT update is performed back for a fixed number of timesteps. Though this comes at the cost of losing long term dependencies in standard RNNs, it is effectively applied in training LSTM networks. In fact, TBPTT is the most widely used variant of BPTT in deep learning for training LSTM networks.

2.4.3. Real-Time Recurrent Learning (RTRL)

Another gradient based learning algorithm for recurrent neural networks is known as Real-Time Recurrent Learning (RTRL). It is an elegant forward-pass only algorithm that computes the derivatives of the RNN with respect to its parameters at each time-step.

Unlike BPTT, which requires an entire forward and a backward pass to compute a single parameter update, RTRL maintains the exact derivative of the loss so far at each time-step of the forward pass, without a backward pass and without the need to store the past hidden states. This property allows it to be more suitable for online continuous training (Haykin, 2009; Sutskever, 2013). For detailed description and implementation of RTRL in deep recurrent neural networks, refer to (Haykin, 2009).

2.4.4. Other Training Approaches to Deep Neural Networks

Modern interest in deep learning began in 2006 when a research paper (Hinton *et al.*, 2006) proposed to train a type of neural network known as deep belief networks (DBN), which have properties that make them interesting for training deep neural networks. DBNs are generative models and can do unsupervised and supervised pre-training to deep neural networks.

According to Hinton *et al.*, (2006) unsupervised pre-training to deep neural networks is an effective technique for initializing the parameters, which greedily trains parameters of each layer to model the distribution of activities in the layer below. The weights are obtained by training a form of neural network known as Restricted Boltzmann Machine (RBM). The training of RBMs is called unsupervised in the sense that we do not need target labels for the training data. The RBMs learn connections between observed variables and latent (hidden) ones, such that the probability of the observations, given by the model, is maximized. Thus, RBMs are generative models explaining the data. The connection weights learnt by this procedure, for each layer, are the initialization of the weights of the neural network. Then, the network can be trained in a classical supervised way (Bengio *et al.*, 2006).

The unsupervised part makes it possible to incorporate a virtually unlimited amount of training data. Since no labels are required, the wealth of data available today, especially through the Internet, can contribute to the creation of robust models. The network obtained after the unsupervised pass, although being a possibly good feature extractor, is not suited to the initial classification problem, which is why supervised fine-tuning is necessary.

Another approach to weight initialization by training, explored in (Bengio *et al.*, 2006), consists in applying supervised pre-training methods. This approach also initializes the weights of one layer at a time. We first build the MLP containing only the first hidden layer and its weight matrix W_1 , and connect it directly to the output layer. This MLP is trained first. Thus, the weights are adjusted so that the produced layer outputs help discriminate the different classes. Instead of waiting for training convergence of this MLP, we stop after a few iterations, e.g. one epoch. First, this is merely a method for weight initialization, and their values will be further adjusted in the next steps, so it is not necessary to waste too much time in this part. Moreover, in the end, we do not want the outputs of this particular layer to be the classification features, but rather to be intermediate features helping to build more complex representations in higher layers. If we wait until convergence of the network, we risk getting activations in the saturated parts of the sigmoid (or *tanh*) function, which will make the final training difficult and slow. Then, we throw away the connections between the hidden and output layers, and add a second hidden layer after the first one. We keep W_1 for the first layer, and randomly initialize a weight matrix W_2 connecting the hidden layers. We repeat the training described above with this 3-layer MLP. Repeating this procedure N times, we get a neural network with N hidden layers, which weights have been discriminately initialized. This network is trained until convergence.

DBNs were influential for several years, but have since lessened in popularity, while models such as feedforward networks and recurrent neural networks have become fashionable. Primarily the reason is that models such as feedforward and recurrent neural networks have achieved many spectacular results, such as their breakthroughs on image classification, speech and handwriting recognition benchmarks (Nielsen, 2019).

2.4.5. Generalization and Regularization

Generalization and regularization are the very essence of learning in deep neural networks. The ability of learning networks to generalize can be greatly enhanced by providing constraints from the task domain.

2.4.5.1. Generalization

In backpropagation learning, we typically start with a training sample and use the backpropagation algorithm to compute the synaptic weights of a multilayer perceptron by loading (encoding) as many of the training examples as possible into the network. The hope is that the neural network so designed will generalize well. A network is said to *generalize* well when the input – output mapping computed by the network is correct (or nearly so) for test data never used in creating or training the network (Haykin, 2009).

A neural network that is designed to generalize well will produce a correct input – output mapping even when the input is slightly different from the examples used to train the network. When, however, a neural network learns too many input – output examples, the network may end up memorizing the training data. It may do so by finding a feature (due to noise, for example) that is present in the training data, but not true of the underlying function that is to be modeled. Such a phenomenon is referred to as overfitting (Haykin, 2009). When the network is overfitting, it loses the ability to generalize the problem.

Generalization is influenced by three factors: (1) the size of the training sample and how representative the training sample is of the environment of interest, (2) the architecture of the neural network, and (3) the physical complexity of the problem at hand (Haykin, 2009).

2.4.5.2. Regularization

Increasing the amount of training data is one way of reducing overfitting. There are other ways we can reduce the extent to which overfitting occurs. One possible approach is to reduce the size of our network. However, large networks have the potential to be more powerful than small networks, and so this is an option we'd only adopt reluctantly. Fortunately, there are other techniques which can reduce overfitting, even when we have a fixed network and fixed training data. These are known as *regularization techniques*. In this section we describe three most commonly used regularization techniques, namely: Weight decay, Dropout, and Early stopping (Haykin, 2009; Goodfellow *et al.*, 2016).

The weight decay technique consists in adding a penalty to the cost function, which depends on the weights of the network. The practical effect of weight decay is that the training procedure will promote solutions with small weights. It is generally observed that

neural networks overfit less with that constraint, which might be explained by the fact that with small weights, the network is less sensible to small changes of the input.

The dropout technique was proposed by Hinton *et al.*, (2012) to reduce overfitting. It consists in randomly ignoring some of the units of the network during training. When dropout is applied to a hidden layer, a sample of units is dropped for each training example, with some probability. The forward pass computes the output of the network without those dropped units and corresponding connections. The backpropagation procedure is performed in this network with missing nodes (Hinton *et al.*, 2012). One of the motivations of dropout is to prevent hidden units to rely on the output of others, and make them useful for classification by themselves. The underlying goal is to reduce overfitting, hence making it a form of regularization.

One manner of controlling the generalization power is to keep a validation set, separate from the training set, on which we can also compute the error. The early stopping method consists in stopping the training procedure when the error on these validation data increases. We may identify the onset of overfitting through the use of cross-validation, for which the training data are split into an estimation subset and a validation subset. The estimation subset of examples is used to train the network in the usual way, except for a minor modification, i.e. the training session is stopped periodically, and the network is tested on the validation subset after each epoch of training.

2.4.6. Summary

In deep neural networks, knowledge is acquired by the network through training on representative samples. The procedure used to perform the learning process, i.e. the training, is known as a learning algorithm, the function of which is to modify the connection weights of the network in an orderly fashion to attain a desired design objective. Backpropagation is a widely used algorithm in training feedforward neural networks for supervised learning tasks. Backpropagation learning algorithm involves computing the derivatives of the error with respect to each weight by the chain rule, computing the gradient of one layer at a time, and iterating backward from the last layer to the input layer.

Backpropagation Through Time (BPTT) is the application of the Backpropagation training algorithm to recurrent neural networks. In RNNs, BPTT works by unrolling all input timesteps. Thus, a sequential aspect is added to the layered structure of the network. Though the gradients of the RNN are easy to compute, in practice, RNNs suffers from problems known as vanishing gradient (i.e. the gradient tends to zero) or exploding gradient (i.e. the gradient tends to infinity), because of the computations involved in the process. The vanishing gradient issue prevents the network to learn long time dependencies. To address the issue, Hochreiter and Schmidhuber introduced Long Short-Term Memory units (LSTM). Another gradient based learning algorithm for recurrent neural networks is known as Real-Time Recurrent Learning (RTRL). It is an elegant forward-pass only algorithm that computes the derivatives of the RNN with respect to its parameters at each timestep.

Generalization and regularization are also the very essence of learning in deep neural networks. A network is said to *generalize* well when the input - output mapping computed by the network is correct (or nearly so) for test data never used in creating or training the network. This problem is referred to as overfitting. There are techniques which can reduce overfitting when we have a fixed network and fixed training data. These are known as *regularization techniques*. The most commonly used regularization techniques are Weight decay, Dropout, and Early stopping.

2.5. Ge'ez Writing System and Historical Ge'ez Manuscripts

The study of the Ge'ez writing system is essential to understand the Ge'ez language and historical Ge'ez manuscripts primarily for the development of OCR system. For successful implementation of pattern recognition techniques in the development of the OCR system, it requires sound knowledge of the script they operate on in terms of shape, number of symbols (the syllables, numerals, and punctuation marks), etc. In the early time, for instance, in a research effort made by Yaregal and Bigun (2006), the nature of prominent structural features of the Ge'ez symbols provides a way to employ structural and syntactic pattern recognition techniques to efficiently design a generic OCR system that invariably works for different font types, sizes and styles. In this section, an attempt is made to

provide such basic information on the Ge'ez writing system in general, and on the nature of the Ge'ez symbols, in particular.

A writing system is built on the smallest meaningful contrastive unit or, basic written symbols, known as grapheme. These basic units are mainly alphabet, syllable, and logogram which form Alphabetic, Syllabic, and Logographic writing systems, respectively. Sometimes, the admixture of these basic units creates further subcategories of the writing system. The world's modern writing systems are mostly classified into five categories, i.e., *Alphabetic* (e.g., Latin, Greek, etc.), *Syllabic* (e.g., Japanese Kana), *Logographic* (e.g., Chinese, Japanese Kanji, etc.), *Abugida* or *Alphasyllabry* (e.g., Indic, Ethiopic, etc.), and *Abjad* or *Consonantary* (e.g., Arabic, Hebrew, etc.) (Adak, 2019). Writing systems are distinguished from other possible symbolic communication systems in that a writing system is always associated with at least one spoken language.

The other important issues that must be noted is that the usage of the term “Ethiopic”, “Ge'ez” and “Amharic” scripts. Ethiopic is a general term coined for Ethiopian Semitic languages (e.g., Ge'ez, Amharic, Tigrigna, Guragegna, etc.). Ethiopic script sometimes interchangeably referred as Ge'ez script, as it is growing out from the 182 core syllables of the ancient Ge'ez language. According to the World Wide Web Consortium⁶, Ethiopic script in its present day form is a multilingual and multinational script comprised of 494 symbols representing: syllables, numerals, punctuation and tonal marks. The Ethiopic range was introduced with version 3.0 of the Unicode standard in 1999. Nowadays, many languages spoken in Ethiopia and Eritrea use the script for writing, but it has been largely used in Ge'ez and Amharic. This can be justified by the fact that Ge'ez was the language of literature in Ethiopia until the middle of the 19th century (Worku and Fuchs, 2003), and currently it is serving as the liturgical language for the ancient Ethiopian Orthodox Tewahedo Church (EOTC), Ethiopian Catholic Church, Eritrean Orthodox Tewahedo Church, and the Bette Israel Jewish Community in Ethiopia (Scelta, 2001; Shiferaw, 2017). Similarly, Amharic language is the working language for the Federal Government of Ethiopia and many regional states in the country.

⁶ <https://www.w3.org/TR/2020/WD-elreq-20200526/>

The main focus of this study, however, is the Ge'ez script. In this study, for the purpose of analysis, the term "Ge'ez" simply refers to the script and not the language. When it compares to its counterparts, such as the Roman script, Ge'ez is significantly larger in size and in scope. It must be acknowledged also that there are no upper or lower case distinctions in Ge'ez writing system, and also there are no ligatures, accented letters, ascent or descent features, and punctuation rules associated with letters (Scelta, 2001; ተግባሩ, 2008 ዓ.ም).

2.5.1. Origin of the Ge'ez Writing System

The Ge'ez script is believed by many scholars to have been derived from the Epigraphic South Arabian script, of Proto-Sinaitic heritage, although there is some dispute surrounding this assertion; some also believe it to have descended from Egyptian hieroglyphics. According to the tradition of the Ethiopian Orthodox Tewahedo Church, the script was divinely revealed to Enos, grandson of the first man, Adam (ተግባሩ, 2008 ዓ.ም). Unlike other Semitic scripts, however, Ge'ez is written from left to right.

The Ge'ez script has evolved through remarkable developments. For instance, vocalization of the Ge'ez occurred in the 4th century, and it is believed many other changes on the script (such as, change in writing direction, order formation, alphabet order change from አ-በ-ገ-ደ to ሀ-ለ-ሐ-መ, and new symbols for numerals) were also occurred during this time. Many scholars believed the symbols for numerals (፩ ፪ ፫ . . .) are derived from the Greek (አቤሲሎም, 2012 ዓ.ም).

According to the language family of contemporary linguistics study, Ge'ez is grouped in the parent Afro-Asiatic category. Many notable scholars in linguistics, such as Baye Yimam, Dessie Qeleb, Yoseph Greenberg, and etc. agree with this classification. Based on this classification, the following figure shows Ge'ez and some other Ethiopian languages which are grouped in the parent Afro-Asiatic category (አቤሲሎም, 2012 ዓ.ም).

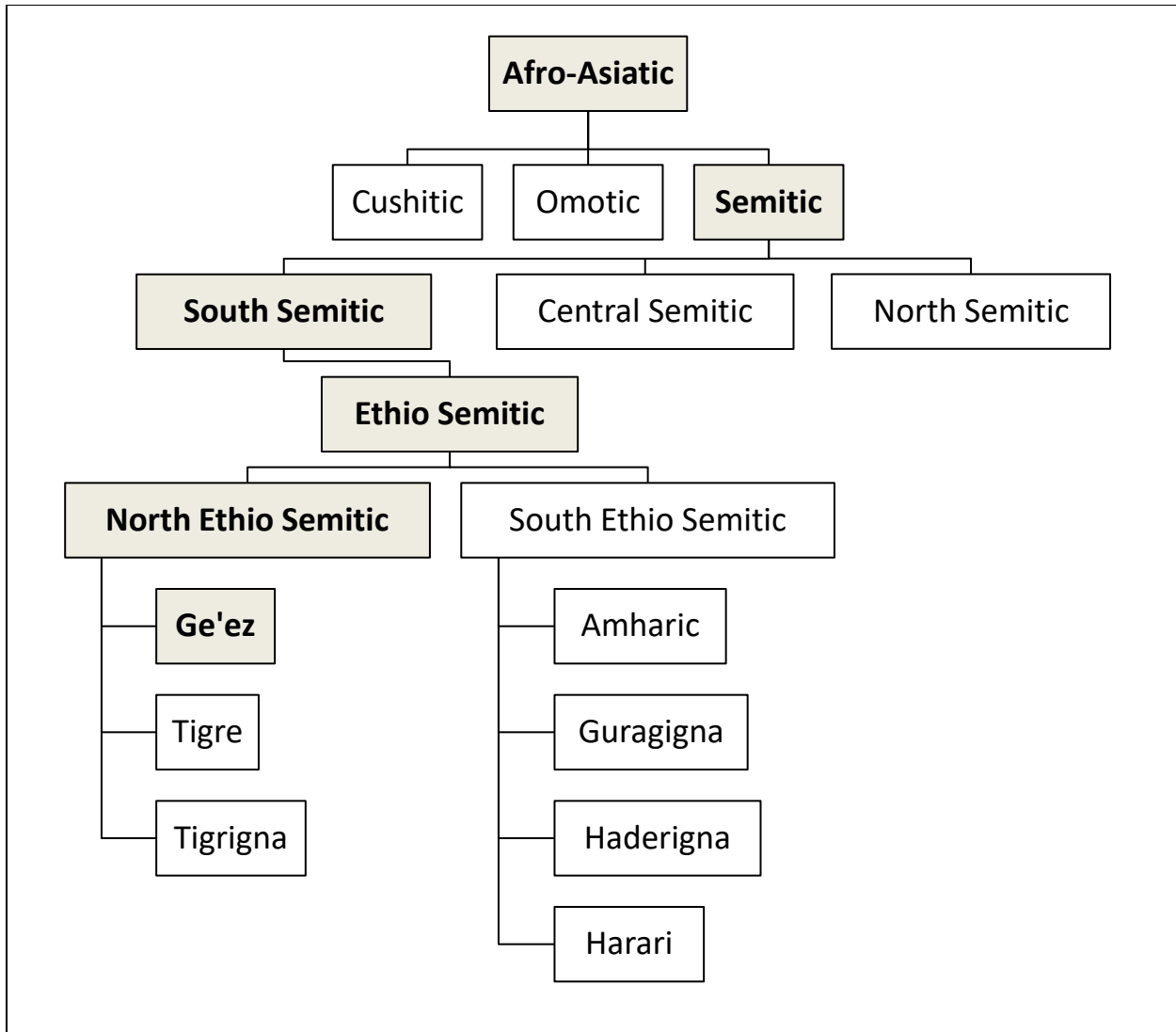


Figure 2. 6: The language family of the Ge'ez language (አብ.ሲ.ሉግሙ, 2012 ዓ.ም)

2.5.2. Symbols of the Ge'ez Writing System (*Fidālat*)

The Ge'ez writing system consists of 26 essential or basic symbols (in Ge'ez known as *Fidālat*) and each of which occurs in a basic form and six other forms. These seven forms of a character are known as *orders*, and the basic forms represent all consonants while the rest with additional strokes and modifications added on to the basic forms indicate a vowel sound associated with it. In other words, the other six forms require only aural adjustments of the basic consonant sound. That is why the Ge'ez writing system is often categorized under Alphasyllabry rather than Alphabetic. The 26 essential or basic

symbols in 7 different forms rises the total core syllables to 182 (26x7) as shown in the table 2.1 below.

Table 2. 1: The core syllables in the Ge'ez writing system (Scelta, 2001)

	<i>Ge'ez</i> ä	<i>Ka'eb</i> u	<i>Salis</i> i	<i>Rab'e</i> a	<i>Hamis</i> é	<i>Sadis</i> i	<i>Sab'e</i> o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
h	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
r	ረ	ሩ	ሪ	ራ	ራ	ር	ሮ
s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
q	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
t	ተ	ቱ	ቲ	ታ	ቲ	ት	ቶ
h	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
n	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
a	አ	ሁ	ሊ	አ	ሌ	እ	ኦ
k	ከ	ከ	ከ	ካ	ኬ	ክ	ኮ
w	ወ	ወ	ወ	ወ	ወ	ወ	ወ
a	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ	ዐ
z	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ
y	የ	የ	የ	የ	የ	የ	የ
d	ደ	ደ	ደ	ደ	ደ	ደ	ደ
g	ገ	ገ	ገ	ገ	ገ	ገ	ገ
t	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ
p	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ
ts	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ
ts	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ
f	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ

ፆ	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
---	---	---	---	---	---	---	---

In Amharic language, however, there are additional eight basic symbols (namely: ሸ, ሹ, ሺ, ሻ, ሼ, ሽ, ሾ, and ሿ) which represent sounds that do not exist in the Ge'ez writing system. It can be observed that these symbols are derived from the Ge'ez script (as ሸ from ሰ; ሹ from ሱ; ሺ from ሰ; ሻ from ሱ; ሼ from ሰ; ሽ from ሱ; ሾ from ሰ; ሿ from ሱ; and ሽ from ሰ). This increases the total basic symbols used in Amharic writing system to 34.

The Ge'ez writing system is also comprised of labialized symbols (Labiovelars). These are represented in four basic forms, namely: ከጐ/kwä/, ከግ/gwä/, ከቄ/qwä/, and ከኀ/hwä/. Unlike to the core symbols which have seven orders, as described above, the labialized symbols have only five orders.

Table 2. 2: Labialized symbols (Labiovelars) in the Ge'ez writing system (አቤሲሎጽግ, 2012 ዓ.ም)

	ä	i	ə	a	e
kw	ከጐ /kwä	ከሩ /kwi	ከጐ /kwə	ከጐ /kwa	ከጐ /kwe
gw	ከግጐ /gwä	ከግሩ /gwi	ከግጐ /gwə	ከግጐ /gwa	ከግጐ /gwe
qw	ከቄጐ /qwä	ከቄሩ /qwi	ከቄጐ /qwə	ከቄጐ /qwa	ከቄጐ /qwe
hw	ከኀጐ /hwä	ከኀሩ /hwi	ከኀጐ /hwə	ከኀጐ /hwa	ከኀጐ /hwe

The Ge'ez writing system is also comprised of punctuation marks and numerals. The punctuation marks consist of a basic wordspace (፣), a sentence-divider or full stop (፥), and other marks like equivalent to the English comma (፣), semi-colon (፥), and adopted symbols like question mark (?), exclamation mark (!), quotation mark (“”), and parenthesis ().

Table 2. 3: Punctuation marks in the Ge'ez writing system (ተግባሩ, 2008 ዓ.ም)

Punctuation marks	Ge'ez	Amharic	English
:	ንዑስ ነጥብ	ሁለት ነጥብ	Ethiopic Wordspace
::	ዐቢይ ነጥብ	አራት ነጥብ	Ethiopic Fullstop
፡	ነጠላ ሠረዝ	ነጠላ ሰረዝ	Ethiopic Comma
፤	ዐቢይ ሠረዝ	ድርብ ሰረዝ	Ethiopic Semicolon

The numeration system consists of digits for 1 to 9, for multiples of 10 (10, 20, 30, 40, 50, 60, 70, 80, and 90), for 100 and 10, 000. There are no zero, decimals and negative numbers in Ge'ez numeration system. The numerals are shown in the table 2.4 below.

Table 2. 4: Numerals in Ge'ez writing system (አቤሴሎም, 2012 ዓ.ም)

Numerals	In Ge'ez words	In Amharic words (English transcription)
፩ - 1	አሐዱ	አንድ (and)
፪ - 2	ክልሌቱ	ሁለት (hulät)
፫ - 3	ሠለስቱ	ሶስት (sost)
፬ - 4	አርባዕቱ	አራት (arat)
፭ - 5	ሓምስቱ	አምስት (amïst)
፮ - 6	ስድስቱ	ስድስት (sïdïst)
፯ - 7	ሰብዓቱ	ሰባት (säbat)
፰ - 8	ሰመንቱ	ስምንት (simïnt)
፱ - 9	ተስሃቱ	ዘጠኝ (zät'äñ)
፲ - 10	ዓሠርቱ	አስር (asir)
፳ - 20	ዕሥራ	ሃያ (haya)
፴ - 30	ሠላሳ	ሰላሳ (sälasa)
፵ - 40	አርባዕ	አርባ (arba)
፶ - 50	ሓምሳ	ኃምሳ (hamsa)
፷ - 60	ስድሳ	ስልሳ (sälsa)
፸ - 70	ሰብዓ	ሰባ (säba)
፹ - 80	ሰማንያ	ሰማንያ (sämanya)
፺ - 90	ተስካ	ዘጠና (zät'äna)

፫ - 100	ምእት	መቶ (mäto)
፪ - 10, 000	እልፍ	እስር ሺህ (asir ših)

In general, Ge'ez script is comprised of 230 symbols representing syllables, labialized, punctuation marks, and numerals. The composition and total number of the symbols (*Fidālat*) in Ge'ez writing system are summarized in the table 2.5 below.

Table 2. 5: Composition and total number of symbols in the Ge'ez writing system (ተግባሩ, 2008 ዓ.ም)

No.	Type of symbols	No. of symbols
1	Core symbols (26 x 7)	182
2	Labialized symbols (4 x 5)	20
3	Punctuation marks	8
4	Numerals	20
Total		230

2.5.3. Order Formation in the Ge'ez Writing System

As described in Section 2.5.2., the Ge'ez writing system has seven orders. The basic forms (i.e. the 1st order) is for representation of consonants, and the rest six orders are created with additional strokes and modifications added on to the basic forms indicate a vowel sound associated with it. In other words, following the 1st order, all the rest six orders are expressed by adding small appendages to the right or left, at the top or at the bottom, by shortening or lengthening one of its main strokes, and by other modification to the 1st order. Except the 6th and the 7th orders, the formation of the other four orders (i.e. 2nd, 3rd, 4th and 5th) follow a pattern with very few exceptions.

The 2nd order is constructed by adding a horizontal stroke at the middle of the right side of the basic form (e.g. ሁ from ሀ; ለ from ለ; and መ from መ). Similarly, the 3rd order is formed by adding the horizontal stroke at the bottom of the right leg of the base character (e.g. ለ from ለ; ጢ from ጢ; and ከ from ከ). The 4th order is formed by adding a diagonal stroke at the bottom of the leg of a one-leg base character or by elongating the right leg

of a two- or a three-leg base character (e.g. ቸ from ቸ; ዳ from ዳ; and ጣ from ጣ). The 5th order is constructed from the base by adding a ring at the right bottom of the right leg (e.g. ኢ from ኢ; ኔ from ኔ; and ሐ from ሐ) (Worku, 1997).

While the 2nd, 3rd, 4th, and 5th orders indicated above are formed according to patterns of great regularity, others, the 6th and 7th, are highly irregular (Bender, Cooper, & Ferguson, 1972). The 6th order is constructed by adding a stroke, loop or other forms in either side of the basic form. Consider as an example syllables ህ from ህ; ል from ል; ሞ from ሞ; ስ from ስ; ር from ር; and ግ from ግ. In the same way, the 7th order is formed from the basic form by elongating the left leg or adding a loop at the top or right side. For instance, syllables ሆ from ሆ; ሎ from ሎ; ቆ from ቆ; ሰ from ሰ; ሞ from ሞ; and ም from ም.

In the development of OCR system for Ge'ez script, a thorough understanding of the order formation and the techniques available for leveraging it is very important. In recent time, a research work by Birhanu *et al.*, (2019a), attempted to exploit the order of the syllables (*Fidālat*) by employing multi-task learning factored CNN for the development of Amharic OCR system. For the detail on the approach followed, refer to (Birhanu *et al.*, 2019a).

2.5.4. Challenges of OCR for the Ge'ez Writing System

Optical character recognition for document images written in ancient languages, such as Ge'ez, is very challenging task. To develop a successful OCR system for Ge'ez script, we need to address some of the challenges related with the script. These challenges are: (a) large number of symbols in the script, (b) high inter-class similarity, (c) lack of knowledge of the writing convention, and (d) class imbalance problem.

A. Large number of symbols in the script

The first challenge relates with the class size, i.e. the large number of symbols in the script. The total number of symbols (the syllables, numerals, and punctuation marks) in Ge'ez script is 230 (two hundred and thirty) without including tonal marks. Existence of such a large number of symbols in the writing system is very challenging for the development of a successful and intelligent OCR system. We need to consider memory

and computational requirements for such a large number of symbols so as to come up with computationally efficient recognizer.

B. High inter-class similarity

The second challenge relates with the shape of the symbols, i.e. high inter-class similarity. In other words, variability between classes is insignificant. From visual observation of the script, it can be noted that the script has relatively small visual differences that correspond to different syllables. Most symbols are very similar in shape. The inter-class variability is even becomes very minimal in the case of handwritings. This is because of the minimal modification and strokes performed on the symbols. In the 1st order (Ge'ez), for instance, ለ and ሰ; ደ and ደ; ዐ and ፀ; ጸ and ጸ; ኀ and ነ; ፈ and ረ has high similarity as one differs from the other in a single stroke. The inter-class similarity can also be observed in the different orders of the writing system. For instance, ሰ and ሰ; ረ and ረ; ጸ and ጸ; ዐ and ዐ; ፈ and ዐ; ፈ and ፈ; ጸ and ጸ, etc. Such level of high similarity in shape poses a great challenge for automatic recognition of handwritten texts. It is challenging even for human beings to identify them without contextual information. The high inter-class shape similarity problem can also be observed in the numerals and punctuations as shown in the table below.

Table 2. 6: High inter-class similarity of the Ge'ez script

<i>Ge'ez</i> 1 st Order	<i>Ka'eb</i> 2 nd Order	<i>Salis</i> 3 rd Order	<i>Rab'e</i> 4 th Order	<i>Hamis</i> 5 th Order	<i>Sadis</i> 6 th Order	<i>Sab'e</i> 7 th Order
ለ and ሰ	ሉ and ሰ	ሊ and ሰ	ላ and ሰ	ሌ and ሰ	ል and ሰ	-
ደ and ደ	ደ and ደ	ደ and ደ	ደ and ደ	ደ and ደ	ደ and ደ	ደ and ደ
ዐ and ፀ	ዐ and ፀ	ዐ and ፀ	ዐ and ፀ	ዐ and ፀ	ዐ and ፀ	ዐ and ፀ
ጸ and ጸ	ጸ and ጸ	ጸ and ጸ	ጸ and ጸ	ጸ and ጸ	ጸ and ጸ	ጸ and ጸ
ኀ and ነ	ኀ and ነ	ኀ and ነ	ኀ and ነ	ኀ and ነ	ኀ and ነ	ኀ and ነ
ፈ and ረ	ፈ and ረ	ፈ and ረ	ፈ and ረ	ፈ and ረ	ፈ and ረ	ፈ and ረ
Numerals						
፩, ፪ and ፫		፬ and ፭	፮ and ፯	፰, ፱ and ፳	፴ and ፵	፶ and ፷
Punctuations						
፥ and ፇ		ፈ and ፈ	ፈ and ፈ	ፈ and ፈ	ፈ and ፈ	

C. Lack of knowledge of the writing convention

The third challenge relates with the writing convention, i.e. lack of sound knowledge on the writing convention of the Ge'ez script. There are some syllables, locally known as “*Mogshe Fidālat*”, which have similar order and phonetic sound associated with them. For instance, ሀ, ሐ and ኀ; ሠ and ሰ; አ and ዐ; ጸ and ፀ; etc. Many scholars believed that these symbols were representing different phonetic sound in the beginning but gradually lost their difference (ተግባሩ, 2008 ዓ.ም፣ አቤሱሎም, 2012 ዓ.ም). We need to give great attention when we use these syllables in writing of a word's sound. In essence, we have to correctly write the word as per the writing convention of the language. Otherwise, it will definitely lead us to wrong meaning and interpretation of the word. The following table depicted this in detail.

Table 2. 7: The writing convention of words in Ge'ez writing system (ተግባሩ, 2008 ዓ.ም)

Syllables	Words in Ge'ez	Meaning in Amharic
ሀ and ሐ	መሀረ	አስተማረ
	መሐረ	ይቅር አለ፣ተወ
ሐ and ኀ	ሐረሰ	አረሰ፣ገመሰ ለእርሻ
	ኀረሰ	አረሰ፣ተንከባከበ ለወለደች ሴት
ሠ and ሰ	ሠረቀ	በራ፣ወጣ ለፀሐይ ብርሃን
	ሰረቀ	ሰረቀ፣አሾለክ ለሌባ
ሠዐ and ሰአ	ሠዐለ	ሣለ፣ስህልን ሣለ
	ሰአለ	ለመነ፣ፈለገ
ጸ and ፀ	ተጸንሰ	ተቸገረ፣አጣ፣ደኸየ
	ተፀንሰ	ተቋጠረ፣ተፀነሰ፣ተረገዘ
ሥ and ስ	ሥን	ወብት፣ማማር
	ስን	ጥርስ
ሥ and ስ	ሥነ	ጽሑፍ፣ድርሰት
	ስነ	ነጌ፣የዝሆን ጥርስ

A writer who lacks sound knowledge of these syllables may use them interchangeably in handwriting. This poses a great challenge for OCR system as the problem inherently pass to the recognizer. In order to overcome such problems, the application of context information in the recognition process as a post-correction task is indispensable.

D. Class imbalance Problem

The fourth challenge relates with frequency, i.e. the naturally occurring frequency of each symbols (the syllables, numerals, and punctuation marks) in real-life actual documents. In a multi-class pattern recognition task, such as OCR, class imbalance occurs when one class contains significantly fewer samples than the rest classes. In a research work by Siranesh (2016), for instance, among from the 26 basic syllables **ጸ** and **ጥ** were found to be rarely occurred in the actual historical Ge'ez manuscripts. In the development of OCR system, therefore, a thorough understanding of the class imbalance problem and the techniques available for addressing it is essential.

2.5.5. Historical Ge'ez Manuscript Collections

According to Bausi *et al.*, (2015) Ethiopia is historically a land of written civilizations since the beginning of the 1st millennium BCE (i.e. much earlier than the date of the earliest surviving manuscripts) and the areas, nowadays, corresponding to the highlands of Eritrea and Northern Ethiopia witnessed the early introduction of the parchment roll and codex, the latter having been strongly fostered in the 4th century. In addition to this, Taddesse stated that the coming of nine saints from Syria in the end of the 5th century was important period for the Ethiopian literature to be flourished because the time was the turning point of writing and translating religious books via Ge'ez language (Taddesse, 1972).

The common writing surface of ancient Ge'ez is "*birana*", a parchment made from animal skin. Recent archaeological evidence suggests that production of parchment in Ethiopia dates back to the pre-Aksumite period in the 1st millennium BCE (Bausi et al., 2015). Because of its organic nature it is subject to degradation over long periods of time. Hence it was a fairly common practice to transfer aging text onto new *birana* in order to preserve the written text (Scelta, 2001). Plants and minerals were used for the preparations of dyes

and pigments of inks. In addition to the organic ink, using iron-gall ink and soot ink were also confirmed recently. The writing instrument used is mainly pens made out of reeds, such as *maqā*, *shambeqo* and *qastanča* (Bausi *et al.*, 2015).

Written manuscripts available today, generally, can be classified as: *translated*, *adopted* and *indigenous works*. The manuscripts have a wide content range from religious to secular nature, such as medicine (e.g. *metsafe madahnit*), history (e.g. *kibre nagast*), astronomy (e.g. *metsafe Henok*), philosophy (e.g. *hateta zezere'a yaqob*), agriculture (e.g. *metsafe gerahit*), and etc (አዲሱ, 2012 ጥ.ፆ). In terms of book forms, According to Bausi *et al.*, (2015) Ge'ez manuscripts are also classified into three physical formats: the miscellaneous forms, the roll (scroll), and the codex.

There are only approximate estimates of the total number of historical manuscripts in Ethiopia and Eritrea. In the early time, based on the assumption that the minimum number of manuscripts necessary for every church for religious services amounts to a few dozen, Sergew roughly estimated of 200,000 surviving manuscripts in codex form (Sergew, 1982). However, there are some disputes surrounding this estimation. Bausi *et al.*, (2015) had made analysis, for instance, given the number of present-day parishes ranging from at least 13,000 to 32,350; the larger average number of manuscripts preserved in the libraries surveyed in the past years; and the persistent use of older as well as new manuscripts along with printed books, they concluded that Sergew's calculation seems probably underestimated. In fact, many scholars believed that several churches and monasteries in Ethiopia and Eritrea are in possession of at least several manuscripts necessary for liturgical activities. In more recent time, another prominent scholar Amsalu estimated of 1,500,000 extant historical Ge'ez manuscripts (Amsalu, 2015; አዲሱ, 2012 ጥ.ፆ).

Bausi *et al.* further stated that monastic libraries also have not yet been systematically explored: the figures of approximately 200 manuscripts for *Dabra Ḥayq Estifānos*, around 570 manuscripts for *Dabra Bizan*, formerly approximately 800 and now approximately 220 manuscripts for *Gunda Gundē*, around 1,000 in the Patriarchate and several hundred at least for the churches of *Dabra Mārḳos*, *Čalaqot*, or the cathedral church of *Aksum Şeyon* may provide some hints (Bausi *et al.*, 2015).

Well-known four modern libraries for historical Ge'ez manuscripts are found in Addis Ababa, namely: Library of the Institute of Ethiopian Studies (IES), National Archives and Library of Ethiopia (NALA), library of the EOTC Patriarchate, and Authority for Research and Conservation of Cultural Heritage (ARCCH). They have rich manuscript collections, approximately 1,500 manuscripts in IES; 859 manuscripts in NALA; 231 manuscripts in the library of the EOTC Patriarchate; and 200 manuscripts in ARCCH⁷. Twelve (12) treasures from NALA are registered as *Memory of the World* by UNESCO⁸ to protect precious manuscripts written in Ge'ez and Amharic script. The vast majority of historical Ge'ez manuscripts that have been investigated and published so far are found outside Ethiopia and Eritrea. The Vatican Library, which was the first collection to be catalogued in printed form, has 1,082 manuscripts, at the least, plus the largest collection of Ethiopian scrolls in the world. *The Bibliothèque nationale de France* has over 1,000 manuscripts, including scrolls. The British Library has at least 624 manuscripts. The Berlin State Library preserves 328 manuscripts plus an important microfilm collection of 182 items from the *Lake Ṭānā* monasteries. Other European and North American institutions also hold important collections of Ethiopic manuscripts (Manchester, Oxford, Frankfurt, Munich, St Petersburg, Moscow, Uppsala, Oslo, Florence, Milan, Parma, Rome (besides the Vatican), Athens, Princeton, Baltimore, etc.). Very important are also the collections hosted in Jerusalem, with probably more than 800 manuscripts (569 preserved in the Ethiopian Archbishopric of Jerusalem, 162 in the monasteries of *Dabra Gannat* and 33 in that of *Dayr al-Sulṭān*)(Bausi *et al.*, 2015).

2.5.6. Digitization of Historical Ge'ez Manuscripts

A number of microfilming and digitization campaigns over the last forty years have made the content of considerable amount preserved manuscripts available to scholars. As far as microfilms are concerned, the collection of the Ethiopian Manuscript Microfilm Library (EMML)⁹, with 9,238 manuscripts, is the most important one. The EMML collection is hosted by the Hill Museum and Manuscript Library (HMML), Saint John's University,

⁷ <http://www.menestrel.fr/?-Ethiopie-822->

⁸ <http://www.unesco.org/new/en/communication-and-information/memory-of-the-world/register/full-list-of-registered-heritage/registered-heritage-page-8/treasures-from-national-archives-and-library-organizations/#c188337>

⁹ <https://hmml.org/about/global-operations/ethiopia/>

Collegeville, Minnesota, which has grown in the course of the last four decades into a major centre for the study, recording, digitization, and cataloguing of Ethiopic manuscripts (among others). It has recently digitized several important collections (for example, the monastic library of *Gunda Gundē*).

More digitization efforts have been sponsored by the Arcadia Fund within the framework of the Endangered Archives Programme (EAP¹⁰) of the British Library. *Mazgaba seelat*, Deeds Project, University of Toronto, stores several thousand images and historical collections of interest to art historians. The Ethiopian Manuscript Imaging Project (EMIP), started in 2005 and has located and digitized scattered smaller collections in the possession of university libraries, dealers and private owners, mostly in North America, but also in England, Israel and Kenya. Quite recently, starting from 2009, the European Research Council-sponsored project Ethio-SPaRe¹¹: Cultural Heritage of Christian Ethiopia: Salvation, Preservation, Research, University of Hamburg, has acquired high quality digital images of more than 2,000 Ethiopic manuscripts from the area of particular historical importance of eastern *Tegrāy*, in northern Ethiopian highlands (Nosnitsin, 2013).

2.5.7. Challenges of OCR for Historical Ge'ez Manuscripts

Optical character recognition on historical Ge'ez manuscripts is a challenging task mainly due to the existence of artefacts/degradations and complexity of the layout. To develop a successful OCR system for historical Ge'ez manuscripts, we need to address some of the common challenges associated with the manuscripts. These challenges are: (a) artefacts or degradations, (b) drawings and ornaments, (c) marginal notes, and (d) physical format and layout variations.

A. Artefacts or degradations

The writing material of historical Ge'ez manuscripts has always been parchment made from animal skin. There are no hints that any other material was ever used for the manuscript production (paper was introduced only in the 19th century, and used mostly for the needs of Europeans) (Bausi *et al.*, 2015). Because of its organic nature it is subject

¹⁰ <https://eap.bl.uk/project/EAP286>

¹¹ <https://cordis.europa.eu/project/id/240720>

to degradation over long periods of time. The major common degradation challenges pertaining to historical Ge'ez manuscripts for OCR system includes ink bleed-through, faded ink, show-through, water blobs and deteriorated parchments. In addition to this, some other artefacts associated with the digitization process of the manuscripts (such as illumination, blur, etc.) may also occurred and poses great challenges. For instance, during optical imaging either indoor or in the field, factors such as noisy environment, sunlight, illumination and occlusion may cause contrast variation, which is non-linear and very expressive. These artefacts/degradations, either in isolation or combination, pose great challenges for OCR system. In the development of successful OCR system, knowledge of the artefacts/degradations helps in to choice the right technique and algorithm to deal with it.

Nowadays, it is customary by the research community of image analysis to deal with the artefacts/degradations during binarization process.

B. Drawings and ornaments



Figure 2. 7: Degradations and decoration in the Four Gospels of Endä Abbä Garimä, 4th - 6th century manuscript (courtesy to Bausi *et al.* 2015, photograph by EBW)

Among the problems that need to be addressed in historical Ge'ez manuscripts is the often complex layout containing drawings and ornaments. For instance, in 'The four Gospels' which is the 14th Century Ge'ez manuscript registered as *Memory of the World* by UNESCO; the first 45 pages of the book are filled with painted pictures (drawings);

particularly of Jesus Christ, his followers, Saints, Angels etc. The book is partially damaged due to mishandling. In addition to this, it can be observed that interlaced border decoration is also ubiquitous in historical Ge'ez manuscripts. This complex layout poses a great challenge for OCR systems. In the development of a successful OCR system, drawings and ornaments need to be treated as graphic region. Therefore, the segmentation of text and non-text regions of a document page needs to be done completely automatically to a high degree of accuracy.



ዐርባዕቱ ወንጌል (በ14ኛው ክፍለ ዘመን የተፃፈ.)
The four Gospels (Bible, New Testament) (14th Century)

Figure 2. 8: Drawing and ornaments in the 14th Century historical Ge'ez manuscript (courtesy to FDRE, Ministry of Tourism and Culture)

C. Marginal notes

Marginal notes at the margin (outer or inner) of pages in manuscripts are also referred to as *marginalia*. They are often set in different fonts and sizes. According to Bausi, it is common in historical Ge'ez manuscripts to preserve notes regarding the institution (usually a monastery or a church), the place or the region where the codex was kept.

Such notes may be inserted in empty spaces or on blank leaves and/or copied onto separate leaves or quires that were then later bound into the codex (Bausi *et al.*, 2015). Some of the manuscripts are heavily annotated and has no relation with the main content. Daniel Kibret has made a detail analysis on marginal notes in historical Ge'ez manuscripts and their implication from document authentication & registration point of view (ዳንኤል ክብረት, 2008 ዓ.ም). These marginalia poses a great challenge for the proper function of OCR engine. Therefore, in the development of OCR system, such handwritten annotations need to be treated as graphic region.

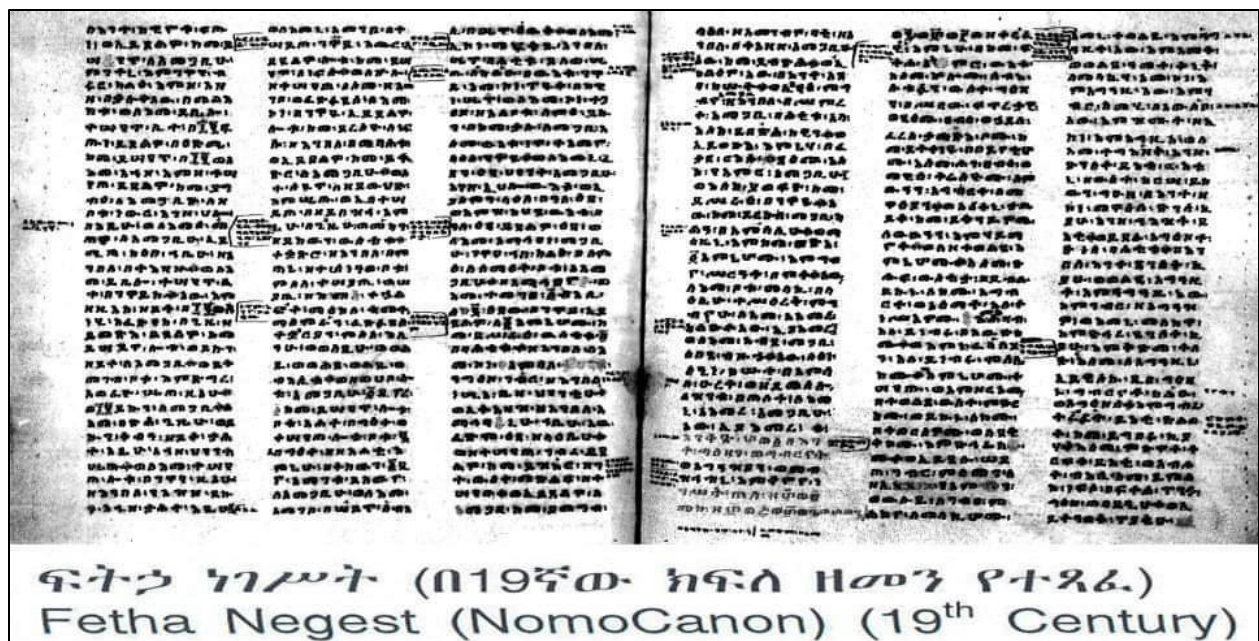


Figure 2. 9: Marginal notes in the 19th Century historical Ge'ez manuscript (courtesy to FDRE, Ministry of Tourism and Culture)

D. Physical format and layout variations

As described above, the physical format of historical Ge'ez manuscripts can be a scroll, codex, and miscellaneous forms. But, the overwhelming majority of Ge'ez texts were transmitted in the codices. The size of the codex varies greatly, depending on the time of preparation and the given text: from "pocket-size" books to volumes more than 45 cm in height, which is so heavy that a grown man could hardly carry them as stated by Nonsnitsin (2012). Nonsnitsin further stated that there are different ways of formatting the written space — and, to a certain extent, the dimensions of the codex — are prescribed for different texts: e.g., a Synaxarion is always written in three columns; the Psalms are

always arranged in one column, each verse beginning on a new line (stichometrically); most of the liturgical books are arranged in two columns; prayer books are written in one column, etc. For some books, more than one type of layout may be used within one manuscript (e.g., Four Gospels). These physical format and layout variations pose great challenges for OCR system. In the development of OCR system, we need to consider these variations. Hence the system needs to be robust and reliable by handling the variations.

2.6. Related Works

Handwriting recognition in historical documents as a research activity of OCR application has grown strongly in the past, primarily when PhD research students in Europe directed their focus towards recognition of medieval time historical documents (Fischer, 2012; Memon *et al.*, 2020). Nowadays, handwriting recognition in historical documents is matured enough and significant number of research groups, journals, conferences, symposiums & workshops are held and dedicated for the advancement of it. In light of the techniques used, different approaches and techniques either in isolation or in combination (e.g., machine learning techniques with image processing techniques) have been investigating to enhance the performance of the recognition in historical documents. Very recently, researchers have focused on developing OCR system for historical documents primarily based on deep learning techniques. The paradigm shift towards deep learning is inevitable, mainly due to the breakthroughs that have been gained using deep learning techniques in the field of artificial intelligence. For a recent survey on handwritten optical character recognition, refer to (Memon *et al.*, 2020).

The main objective of this section, however, is to present previous research works on OCR for Ge'ez and Amharic scripts. In the subsequent section, the beginning of the idea of harnessing OCR technology to Ge'ez and Amharic scripts, its historical developments, declination and the rise of it are discussed. Following it, list of research works on OCR for Ge'ez and Amharic scripts with a detail discussion of approach followed, pros and cons is presented. Finally, the current challenges and future directions of OCR for the scripts are provided. From in-depth exploration of the literature, in fact, it is well observed that historical Ge'ez manuscripts and the Ge'ez language are underrepresented in the

research areas of *document image analysis (DIA)* and *natural language processing (NLP)* respectively. Amharic language, however, relatively has significant number of research works in terms of OCR.

2.6.1. Historical Developments of OCR for Ge'ez and Amharic Scripts

The early attempted of applying OCR techniques to the Amharic language was recorded in 1997 by Worku (1997) in Addis Ababa University at the then School of Information Studies for Africa (SISA). Following it, researches on the area of OCR to Amharic script had continued primarily at SISA in order to exploit the potential of OCR system. The various studies were a Master's thesis work to develop a reliable system to handle different types of printed & type-written Amharic documents (Ermias, 1998; Dereje, 1999; Berhanu 1999; Million, 2000; Yaregal, 2002). Besides that, there were attempts of handwritten Amharic text recognition to specific document types, such as for reading bank Checks (Nigussie, 2000) & text postal addresses (Mesay, 2003), and a recognition for special handwritten Amharic text, locally known as '*Yekum Tsifet*' (Wondwossen, 2004). The studies were primarily focused on adopting algorithms for thinning/skeletonization, underline removal, enhancement, segmentation, normalization, feature extraction and recognition of a wide variety of font type and size.

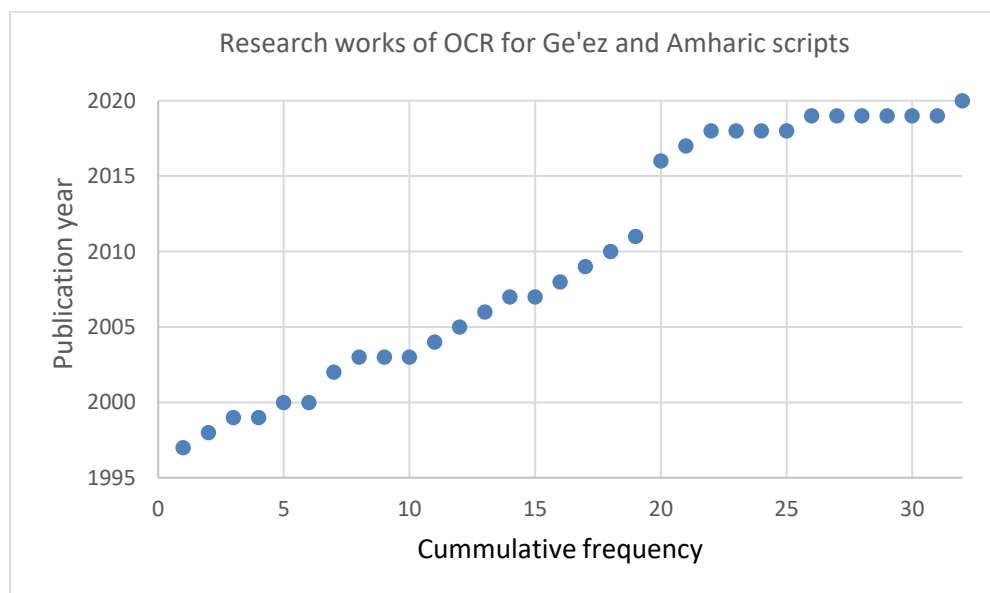


Figure 2.10: Research works of OCR for Ge'ez and Amharic scripts

The above figure 2.10 shows the developments of research works of OCR for Ge'ez and Amharic scripts as measured by the cumulative frequency of its publication. From its inception in 1997 until 2011, there was at least one research work of OCR for Ge'ez or Amharic script almost in a yearly manner. The subsequent five years (between 2011 and 2016), however, there was a decline for the reasons not known yet. In 2016 a thesis work by Siranesh (2016) had occurred and adopted deep learning technique. It is considered as the early attempt of adopting deep learning technique to OCR for Ge'ez script. The last three years period is an important period for the research works of OCR for the Ge'ez script to be flourished primarily because of the deep learning techniques. In the early time, mainly data storage and processing speed limited harnessing the full potential of OCR technology. In more recent time, however, these limitations do not affect any longer. Hence it opens the door to incorporate more complex deep learning techniques. In 2019 only, there was around a publication of six research works, and all of them employed deep learning techniques for the pattern recognition task. The following section provides list of the research works on OCR for Ge'ez and Amharic scripts with a detail discussion of the approach followed, pros and cons.

2.6.2. List of Research Works on OCR for Ge'ez and Amharic Scripts

Organizing and preparing this list aims to identify previous research works on OCR for Ge'ez and Amharic scripts. Instead of the traditional literature review method (narrative review), a review protocol was implemented. In this regard, the review protocol enhances the consistency of the list and reduces the researcher's biasness. This is mainly due to the fact that the researcher has to set criteria for the inclusion and exclusion of any study along with a search strategy (Kitchenham *et al.*, 2010).

According to the review protocol, the searches were mainly performed in the institutional repositories of various universities in Ethiopia and standard databases across the world. These databases include IEEE explore, Elsevier and Springer. In addition to that, journals, conferences, symposiums and workshops on optical character recognition were explored. The initial search was based on a set of keywords; this resulted in a collection of materials. Following it, a thorough review of the materials was performed. After an in-depth review of the collected materials, some of them that were not clearly related to the application

area were excluded. The excluded materials were appeared in the search because of keyword match. In addition, materials were also excluded based on redundancy, non-availability of full-text and relevance. In the review, only studies that were published from 17th May 1997 to 30th September 2020 were considered.

Afterwards, Quality Assessment Criteria (QAC) was applied on the remaining collected materials. According to Kitchenham *et al.*, Quality Assessment Criteria (QAC) is based on a principle to make a decision related to the overall quality of the selected set of studies (Kitchenham *et al.*, 2010). Hence then, the following QAC inquiries were formulated and used to assess the quality of the selected studies.

- a) *Does the study relevant to the objective of the list?*
- b) *Does the study describe context of the research problem?*
- c) *Does the study explain the research methodology & approach followed clearly?*
- d) *Does the study explain the data collection procedure used clearly?*
- e) *Does the study explain the data analysis process with proper example?*

Finally, the selected studies were evaluated using the above mentioned QAC in order to determine the credibility of a particular acknowledged study. The quality of the study was measured based on the total score of the above questions. 2 marks was assigned for each question. The study would be selected if the total score was greater than or equal to 5 out of the maximum value 10. Studies below the total score of 5 were not selected and considered. By strictly following these criteria, the list of research works on OCR for Ge'ez and Amharic languages was prepared and presented. Table 2.8 shows list of the research works.

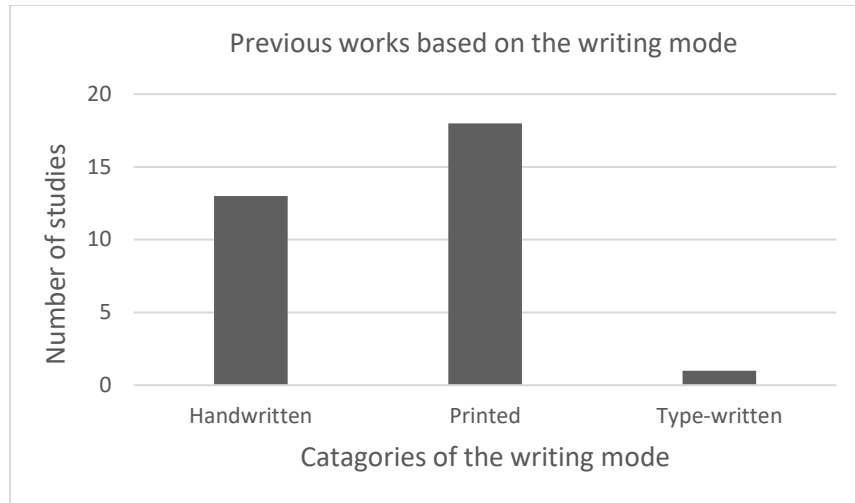


Figure 2.11: Research works of OCR for Ge'ez and Amharic scripts based on the writing mode

Table 2. 8: Research works on OCR for Ge'ez and Amharic scripts [17th May 1997 – 30th Sep. 2020]

Author & year	Writing mode, digitizer, file [Resolution]	Recognition technique [Level]	Approach followed	Pros	Cons	Application area [Dataset]
Worku A. 1997	Printed, flatbed scanner, BMP (Binary image) [300 dpi]	Structural [Character]	<i>Adopted a stage-by-stage segmentation algorithm; topological feature based on the structural shape of thinned image characters; binary tree-structured classification</i>	<i>The early recorded attempt of OCR to Amharic script</i>	<i>Limited handcrafted features were represented, designed for a single font & point type (Washera font style and 12 point type).</i>	Amharic, printed text [Private]
Ermias A. 1998	Printed, flatbed scanner, BMP (Binary image) [300 dpi]	Structural [Character]	<i>Approach followed was similar to & an extension of Worku's work (the above study) but gave emphasis to Italic and removing underline</i>	<i>Adopted thinning and underline detection & removal. It aimed formatted Amharic text</i>	<i>Limited handcrafted features were represented.</i>	Amharic, printed text [Private]
Dereje T. 1999	Typewritten & Typed, flatbed scanner, BMP (Binary image) [300 dpi]	Structural [Character]	<i>It was an extension of Worku's work (the first study) but with some adjustments to make it suite to type-written text</i>	<i>Adopted noise detection and removal; had considered image enhancement</i>	<i>Limited handcrafted features were represented (identified from contour analysis).</i>	Amharic, type-written and printed text [Private]

				and character building		
Berhanu A. 1999	Printed, flatbed scanner, BMP (Binary image) [300 dpi]	Neural Network [Character]	Adopted the stage-by-stage segmentation of Worku's work (the first study), a 16x16 pixels of character image as a feature vector & ANN for classification; trained using BP.	The early recorded attempt of a neural network approach to Amharic OCR.	Very small character classes were considered. The training and test sets were drawn from different distribution of the input-target pairs.	Amharic, printed text [Private]
Million M. 2000	Printed, flatbed scanner, BMP (Binary image) [300 dpi]	Structural [Character]	Adopted and modified the stage-by-stage segmentation of Worku's work, proposed hybrid-thinning; used it for the skeletonization process, topological features based on Worku's procedure were extracted. Then, a feature based binary tree classification was used for the recognition.	The early recorded attempt of Omni-font recognition to Amharic script. It had considered a wide variety of font type & size.	Limited handcrafted features were represented (only primary & secondary global descriptors identified by contour analysis were considered).	Amharic, printed text [Private]
Nigussie T. 2000	Handwritten, flatbed scanner, BMP (Binary image) [300 dpi]	Neural Network [Character]	Adopted the stage-by-stage segmentation of Worku's work (the first study), a 16x16 pixels of character image as a feature vector & ANN for classification; input-hidden-output nodes was designed & trained using BP.	The early recorded attempt of task specific (specific document type) handwritten text recognition to Amharic script	The training and test sets were not drawn from the same distribution of the input-target pairs. It didn't consider actual checks & document image analysis task.	Amharic, handwritten legal amount field on Bank Check [Private]
Yaregal A. 2002	Printed, flatbed scanner, BMP (Binary image) [300 dpi]	Structural/ Syntactical and ANN [Character]	modified the stage-by-stage segmentation of Worku's work (the first study), structural and syntactical features were considered then fed to ANN for classification; input-hidden-output nodes was	The early recorded attempt of structural and syntactical feature extraction to OCR of Amharic script.	Limited handcrafted features (only 19) were considered. Due to that, some characters would likely ended up with a similar binary representation.	Amharic, printed text [Private]

			<i>designed & trained with BP.</i>			
Worku A. and Fuchs, S. 2003	Handwritten, flatbed scanner, BMP (Binary image) [~300 dpi]	Hidden Markov Random Field [Character]	<i>New feature extraction technique (with & without context info) based on the syntactical structure, and pseudo-marginal probability classification were applied.</i>	<i>The early recorded conference paper on OCR to Amharic script (In the 2003 Conference on Computer Vision and Pattern Recognition Workshop-CVPRW'03).</i>	<i>Difficult to be applied to actual Amharic checks which are characterized with different artifacts and irregular writing.</i>	Amharic, handwritten legal amount field on Bank Check [Private]
Cowell J. and Hussain, F. 2003	Printed, flatbed scanner, JPG (Gray-scale) [300 dpi]	Statistical [Character]	<i>Binarized by global threshold, segmented characters by region growing techniques. Then two statistical algorithms applied. Characters were normalized. Two set of templates were used. One compares the character against a series of templates. The other, derived a signature from the character & compares it against a set of signature templates.</i>	<i>The 2nd early recorded conference paper on OCR to Amharic script (In the Proceedings of the 7th Int. Conference on Information Visualization-IV'03). The confusion matrix was examined in detail to gain knowledge of the level of confusion.</i>	<i>The recognition system was highly sensitive to the quality of the input character in both situations. The situation with the signature templates was worse.</i>	Amharic, printed text [Private]
Mesay H. 2003	Handwritten, flatbed scanner, BMP (Binary image) [300 dpi]	Neural Network [Character]	<i>Adopted the stage-by-stage segmentation of Worku's work (the first study), normalized characters; used Local Line Fitting representation, and Least square technique was applied to fit a linear model to the distribution of extracted features; fed to ANN, and trained with BP</i>	<i>The study used 10-fold cross validation technique & a separate testing set, though it was constrained to postal addresses.</i>	<i>The training and test sets were not drawn from real envelopes; using linear regression model for handwriting recognition characterized by non-linear dependency was not a wise decision.</i>	Amharic, handwritten text postal addresses [Private]
Wondwosen M. 2004	Handwritten, flatbed scanner, BMP	Neural Network [Character]	<i>Adopted the stage-by-stage segmentation of Worku's work (the</i>	<i>The recommendations were strong. Hence</i>	<i>The study didn't consider actual</i>	Amharic, traditionally

	(Binary image) [300 dpi]		<i>first study), a 16x16 pixels of character image as a feature vector & ANN for classification. input-hidden-output nodes was designed & trained using BP.</i>	<i>the present study considers all of them carefully.</i>	<i>historical manuscripts. The training & test sets were not drawn from the same distribution of the input-target pairs.</i>	handwritten text [Private]
Million M. and Jawahar C. 2005	Printed, flatbed scanner, (Gray-scale) [300 dpi]	Statistical [Character]	<i>The images were binarized and noise removed (Gaussian filtering), and skew corrected (projection profile). Segmentation followed top-down approach (projection profiles), characters scaled to 20x20 pixels and decomposed to their constituents (connected component analysis). Proposed a two-stage feature extraction scheme; PCA followed by LDA for optimal discriminant feature extraction. A DAG classifier with SVMs was applied and trained pair-wise using the two-stage feature extraction.</i>	<i>It is a conference paper presented on the prestigious ICDAR/2005 conference. The study reported the challenges towards the recognition of Omni-font Amharic scripts and possible solutions. The techniques (PCA & LDA) are powerful for feature dimensionality reduction and data representation learning.</i>	<i>Objective evaluation methods were not considered for binarization, skew correction and page segmentation tasks. No language model or post-processing in the recognition process. Actual real-life documents were not sampled when the training set prepared.</i>	Amharic, Synthesize printed text [Private]
Yaregal A. and Bigun, J. 2006	Printed	Structural/ Syntactical [Character]	<i>Directional field tensor was proposed as a tool for character segmentation and extracting primitive structural features & their spatial relationships. A special tree structure was used to represent the spatial relationship of the primitive structures. For each character, a unique string pattern was</i>	<i>It is a conference paper presented on the prestigious ICPR/2006 conference. Attempted to design a generic recognition system that invariably works for different font</i>	<i>Document image analysis tasks were not considered in the recognition process. No post-processing or post-correction task in the recognition process.</i>	Ethiopic, Printed text [Private]

			<i>generated from the tree and recognition was achieved by matching the string against a stored knowledge base of the alphabet.</i>	<i>types, sizes and styles. The recognition system does not need training.</i>		
Yaregal A. and Bigun, J. 2007a	Printed, flatbed scanner, (Grayscale) [300 dpi]	Structural/Syntactical and ANN [Character]	<i>Based on the previous work on directional field tensor (listed above), proposed ANN classifier that took 1D string patterns as an input generated from the spatial r/ships of primitive structures of characters. During the recognition process, the size of Gaussian window had to be optimized according to font sizes and document types.</i>	<i>Presented on the Int. Workshop on Advances in Pattern Recognition IWAPR/2007. Attempted to design a generic recognition system that invariably works for different font types, sizes and styles.</i>	<i>Document image analysis tasks were not considered in the recognition process. The user has to set the size of the Gaussian window manually as per the font size of the actual document. No post-correction task in the recognition process.</i>	Ethiopic, Printed text [Private]
Yaregal A. and Bigun, J. 2007b	Printed, flatbed scanner, (Grayscale) [300 dpi]	Structural/ANN/Template matching [Character]	<i>This work is based on the previous work on directional field tensor and ANN classifier (listed above). However, in this work, results from a similarity-based pattern matching and ANN were investigated and compared. Template matching was further applied for confusing characters.</i>	<i>It is a conference paper presented on the prestigious ICDAR/2007 conference. Attempted to design a generic recognition system that invariably works for different font types, sizes and styles.</i>	<i>Document image analysis tasks were not considered in the recognition process. The user has to set the size of the Gaussian window manually as per the font size or type of the actual document; No post-correction task in the recognition process.</i>	Ethiopic, Printed text [Private]
Yaregal A. and Bigun, J. 2008	Handwritten, flatbed scanner, (Grayscale) [300 dpi]	Structural/Syntactical [Character]	<i>Proposed structural & syntactic model by using direction field tensor. A special tree structure was used to hold the r/ship and the tree was traversed to generate a set of</i>	<i>Conference paper presented on prestigious ICFHR/2008 conference. Attempted to design a generic recognition</i>	<i>Document image analysis tasks were not considered in the recognition process. The user has to set the size of the Gaussian</i>	Ethiopic, Historical & handwritten character [Private]

			<i>unique sequence of primitive strokes for each character. Then the generated sequence of strokes was matched against a stored knowledge base of primitive strokes for each Ethiopic character.</i>	<i>system that invariably works for different handwriting styles and sizes. The recognition system does not need training.</i>	<i>window manually as per the font size or type of the actual document. No post-correction task in the recognition process.</i>	
Yaregal A. and Bigun, J. 2009	Handwritten, flatbed scanner, (Grayscale) [300 dpi]	Structural/ Syntactical/ HMM [Word]	<i>Proposed Amharic word recognition in unconstrained handwritten text using HMM. Text lines & primitive structural features (primitive strokes and their spatial relationships) were extracted by using direction field tensor. For each character, primitive structural features were stored as feature list.</i>	<i>Presented on the prestigious ICDAR/2009 conference. The early recorded attempt of Word level recognition to Amharic OCR.</i>	<i>Document image analysis tasks were not considered in the recognition process. No language model or post-processing in the recognition process.</i>	Amharic, handwritten text [Private]
Abay T. 2010	Printed, flatbed scanner, JPG (Grayscale) [300 dpi]	Neural Network [Character]	<i>First applied Wiener adaptive filtering method for noise removal then binarization (Otsu's method) in the pre-processing stage. Adopted the stage-by-stage segmentation suggested by Pal and Chaudhuri (1995), normalized characters into 20x20 pixels by using linear interpolation technique (set by experimentation). Hit- and-miss morphological analysis for thinning was employed. ANN with a network 400-0-1, input-hidden-output nodes was designed & trained using BP.</i>	<i>Binarization task (Otsu's global thresholding method) was applied. In the thesis, the body of knowledge of the area was organized & explained in a manner that beginners could see the big picture about OCR.</i>	<i>Feature extraction task was not considered in the system; the thinning would likely made loss of information about the character to be recognized. Due to the thinning, multiple characters would likely ended up with a similar binary representation. Using a single node on the output layer for a multi-class pattern classification problem was also not a good decision.</i>	Amharic, Printed real-life documents [Private]

<p>Fitsum D. 2011</p>	<p>Printed, flatbed scanner, BMP (Binary image) [300 dpi]</p>	<p>Statistical [Character]</p>	<p>The document was binarized, deskewed and enhanced. Then segmentation applied, and then a total of 1500 samples (100 different samples from selected 15 characters, scaled to 30x30) were split into 1200 for training & 300 for testing. A dimension reduction tools (PCA, NPE and MDS) were investigated and fed to the SVM for training.</p>	<p>3 fonts (Nyala, Power Geez & Agafari) & d/t kernel types (linear, polynomial and GRBF) were investigated. NPE, PCA and MDS techniques were also investigated for feature dimensionality reduction.</p>	<p>Objective evaluation methods were not considered for binarization and segmentation tasks. No language model or post-processing in the recognition process.</p>	<p>Amharic, Printed character [Private]</p>
<p>Siranesh G. 2016</p>	<p>Handwritten, digital camera (16MP), JPEG (RGB)</p>	<p>Deep Learning [Character]</p>	<p>RGB to grayscale conversion, Binarization (Hybrid method), skew correction (Bounding box technique) and morphological operation were applied, and Segmented characters using projection profile into 30x30. The network was: 900-[100x3]-24, input-[hidden]-output nodes (Softmax) and trained on RBM, greedy layer-wise unsupervised training strategy.</p>	<p>The early attempt of deep learning technique to OCR of Geez script. The feature extraction process (based on the three hidden layers) was investigated in detail.</p>	<p>For binarization and skew correction tasks, objective evaluation methods were not considered. No regularization techniques were applied to overcome overfitting problem. No language model in the recognition process.</p>	<p>Geez, Historical manuscripts [Private]</p>
<p>Shiferaw T. 2017</p>	<p>Handwritten, Mobile phone scanner app, (RGB) [8MP 16MP]</p>	<p>Statistical [Character]</p>	<p>RGB to grayscale conversion, binarization (Iterative method), noise removal (bi-level noise filtering) applied. Performed stage by stage segmentation for line extraction and bounding box projection for characters. Size normalization</p>	<p>The study considered actual historical Geez manuscripts and investigated them in detail. In the thesis, the body of knowledge of</p>	<p>Skew correction was not considered, and objective evaluation metrics were not considered for binarization & segmentation tasks.</p>	<p>Geez, Historical manuscripts [Private]</p>

			<i>(nearest-neighbor interpolation) and thinning were applied. For feature extraction (Extent, CC analysis & projection profile) applied. RBF kernel function & SVM were used for classification.</i>	<i>the area was organized & explained in a manner that beginners could see the big picture about OCR.</i>	<i>No language model or post-processing in the recognition process.</i>	
Betselot et. al, 2018	Handwritten, flatbed scanner, (Grayscale) [300 dpi]	Statistical [Character]	<i>In the pre-processing stage: smoothing & noise removal (median filtering), cropping and normalization were applied. A combination of Histogram of oriented gradients (HOG), Local Binary Pattern (LBP) and geometrical features were used and then LDA (for dimension reduction) and DCA to integrate two set of features applied and investigated, then multiclass SVM with linear kernel was probed for classification.</i>	<i>Large combinations of handcrafted statistical features (3839 features) were considered (along with geometrical features). The proposed model was trained on extra Chars74K benchmark numeric data set, and the model was validated using 10-fold cross-validation.</i>	<i>The collected data were not sampled from the real-world distribution (just drawn from handwritten characters written in isolation on white papers). Document image analysis and language model were not considered in the recognition process.</i>	Amharic, handwritten character [Private]
Birhanu et. al, 2018	Printed	Deep Learning [Character]	<i>Proposed a model of deep CNN (multiple convolutional, ReLU, Maxpooling, fully connected layers & Softmax output) were used. The dataset was curated using OCRopus (Open source OCR system developed by Breuil et al.). Trained over a dataset of 80,000 Amharic character images, normalized into 32x32 pixels. Trained with SGD.</i>	<i>The study considered CNN (auto-derived feature extractor) for feature extraction task (the early recorded attempt to Amharic OCR). Early stopping and dropout regularization , techniques were used.</i>	<i>Document image analysis tasks were not considered in the recognition process. No language model or post-processing task in the recognition process. The dataset was limited to synthetic character images of Power Geez font only.</i>	Amharic, Synthetic character [Private]

Abeto A. 2018	Printed, flatbed scanner and digital camera of 10MP, JPEG (RGB) [200-1200dpi]	Deep Learning [Character]	<i>Binarization (Otsu's method), skew correction (Hough transform) and segmentation based on projection profile were applied. Anti-aliasing filter was used for size normalization (32x32). CNN was used as a feature extractor. The designed network was: 2 convolution, 2 ReLU & 2 max pooling layers, with two fully connected layer. Trained with SGD. Synthesized character images were added up to the dataset.</i>	<i>The study considered auto-derived feature extractor (CNN) for feature extraction task (the early recorded attempt to Amharic OCR next to the above study). Data augmentation and validation set were applied.</i>	<i>For binarization and skew correction tasks, objective evaluation methods were not considered. Providing synthetic training samples for discrimination method of classification (such as ANN) is not helpful. No language model in the recognition process.</i>	Amharic and English scripts, real-life and synthetic documents [Public ¹²]
Direselign et. al, 2018	Printed	Deep Learning [Text-line]	<i>In the pre-processing stage, text-line normalization by the method of center normalization scaled to the height of 48 pixels followed by 1D LSTM networks, along with CTC. Trained over on a dataset consisted of synthetic text-line images from different sources written in Amharic, Geez and Tigrigna languages. The text-lines were generated using OCRopus. The performance was measured using edit distance.</i>	<i>The early recorded attempt of text-line level recognition to OCR of Ethiopic scripts. Text-line recognition task was formulated as a sequence pattern classification problem. Error analysis was investigated in detail.</i>	<i>Document image analysis tasks were not considered in the recognition process. No post-processing or post-correction task in the recognition process. The training & test sets were not drawn from the same distribution of the input-target pairs.</i>	Ethiopic, Synthetic texts [Private]
Birhanu et. al, 2019a	Printed	Deep Learning [Character]	<i>Proposed Factored CNN (FCNN) based on the rows and columns of the 'Fidel Gebeta', with two Softmax</i>	<i>The study attempted to exploit the placement of the characters in</i>	<i>Document image analysis tasks were not considered in the recognition process.</i>	Amharic, Synthetic characters [Private]

¹² <https://drive.google.com/file/d/1ywg25O6FAFZVixfr1YSIppqEIU4uhOGm/view> [Abeto's Dataset]

			<p>classifiers that shared layers at the lower stage & task specific layer at their last stage. Both classifiers were trained jointly and can detect the row & column location of a character. Trained with Adam optimizer. The author used the same dataset of his previous work (discussed above) but labelled each character image in a row-column order.</p>	<p>the 'Fidel Gebeta' (the early recorded such attempt to Amharic OCR).</p> <p>Multi-task learning was applied to train the two tasks (row and column detectors) jointly, hence reduced the number of multi-classes into 40 classes (33 rows and 7 columns) only.</p>	<p>No language model or post-processing task in the recognition process.</p> <p>The dataset was limited to synthetic character images of Power Geez font only.</p>	
Fetulhak A. 2019	Handwritten, Mobile phone scanner app, (RGB) [332 dpi]	Deep Learning [Character]	<p>Proposed a model of deep CNN (multiple convolutional, ReLU, Maxpooling, fully connected layers & Softmax output) were used. Trained over a dataset of 132, 500 handwritten Amharic character images, normalized into 28x28 pixels. Trained with RMSprop optimizer.</p>	<p>Hyperparameters and optimizers (RMSprop, SGD & Adam optimizer) were investigated in detail.</p> <p>Dropout regularization and data augmentation techniques were used.</p>	<p>The collected data were not sampled from the real-world distribution (drawn from handwritten characters written in isolation on white papers). Page layout analysis & post-correction were not considered in the recognition process</p>	Amharic, handwritten character [Private ¹³]
Mesay et. al, 2019	Handwritten, digitized by Digimemo devices	Deep Learning [Character]	<p>Adopted and modified a deep CNN architecture from Arabic handwritten character recognition by El-Sawy et. al, (2017). Modified it into two convolutional, two Maxpooling,</p>	<p>Multi-task learning (hard parameter sharing) using rows and column classes of the Amharic characters was</p>	<p>Document image analysis and language model (post-processing task) were not considered in the recognition process.</p>	Ethiopic (Amharic & Geez languages), handwritten character [DEHR ¹⁴]

¹³ Interested individual may access the dataset through contact address of the author: afetulhak@yahoo.com

¹⁴ DEHR (Dataset for Ethiopic Handwriting Recognition) can be accessed by contacting the curator via: <http://www.hh.se/staff/josef/>.

			<p>fully connected layer & Softmax output layer.</p> <p>Trained over a dataset consists of 1,192,500 handwritten Ethiopic character images, scaled to 32x32 pixels.</p>	<p>investigated in detail.</p> <p>Early stopping and dropout regularization, and data augmentation techniques were used.</p>	<p>Activation function, such as ReLU function, was not considered in the architecture.</p>	
Fitehalew A. 2019	<p>Handwritten, Digital camera and scanner, (RGB) [300-600dpi]</p>	<p>Deep Learning [Character]</p>	<p>Binarization (Otsu's method), Noise removal (non-local mean denoising), skew correction (Hough method), morphological operation were applied, and Segmented characters using contour analysis</p> <p>Size of 28x28, fed to deep CNN consists of 3 convolutional, ReLU, one Maxpooling, 2 fully connected and Softmax (28 nodes) layers. Trained with Adam optimizer.</p>	<p>The confusion matrix was examined in detail to gain knowledge of the level of confusion.</p> <p>Early stopping and dropout regularization techniques were used to overcome overfitting problem.</p>	<p>Objective evaluation metrics were not considered for binarization, skew correction and page segmentation tasks.</p> <p>No language model or post-processing in the recognition process.</p>	<p>Geez, Historical manuscripts [Private]</p>
Birhanu et. al, 2019b	<p>Printed, flatbed scanner, (Grayscale) [300 dpi]</p>	<p>Deep Learning [Text-line]</p>	<p>Proposed a model consisted of BLSTM (for feature extraction & sequence learning) and CTC for sequence labeling. Trained serially over a dataset curated using OCRopus. The dataset consists of 337,332 Amharic text-line images, scaled to 48x128 pixels. Trained with Adam optimizer.</p>	<p>Text-line recognition problem was formulated as a sequence pattern classification problem.</p> <p>Early stopping technique was applied to overcome overfitting problem.</p>	<p>Objective evaluation metrics for binarization tasks were not considered; post-processing task was not considered in the recognition process. The dataset was limited to two font types only.</p>	<p>Amharic, Printed and synthetic documents [ADOOCR¹⁵]</p>
Halefom et. al, 2019	<p>Handwritten, flatbed scanner, (RGB)</p>	<p>Deep Learning [Character]</p>	<p>Proposed a hybrid of CNN and XGBoost model. The CNN was used</p>	<p>The study compared MLP & XGBoost as</p>	<p>The collected data were not sampled from</p>	<p>Ethiopic, Handwritten Character [Private]</p>

¹⁵ <http://www.dfki.uni-kl.de/~belay/>.

			for feature extraction of 502 handwritten characters, & trained over a dataset consisted of 85,843 character images scaled to 28x28 pixels, then the extracted features were given to the XGBoost; MLP for classification. The network was trained with SGD.	classifier end part of the CNN. Augmentation, early stopping, and dropout techniques were also applied. Error analysis was investigated in detail.	the real-world distribution. Document image analysis and post-correction tasks were not considered in the recognition process	
Birhanu et al, 2020	Printed, flatbed scanner, (Grayscale) [300 dpi]	Deep Learning [Text-line]	Proposed a unified model of deep CNN, BLSTM and CTC for feature extraction, sequence learning & sequence labeling, respectively. Trained end-to-end over the same dataset of his previous work discussed above.	The study attempted end-to-end learning to Amharic OCR and CNN is employed for feature extraction task. Dropout regularization technique was applied.	All the cons in the previous work (mentioned above) are also inherited and associated to this work.	Amharic, Printed and synthetic text-lines [ADOCR]

By referring to Table 2.8, some analysis on the previous research works can be made. Based on the writing mode, the research works so far can be categorized as printed, type-written and handwritten. The writing mode of the majority of the research works is printed (18 studies) and followed by handwritten (13 studies). Among from the 13 studies of the handwritten, only 4 studies focused on the historical Ge'ez manuscripts (depicted in the table with green light color).

Previous studies primarily focused towards character level recognition (28 studies from the total 32 studies). Only one study, i.e. (Yaregal and Bigun, 2009) attempted to model word level recognition using HMM. In contrast, three studies, i.e. (Direselign et al., 2018), (Birhanu et al., 2019b) and (Birhanu et al., 2020) attempted to model text-line level recognition. One of the advantages of text-line level recognition is that it does not require text-line to word and word to character segmentations, which is one of the most common reasons for high word or character error rate.

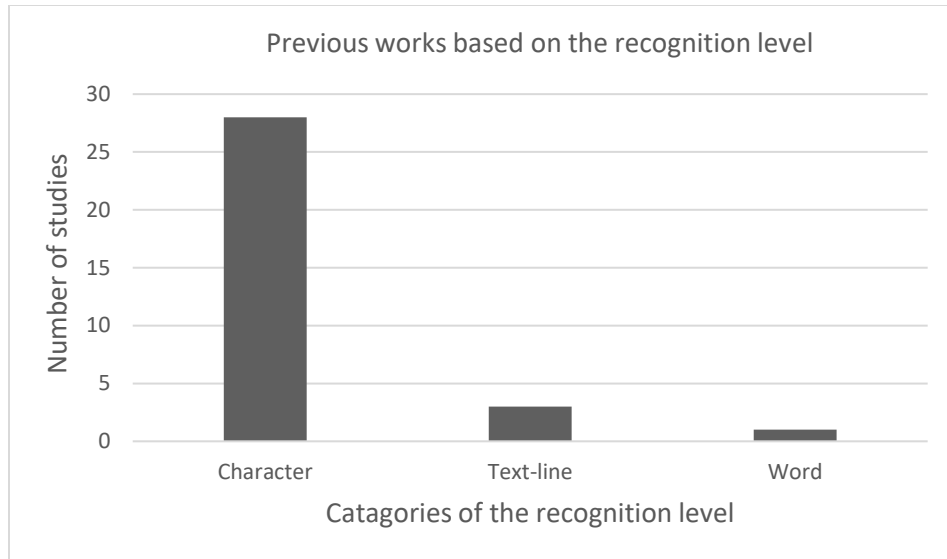


Figure 2.12: Research works of OCR for Ge'ez and Amharic scripts based on the recognition level

Based on the type of dataset utilized, previous studies can be categorized into *handwritten* (i.e. the dataset is drawn from documents of purely modern A4 size paper written with ballpoint pen), *historical* (i.e. the dataset is drawn from actual historical manuscripts), *printed text* (i.e. the dataset is drawn either from newly printed out texts or real-life diversified printed documents such as books, magazines etc.), *synthetic* (i.e. the dataset is drawn from purely synthetic texts but not printed out), and *type-written* (i.e. the dataset is drawn from documents prepared using typewriter device).

As it can be observed from the graph below, from the previous research works, only four studies i.e. (Yaregal and Bigun, 2008), (Siranesh, 2016), (Shiferaw, 2017) and (Fitehalew, 2019) attempted to apply OCR techniques to the historical Ge'ez manuscripts. All of them focused towards character level recognition. In light of the pattern recognition techniques, the studies applied hybrid (structural/syntactical approach), deep Multilayer perceptron (MLP), Support Vector Machines (statistical approach), and deep convolutional neural networks (CNN), respectively.

The first study (Yaregal and Bigun, 2008) applied structural and syntactical pattern recognition approach to OCR of handwriting recognition either scale to modern or historical documents. In this work, directional field tensor was proposed as a tool for character segmentation and extracting primitive structural features and their spatial relationships. A special tree structure was used to represent the spatial relationship of the

primitive structures and traversed to generate a set of unique sequence of primitive strokes for each character. Then the generated sequence of strokes was matched against a stored knowledge base of primitive strokes for each character. The study attempted to design a generic recognition system that invariably works for different handwriting styles and sizes. But, document image analysis tasks were not considered in the recognition process.

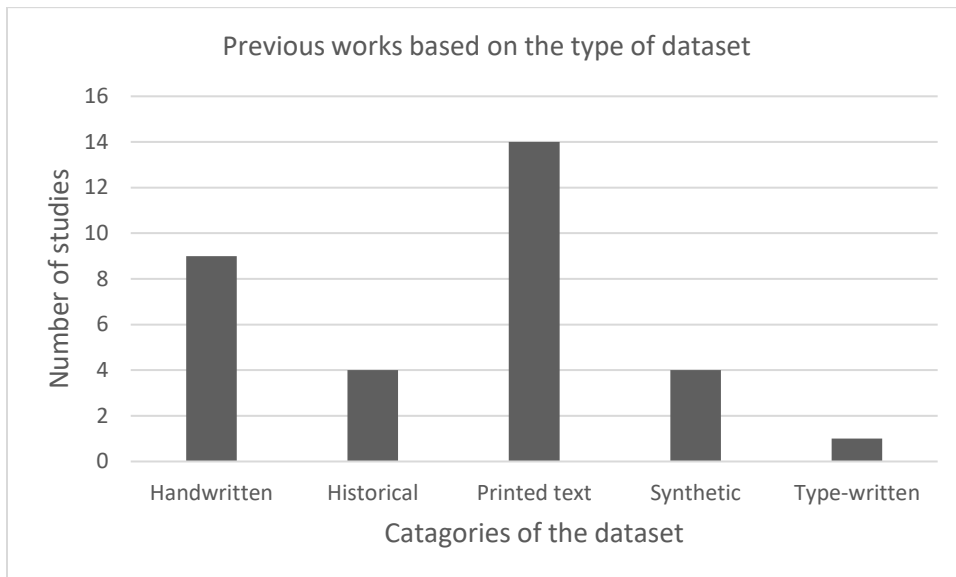


Figure 2.13: Research works of OCR for Ge'ez and Amharic scripts based on the type of dataset

The second study (Siranesh, 2016) applied deep learning techniques to OCR on historical Ge'ez manuscripts. The document page was decomposed stage-by-stage into characters using projection profiles and normalized to 30x30 pixels size. Multilayer perceptron (MLP) network with three hidden layers was employed for feature extraction process and Softmax output layer for the classification task. The network was trained over Restricted Boltzmann Machine (RBM) in a greedy layer-wise unsupervised training manner. Finally the whole network was fine tuned in supervise manner using the Softmax function criteria. But, no regularization techniques were applied to overcome the overfitting problem. The scope of the study was limited to 24 basic syllables (consonants) of the Ge'ez script only.

The third study (Shiferaw, 2017) applied statistical pattern recognition approach to OCR on historical Ge'ez manuscripts. The document page was decomposed stage-by-stage into characters using projection profiles and character's size was normalized using

nearest-neighbor interpolation technique. Following thinning/skeletonization process, extent, connected component analysis and projection profile were applied to extract handcrafted statistical features (produced a total of six features for each character image). Finally one-against-all multiclass classification of SVM with RBF kernel function were employed for the classification task. The scope of the study was limited to 202 syllables of the Ge'ez script.

The fourth study (Fitehalew, 2019) applied deep learning techniques of convolutional neural network topology to OCR on historical Ge'ez manuscripts. The characters were directly segmented using contour analysis method. Following it, extracted characters were then normalized to 28x28 pixels size. Deep convolutional neural network (CNN) was employed for feature extraction process and Softmax output layer (28 nodes) for the classification task. The network was trained with Adam optimizer. Early stopping and dropout regularization techniques were applied to overcome the overfitting problem. The scope of the study was limited to 28 essential syllables of the Ge'ez script only.

2.6.3. Challenges of OCR for Ge'ez and Amharic Scripts in the Era of Deep Learning

Research works on OCR system development for Ge'ez and Amharic languages have sustained to the present time since the date of its inception at SISA in 1997, mainly due to the significant efforts that SISA's postgraduates have made. Some of them have continued playing a vital role in the domain. In more recent time, newly emerging PhD students in Europe and Asia have also made significant conference paper contributions. Despite all these remarkable efforts, there is still a long way to go for a better advancement of OCR for Ge'ez and Amharic scripts in the current era of deep learning. In the past, mainly data storage and processing speed limited harnessing the full potential of OCR technology. Today, these limitations do not affect any longer. Hence it opens the door to incorporate more complex deep learning techniques that can handle large amounts of data in multiple passes, along with that, it allows multiple experimentations at a time.

In the current era of deep learning, there are three (3) key challenges primarily for the development of research works on OCR for Ge'ez and Amharic scripts. The three key

challenges are: (a) lack of benchmarking dataset, (b) lack of computing facility, and (c) lack of awareness.

A. Lack of benchmarking dataset

The first challenge of OCR for Ge'ez and Amharic scripts in the current era of deep learning is lack of benchmarking dataset. The public availability of benchmarking dataset in the form of free download over the internet is very important for research purpose. It supports the development of novel recognition systems and allows for a comparison of different recognizers on the same data. Thus it can attract individuals or research groups to conduct researches on OCR system.

B. Lack of computing facility

The second challenge is lack of computing facility or labs in higher education institutions of Ethiopia. Incorporating deep learning techniques to OCR system, requires cutting age computing devices, such as advanced GPUs, in the development process to train a recognizer over a large dataset. In addition, proprietary of Software also affects the availability of machine learning libraries in order to apply deep learning techniques.

C. Lack of awareness

Despite the high potential of OCR systems (for research purpose, commercial and industrial importance), there is a serious lack of awareness about OCR research in the community. This can be easily observed from the total number of research works published in the recent times. Apart from PhD fellow Ethiopians in abroad, it is rare to get latest developments of OCR research work from higher education institutions of Ethiopia. There are no locally initiated research groups or competitions, workshops, conferences, symposiums, etc. dedicated their time and effort for the advancement of OCR for Ge'ez and Amharic languages.

2.6.4. Future Directions of OCR for Ge'ez and Amharic Scripts

Despite the challenges, research works on OCR system for Ge'ez and Amharic scripts have to be continued to grow in quantity and quality. Bringing deep learning and other emerging techniques to the development of OCR system is a wise decision and inevitable. The focus of current researches on OCR has to be directed towards bridging the gap that

comes mainly from lagging behind the time. When we look at the recent advancements of deep learning, we have no choice but to be involved and committed to study and conduct research in the field. Whatever the nature and the degree of our involvement, we cannot afford not to take the advantages and enrich ourselves as well as others who will benefit from the research output or utilize it for social good.

In the near future, benchmarking dataset has to come into a reality. Following it, a national or international contest of novel recognition systems over the same dataset can be launched and held in a timely manner. Stakeholders or concerned bodies have to work on it, because such activities create awareness among researchers and practitioners in the field. For instance, the newly inaugurated Artificial Intelligence Center (AIC) of Ethiopia can take this initiative and bridge the gap.

Researchers also have to put forth their efforts to contribute to the body of knowledge at international conferences, such as ICDAR, ICFHR and IAPR. In this regard, research works in the domain of document image analysis and handwriting recognition from Ethiopia are underrepresented comparing to other developing nations. From the literature, it is observed that international conferences of ICDAR and its affiliations, such as ICFHR, DIBCO, H-DIBCO, DSECO and IAPR are dedicated for the advancements of document image analysis (including binarization and skew estimation) and pattern recognition. The early recorded conference paper on OCR to Amharic script was presented by Worku and Fuchs (2003) in the proceedings of the 2003 Computer Vision and Pattern Recognition Workshop (CVPRW'03). Almost in the same time, another conference paper on OCR to Amharic script was presented by Cowell and Hussain (2003) in the proceedings of the 7th International Conference on Information Visualization (IV'03). Following it, scholars (Million and Jawahar, 2005), (Yaregal and Bigun, 2006), (Yaregal and Bigun, 2007a), (Yaregal and Bigun, 2007b), (Yaregal and Bigun, 2008), and (Yaregal and Bigun, 2009) were the major contributors to the prestigious conferences, namely ICDAR/2005, ICPR/2006, IWAPR/2007, ICDAR/2007, ICFHR/2008, and ICDAR/2009, respectively. An extensive literature survey reveals no conference paper contributions to international conferences between 2010 and 2017. But, the trend in the last few years seems changing. Betselot *et al.*, (2018), Birhanu *et al.*, (2018), Direselign *et al.*, (2018), Birhanu *et al.*,

(2019a) and Birhanu *et al.*, (2019b) have made contributions to international conferences, namely ICOEI/2018, ICCT/2018, ICSSE/2018, ICIP/2019, and ICDAR/2019, respectively.

The other issue that the future of OCR system for Ge'ez and Amharic languages has to consider is incorporating post-processing task in the recognition process. Equal emphasis needs to be given to the post-correction methods such as dictionaries, statistical language models and natural language processing (NLP). Because the characters to be recognized are not random sequences, but they are meaningful words. It is these sequences of words that convey a meaning and form sentences which are syntactically, grammatically, and semantically coherent. For that reason, the final transcriptions are required to form sequences of dictionary words. Therefore, for a better recognition, equal emphasis needs to be given to all the tasks in the recognition process including post-processing task.

Taking all the above points into consideration, generally, the focus of future researches on OCR to Ge'ez and Amharic scripts has to be directed towards achieving accuracy of human level of performance.

2.6.5. Summary

The early attempted of applying OCR techniques to the Amharic language was recorded in 1997 by Worku in Addis Ababa University at the then School of Information Studies for Africa (SISA). Following it, researches on the area of OCR to Amharic script had continued primarily at SISA in order to exploit the potential of OCR system. From its inception in 1997 until 2011, there was at least one research work of OCR for Ge'ez or Amharic script almost in a yearly manner. The subsequent five years (between 2011 and 2016), however, there was a decline for the reasons not known yet. The last three years period is an important period for the improvement of research works of OCR for the Ge'ez or Amharic script primarily because of the deep learning techniques. In the early time, mainly data storage and processing speed limited harnessing the full potential of OCR technology. In more recent time, these limitations do not affect any longer. Hence it opens the door to incorporate more complex deep learning techniques.

Extensive study of the previous related research works reveals that only four studies i.e. Yaregal and Bigun (2008), Siranesh (2016), Shiferaw (2017) and Fitehalew (2019) attempted to apply OCR to historical Ge'ez manuscripts. All of them focused towards

character level recognition. In light of the pattern recognition techniques, the studies applied hybrid (structural/syntactical approach), deep Multilayer perceptron (MLP), statistical approach, and deep convolutional neural networks (CNN), respectively. It can be observed that two of the above mentioned research works, i.e. Siranesh (2016) and Fitehalew (2019), purely implemented deep learning techniques.

In the current era of deep learning, there are three (3) key challenges primarily for the development of research works on OCR for Ge'ez and Amharic scripts. The three key challenges are: (a) lack of benchmarking dataset, (b) lack of computing facility, and (c) lack of awareness.

CHAPTER THREE

METHODS AND APPROACHES

Experimental research design with prototyping approach was employed to build the proposed handwriting recognition system since it is very helpful to improve the system through experiment.

3.1. Architecture of the Proposed Handwriting Recognition System

One of the best workflow preferred in designing handwriting recognition system is the one that allows modular and sequential processing of information. Typically, a given module in the proposed system takes an intermediate output of another module and transforms it into representations which make useful information more explicit. The proposed system is made based on real-world large scale digitization scenarios. Its architecture is comprised of tasks, namely: pre-processing (binarization and skew estimation), page layout analysis (page segmentation and region classification), recognition model (trained model), and post-processing.

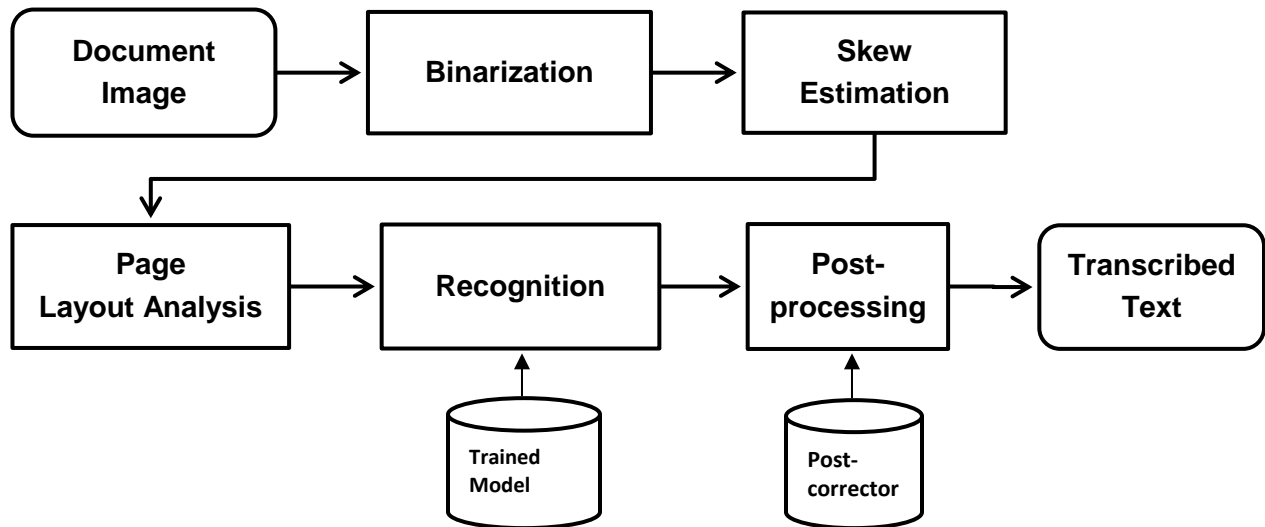


Figure 3. 1: Architecture of the proposed handwriting recognition system

As shown in the above figure 3.1, the architecture defines how the system is constructed, describes what the critical components are and how they fit together. The proposed system accepts scanned document image as an input. Binarization, skew estimation and page layout analysis (page segmentation and region classification) tasks are performed

sequentially prior to the text recognition. All the tasks are equally important and contribute to the final recognition process. Thus we need to perform an objective evaluation of the suitability of each module for the practical handwriting recognition problem.

The final stage is post-processing. Most popular post-processing methods include character error models, dictionaries, statistical language models and natural language processing (NLP) in isolation or combination of them.

3.2. Experimental Setup for Pre-processing

This experimental setup is designed with a goal mainly to select the best algorithm for the binarization task and also designed to investigate the skew estimation method. The best algorithm needs to be chosen by experimenting and examining the results.

3.2.1. Document Image Binarization

Based on their good records in the document image analysis literature, four binarization methods are proposed in this experimental setup. Three of them are the standard approaches known as Otsu's global method, Otsu's local method and Sauvola's method. The remaining one is Gato's adaptive method (Otsu, 1979; Sauvola & Pietikäinen, 2000; Gatos, Pratikakis, & Peranto, 2004). The best method among from them needs to be chosen by experimenting and examining the results from a testing dataset. The testing dataset for the binarization of degraded documents with Ground truth are collected from the DIBCO contest held in 2019¹⁶. It consists of nine (9) images which have representative degradations of a similar problem at hand.

The widely used performance evaluation metrics known as F-Measure (FM), pseudo F-measure (ps-FM), Peak Signal-to-Noise Ratio (PSNR), and Distance Reciprocal Distortion (DRD) are used for the evaluation purpose. These evaluation metrics were also adopted by DIBCO in its latest international Document Image Binarization Contest held in 2019.

The steps involved in using DIBCO evaluation tool are:

¹⁶ <https://users.iit.demokritos.gr/~bgat/DIBCO2009/benchmark/>

1. Prepare the Ground Truth image and the corresponding binarized image for evaluation (E.g. T1_GT.tiff and T1_bin.bmp)
2. Run the program 'BinEvalWeights.exe' to generate the .dat files containing the "Recall/Precision weights" of the image.

Example Run:

```
C:\Users\BMT\Desktop\Mesfin\DIBCO2019\BinEvalWeights\BinEvalWeights\BinEvalWeights.exe T1_bin.bmp
```

3. Prepare '4 Inputs' that correspond to filenames of:
 - a. Ground Truth image (E.g. T1_GT.tiff)
 - b. Binarized image for evaluation (E.g. T1_bin.bmp)
 - c. "Recall Weights" .dat file (E.g. T1_RWeights.dat)
 - d. "Precision Weights" .dat file (E.g. T1_PWeights.dat)
4. Run the program 'DIBCO_metrics.exe'

Example Run:

```
C:\Users\BMT\Desktop\Mesfin\DIBCO2019\dibco_metrics\DIBCO_metrics\DIBCO_metrics T1_GT.tiff T1_bin.bmp T1_RWeights.dat T1_PWeights.dat
```

All the binarization methods and the evaluation metrics are discussed in detail in Section 2.3.1.

3.2.2. Document Image Skew Estimation

The detection and correction of document skew is one of the most important tasks in the document image analysis. Many document page segmentation algorithms are designed to process document images with zero skew. Thus we need to apply skew correction process prior to document page segmentation. Because it affects the handwriting recognition process indirectly. Therefore, skew estimation method known as Hough transform method is selected to process skew detection and correction. Hough transform is a widely known technique in computer vision and image analysis.

The steps involved in estimating the skew angle using the Hough transform method are:

1. Read the document image and convert it into binary image.
2. Find edges using Sobel filter.

5. Do the Hough transform to generate text lines of the image

For each black pixel in a binary image

For theta (-angle to +angle) do

$$\rho = x \cos(\theta) + y \sin(\theta)$$

6. Find peaks over the Hough transform by filtering the accumulation buffer in order to obtain the highest values.
7. Plot the generated lines in the Hough space.
8. Find theta (θ) between lines and the x-axis.
9. Compute accumulator array for theta.
10. Compute the skew angle which is the maximum value on accumulator array.

The quality of the Hough transform method is investigated by experimenting and examining the results over a dataset. The dataset¹⁷ with Ground truth is employed from the DISEC'13 competition which was an international Document Image Skew Estimation Contest (DISEC) organized in the context of ICDAR conference in 2013.

In order to measure the performance of the Hough transform method, three criteria are used: (a) the Average Error Deviation (AED), (b) the average error deviation of the Top 80% (TOP80), and (c) the percentage of Correct Estimation (CE) with the threshold of 0.1⁰. The threshold of 0.1⁰ was chosen due to the fact that a skew angle greater than this threshold may be visible to a human observer. These performance evaluation criteria were also adopted by DISEC'13 competition organized in the context of ICDAR conference in 2013.

Skew estimation methods and the criterion are discussed in detail in Section 2.3.2.

3.3. Experimental Setup for Page Layout Analysis

The main motivation behind this experimental setup is to evaluate the page layout analysis task using realistic document images and an objective performance analysis system known as Aletheia. A number of distortions frequently visible in digitized historical Ge'ez manuscripts are the major hurdles when building a complete OCR system for mass digitization of the historical documents. Because it is often not sufficient to simply segment

¹⁷ <https://users.iit.demokritos.gr/~alexpap/DISEC13/resources.html>

the scanned pages into text and non-text areas. A detailed page layout analysis consisting of the following points are required:

- Accurate semantic distinction of region types: image, text, paragraph and caption.
- A reading order that includes all text regions on a page.
- Accurate detection of text lines, words and glyphs.

Therefore, this experimental setup aims to investigate the performance of Leptonica which is an open source C library for efficient document or natural image analysis operations. Its performance is investigated using a testing set consists of five (5) document pages of historical Ge'ez manuscripts with various page layouts. Each document page got processed using Leptonica and the results are stored using the PageXML standards and compared against the ground truth created manually using Aletheia tool. As input: the ground truth XML file, the segmentation result XML file and the black-and-white document image are required. For the evaluation, the ground truth regions are compared to the segmentation result regions. Differences are logged as evaluation errors (such as merge, split, miss, partial miss, misclassification, false detection and overall error).

The page layout analysis task is discussed in detail in Section 2.3.3.

3.4. Experimental Setup for Training OCR Engine

Mainly due to the introduction of multilingual open source OCR engines (e.g. Tesseract OCR engine by Google), it is now possible to train models in order to recognize even historical documents with excellent accuracies. This section presents the experimental setup designed for training Tesseract OCR engine. Tesseract was selected as OCR engine primarily due to its support to Ethiopic script as well as its open source ethos and popularity with large scale digitization. The latest Tesseract uses LSTM network based models. The challenges at hand can be formulated as optimization problem, which hypothesized to maximize Tesseract's recognition model accuracy over a set of training samples from actual historical Ge'ez manuscripts. Training Tesseract OCR engine involves three main steps: training data preparation, running the training process, and performance evaluation of the newly trained recognition model.

3.4.1. Training Data Preparation

Ground truth creation of training samples from historical manuscripts for training OCR engine is one of the major challenges in the field. This section presents the procedure of the creation of training samples from actual historical Ge'ez manuscripts. The primary use of the ground truth is to train the Tesseract OCR engine, and also it is required for evaluating the performance of the newly trained recognition model. It is created with maximum precaution to ensure that clean error-free training samples are obtained. Creation of the ground truth is difficult by the fact that the training samples have to be extracted from the actual manuscripts. However, it can be realized efficiently by using tools with a minimum of human interaction.

Therefore, ground-truthing tool known as Aletheia is employed to generate text line images of sample data with the corresponding ground truth for training LSTM based Tesseract. Aletheia was selected primarily due to its efficiency and popularity with large scale digitization of historical documents.

3.4.2. Running the Training Process

Tesseract can be trained via a collection of command line tools and Linux shell scripts. The procedure is accomplished using OCR-D on Ubuntu 20.04. Training is made after installing all the required pre-requisite libraries. The output of the training is a "gez.traineddata" file. By convention, Tesseract models use (lowercase) three-letter codes defined in ISO 639 with additional information separated by underscore. Hence the recognition model for the Ge'ez language named as 'gez'.

3.4.3. Performance Evaluation

Performance evaluation metric known as Character Error Rate (CER) is used to evaluate the newly trained recognition model. The evaluation metric is discussed in detail in Section 2.3.4.

3.5. Hardware and Software Employed for Implementation

The experiments were performed using HP PRO 3500 Series MT with hardware specification: Intel® Core™ i3-3240 CPU @ 3.40 GHz, 4GB Ram, and x64-based processor. The operating system was Windows 10 Pro.

3.5.1. MATLAB

MATLAB version R2020b was used to implement the image processing libraries. Primarily due to its excellent documentation and community support, MATLAB was a natural choice for the implementation of image processing tasks.

3.5.2. MATLAB Runtime

MATLAB Runtime version 9.0 (R2015b) was used to run DIBCO evaluation tool.

3.5.3. Aletheia

Aletheia is an advanced tool for creating page layout and text ground truth for document images. Aletheia Pro version 1.2.4 is employed for the training data generation. It is known for its robustness and script independent in the process of document image analysis. It supports top-down as well as bottom-up workflows. It is developed in Prima research lab, University of Salford, United Kingdom (PRImA Research Lab, 2019).

3.5.4. Tesseract

Tesseract is a popular open-source LSTM-based multilingual OCR engine, developed initially by Hewlett Packard and later sponsored by Google. The original model has been improved and now reaching version 4.1.1 at the time of writing.

3.5.5. Ubuntu

Ubuntu is a complete Linux operating system, freely available with both community and professional support. Ubuntu 20.4 was used for running the training process with terminal commands.

3.5.6. Python

By default OCR-D toolkit uses Python 2 for training. However latest version can also be used. Hence Python 3 was used for running the training process.

CHAPTER FOUR

EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Results and Discussion of the Pre-processing

This section presents evaluation results to select the best algorithm or method for the binarization and skew estimation tasks. The best algorithm or method is chosen by experimenting and examining the results over a testing dataset based on objective evaluation metrics. Following it, comparison and discussion are made based on the experimental results.

4.1.1. Document Image Binarization

The testing dataset consists of nine (9) degraded images and their associated ground truth collected from the DIBCO competition held in 2019. The selection of the images in the testing dataset was made so that should contain representative degradation similar with the problem at hand. Evaluation results for each test image (T1, T2 ... and T9) with respect to the metrics used for the binarization methods are presented in Table 4.2. The evaluation was based upon the four distinct metrics presented in Section 2.2.1. The final ranking is shown in Table 4.1. It was calculated after first, sorting the accumulated ranking value for all measures for each test image. Thereafter, the summation of all accumulated ranking values for all test images denote the total score which is shown in Table 4.1.

At Table 4.2, for each encountered measure, the detailed performance of each algorithm is given. The final ranking as shown in Table 4.1, ('Total Score') was calculated by firstly, sorting the accumulated ranking value for all measures for each test image. The summation of all accumulated ranking values for all test images denote the final 'Total Score' which is shown in Table 4.1. Let $R^i(j,m)$ be the rank of the method i concerning the j^{th} image when using the m^{th} measure. Then, for each binarization method i , the Total Score S_i is given by the following Equation:

$$S_i = \sum_{j=1}^K \sum_{m=1}^L R^i(j,m) \quad (4.1)$$

where K is the number of images used in the evaluation (i.e. $K = 9$) and L is the number of the evaluation metrics (i.e. $L = 4$).

To calculate the 'Total Score' for Otsu's global method:

$$\text{Score}_{(\text{Otsu_Global})} \Rightarrow \mathbf{FM} = 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 = 36$$

$$\mathbf{ps-FM} = 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 = 36$$

$$\mathbf{PSNR} = 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 = 36$$

$$\mathbf{DRD} = 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 = 36$$

$$\text{Therefore, Total Score} = 36 + 36 + 36 + 36 = \mathbf{144}$$

To calculate the 'Total Score' for Otsu's local method:

$$\text{Score}_{(\text{Otsu_Local})} \Rightarrow \mathbf{FM} = 3 + 3 + 1 + 3 + 1 + 1 + 1 + 3 + 2 = 18$$

$$\mathbf{ps-FM} = 3 + 3 + 3 + 3 + 1 + 2 + 3 + 1 + 2 = 21$$

$$\mathbf{PSNR} = 3 + 3 + 3 + 3 + 1 + 1 + 3 + 2 + 2 = 21$$

$$\mathbf{DRD} = 3 + 3 + 3 + 3 + 1 + 1 + 3 + 3 + 3 = 23$$

$$\text{Therefore, Total Score} = 18 + 21 + 21 + 23 = \mathbf{83}$$

To calculate the 'Total Score' for Sauvola's method:

$$\text{Score}_{(\text{Sauvola})} \Rightarrow \mathbf{FM} = 2 + 1 + 3 + 1 + 3 + 2 + 2 + 1 + 1 = 16$$

$$\mathbf{ps-FM} = 2 + 1 + 2 + 2 + 3 + 1 + 1 + 3 + 1 = 16$$

$$\mathbf{PSNR} = 2 + 1 + 1 + 2 + 3 + 2 + 1 + 1 + 1 = 14$$

$$\mathbf{DRD} = 2 + 1 + 1 + 2 + 3 + 2 + 1 + 1 + 1 = 15$$

$$\text{Therefore, Total Score} = 16 + 16 + 14 + 15 = \mathbf{61}$$

To calculate the 'Total Score' for Gato's Adaptive method:

$$\text{Score}_{(\text{Adaptive})} \Rightarrow \mathbf{FM} = 1 + 2 + 2 + 2 + 2 + 3 + 3 + 2 + 3 = 20$$

$$\mathbf{ps-FM} = 1 + 2 + 1 + 1 + 2 + 3 + 2 + 2 + 3 = 17$$

$$\mathbf{PSNR} = 1 + 2 + 2 + 1 + 2 + 3 + 2 + 3 + 3 = 18$$

$$\text{DRD} = 1 + 2 + 2 + 1 + 2 + 3 + 2 + 2 + 2 = 17$$

Therefore, Total Score = 20 + 17 + 18 + 17 = 72

Table 4. 1: Overall evaluation results and the final ranking of the binarization methods

Method	FM	ps-FM	PSNR	DRD	Total Score	Overall Rank
Otsu_Global	7.64	7.60	8.77	27.79	144	4 th
Otsu_Local	71.97	76.02	14.94	9.41	83	3 rd
Sauvola	74.14	80.30	15.87	6.63	61	1st
Adaptive	72.84	78.64	15.01	6.69	72	2 nd

Note: The best results are shown in bold.

The detailed performance for each binarization methods is also given in Table 4.2.

Table 4. 2: Evaluation results of binarization for each test image with respect to the metrics used

Metrics	Method	T1	T2	T3	T4	T5	T6	T7	T8	T9
FM	Otsu_Global	14.9006	12.8416	8.1776	9.3087	8.3993	4.8511	1.4718	2.5214	6.2393
	Otsu_Local	44.4327	67.7965	48.9389	63.2103	85.3138	92.8829	80.8532	74.2212	90.0449
	Sauvola	53.63	73.3612	41.2534	76.1372	77.1447	91.3149	80.0514	79.9096	94.4817
	Adaptive	56.006	69.1116	47.2924	76.1213	81.9821	83.3187	79.9856	75.3027	86.4697
ps-FM	Otsu_Global	15.0728	12.6474	8.1534	9.2153	9.8798	5.7331	0	1.6175	6.0947
	Otsu_Local	44.4981	68.5106	49.7442	63.4585	85.5918	93.8663	87.2958	99.9766	91.1966
	Sauvola	53.8065	76.7713	50.5516	78.4874	77.2003	94.7663	92.3783	99.7318	99.043
	Adaptive	56.2669	71.631	52.8831	78.8612	82.3783	85.8218	91.2377	99.7611	88.9277
PSNR	Otsu_Global	4.5985	6.5215	7.3313	6.2235	9.4035	10.323	14.9592	11.3498	8.2283
	Otsu_Local	6.9445	11.3258	11.2705	10.4821	17.4052	21.5395	19.7167	17.6033	18.1682
	Sauvola	8.5715	12.8572	12.41	13.4457	15.0225	20.8373	20.0166	18.4707	21.2233
	Adaptive	9.0067	11.8798	12.3107	13.4924	16.362	17.5653	19.9165	17.7405	16.8086
DRD	Otsu_Global	45.1834	30.6362	50.3437	30.1252	20.7442	20.6821	7.2204	21.9757	23.1546
	Otsu_Local	27.1274	10.2824	20.3962	12.2289	3.3472	1.7144	3.2743	4.0699	2.2351
	Sauvola	18.2521	7.2583	14.0178	5.9303	5.9314	1.7531	2.3823	3.4831	0.67032
	Adaptive	15.8394	8.6869	15.0362	5.3471	3.9665	3.071	2.3915	3.9449	1.9633

Note: The best results are shown in bold.

The best overall performance is achieved by **Sauvola**. Example of binarization results of Sauvola's method is shown in Figure. 4.1.



(a)



MS 2657
Enoch; in Ge'ez. Ethiopia, late 15th c.

MS 2657
Enoch; in Ge'ez. Ethiopia, late 15th c.

(b)



(c)



(d)

Figure 4. 1: Example of binarization results of Sauvola's method

4.1.2. Document Image Skew Estimation

To measure the quality of the skew estimation using the Hough transform method, well-known experimental dataset from ICDAR 2013 Document Image Skew Estimation Contest (DISEC'13) is employed. The experimental dataset consists of 20 unique images with a total of 200 images. For each unique image 10 rotated samples are generated.

These rotation angles are randomly selected from the limited range (-15° , $+15^\circ$). The ground truth angle value is established manually for all samples. The following frequency table shows the ground truth angle distribution of the DISEC'13 experimental dataset.

Table 4. 3: Frequency table of the ground truth skew angle distribution

Bins	Midpoint	Absolute Frequency	Relative Frequency	Cumulative Relative Frequency	Density
[-15,-10)	-12.5	36	0.18	0.18	0.036
[-10,-5)	-7.5	32	0.16	0.34	0.032
[-5,0)	-2.5	30	0.15	0.49	0.03
[0,5)	2.5	37	0.185	0.675	0.037
[5,10)	7.5	33	0.165	0.84	0.033
[10,15]	12.5	32	0.16	1	0.032

The following Histogram shows the ground truth angle distribution of the DISEC'13 experimental dataset.

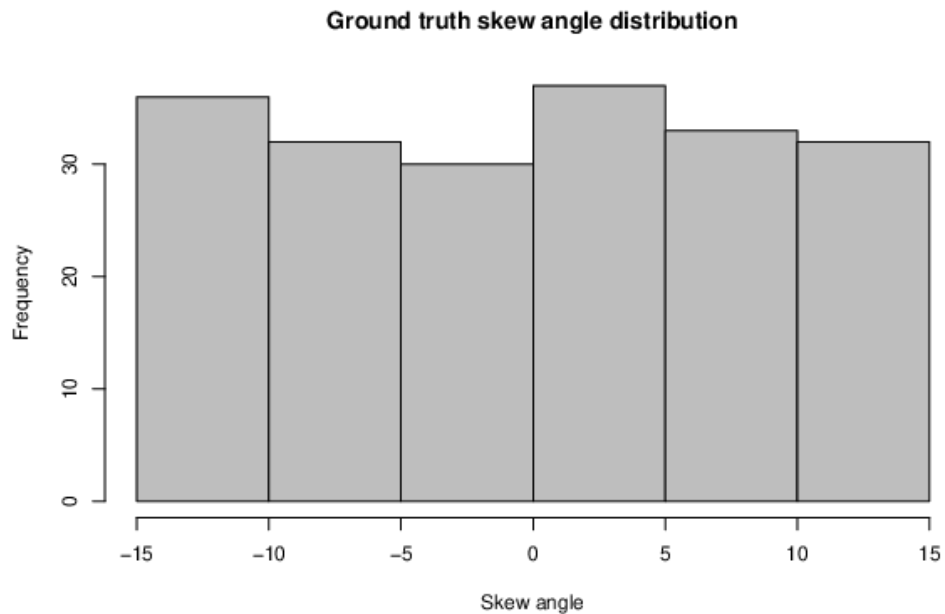


Figure 4. 2: The Histogram of angle values for the DISEC'13 experimental dataset

To measure the quality of the proposed Hough transform method, the DISEC'13 experimental dataset and its evaluation criterion AED, TOP80, and CE were used and obtained values equal to **0.3115**, **0.058**, and **76.00** respectively. The estimated skew

angle values using the proposed Hough Transform method is presented in terms of frequency table in Table 4.4.

Table 4. 4: Frequency table of estimated skew angles using Hough transform

Bins	Midpoint	Absolute Frequency	Relative Frequency	Cumulative Relative Frequency	Density
[-15,-10)	-12.5	34	0.17	0.17	0.034
[-10,-5)	-7.5	34	0.17	0.34	0.034
[-5,0)	-2.5	30	0.15	0.49	0.03
[0,5)	2.5	38	0.19	0.68	0.038
[5,10)	7.5	32	0.16	0.84	0.032
[10,15]	12.5	32	0.16	1	0.032

The following Histogram shows the estimated skew angle values using the proposed Hough transform.

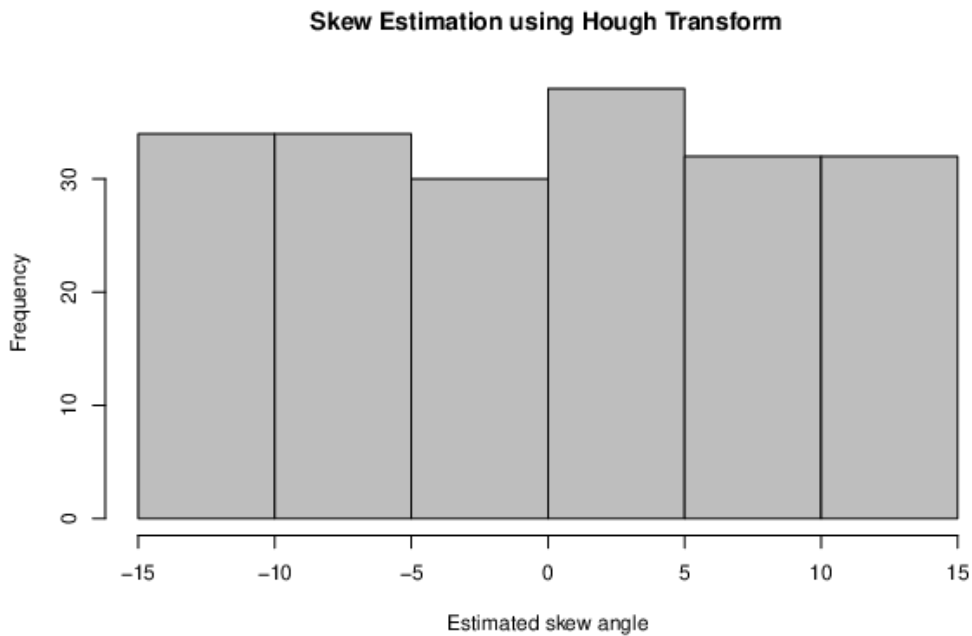


Figure 4. 3: The Histogram of the estimated skew angle values using Hough transform

4.1.3. Comparison

After a careful analysis of the binarization evaluation results presented in Table 4.1, it can be easily observed that the best performance is achieved by Sauvola's method. Sauvola's

method outperforms all other methods on all metrics that were used. Similarly, the second-ranked Gato’s Adaptive method is also the second-best using all the metrics that were used. On the other hand, Otsu’s global method achieves the worst performance on all metrics that were used. However, in the previous research works of OCR system for historical Ge’ez manuscripts such as in (Siranesh, 2016), (Shiferaw, 2017) and (Fitehalew, 2019) attempted to incorporate binarization technique based on mainly Otsu’s method. But, none of them had applied objective evaluation method which accounts for the performance of the binarization process.

When it comes to the skew estimation evaluation results, comparison is made with other skew estimation methods performed over the same dataset in Table 4.5. Huang *et al.* (2019) reported the performance of Projection profile and the Standard Hough Transform (SHT) methods on the same dataset. The SHT method was applied on extracted contours of objects using Canny filter. From Table 4.5, the proposed Hough transform method outperforms all the rest methods on TOP80 criterion with high margin and achieves relatively nearly the same performance with the Projection profile method on the AED criterion.

Table 4. 5: Evaluation results and comparison of skew estimation methods

Method	AED (°)	TOP80 (°)	CE (%)
Projection profile	0.290	0.195	-
SHT	6.120	4.374	-
Hough transform	0.3115	0.058	76.00

Note: The best results are shown in bold.

Though the previous research works of OCR system for historical Ge’ez manuscripts such as (Siranesh, 2016) and (Fitehalew, 2019) attempted to incorporate skew estimation technique based on Projection profile and Hough transform respectively, none of them had applied objective evaluation method which accounts for the performance of the skew estimation task.

4.1.4. Discussion of Results

The performed experiments have produced encouraging results, which ensure the applicability of empirical investigation of binarization and skew estimation methods. The

result demonstrates or has proven that global thresholding method is NOT sufficient on low quality and degraded historical document images. On other words, local or adaptive thresholding methods have demonstrated good records on low quality and degraded historical document images. Even though Sauvola's method outperforms all other methods on all metrics that were used, it requires further investigation to determine its optimal parameter values (i.e. window size and weight). Size of the testing set has also played a significant role to assess robustness of the method. For instance, the third-ranked Otsu's local method outperforms all other methods on all metrics in a single testing case, **T5**. However, the performance is not consistent to the rest of all testing cases.

On the other hand, the experimental results of the skew estimation method shows that the proposed Hough transform method has high precision, i.e. 76% correct estimation. The method was succeeded to perform under the well accepted threshold of 0.1^0 in the TOP80 criterion. This demonstrates that the method behaves accurately in its desired operation status as well as it shows the method is robust and treats most of the cases in the same way. However, the method fails to perform under the threshold of 0.1^0 in an AED criterion. But this can be improved with some optimization techniques which require further investigation.

4.2. Results and Discussion of the Page Layout Analysis

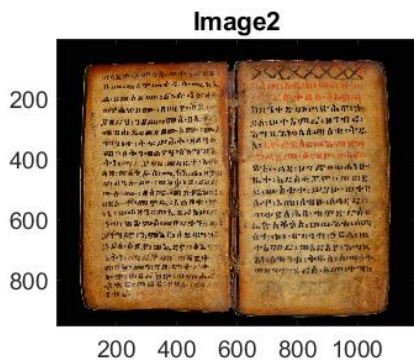
The effectiveness of Leptonica which is an open source C library for image analysis was investigated. Leptonica library can be used in conjunction with Tesseract for OCR. The official GitHub repository for Leptonica is: <https://github.com/DanBloomberg/leptonica>. In this experimentation, Leptonica's page segmentation and region classification performance over a testing set is investigated using Aletheia tool. The testing set consists of five (5) document images of realistic historical Ge'ez manuscripts with a wide variety of complex layouts and physical formats. The following figure shows the page layout description of each document image in the testing set.



Description:

Document Image1 consists of dimension of 800x870 pixels and a resolution of 600dpi. It has a text region of two columns, marginal note on the top, dispersed decoration at the end of the left column and contrast background. It has made of uncleaned parchment.

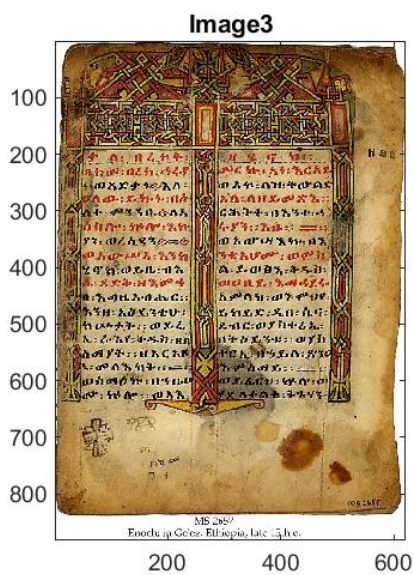
(a)



Description:

Document Image2 consists of two pages at a time and has a dimension of 1194x952 pixels and a resolution of 72dpi. It has a text region of a single column in each page and interlaced border decoration at the top of the right page.

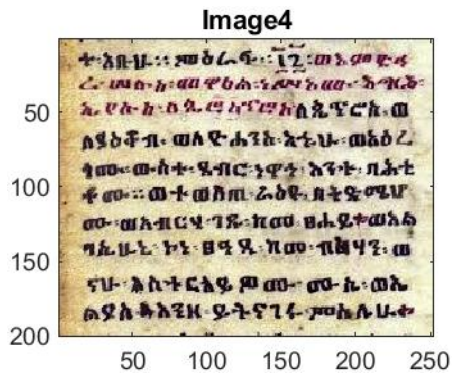
(b)



Description:

Document Image3 consists of dimension of 620x881 pixels and a resolution of 96dpi. It has a text region consists of two columns. It is highly decorated with drawings and has interlaced ornaments surrounding the text region & marginal notes. It has made of deteriorated parchment and noticeable water-blobs.

(c)



Description:

Document Image4 consists of dimension of 252x200 pixels and a resolution of 96dpi. It has a text region of one column.

(d)



Description:

Document Image5 consists of dimension of 296x170 pixels and a resolution of 96dpi. It has a text region of one column with different length of text lines. It has Interlaced border decoration at the top and noticeable degradation of faded ink and show-through.

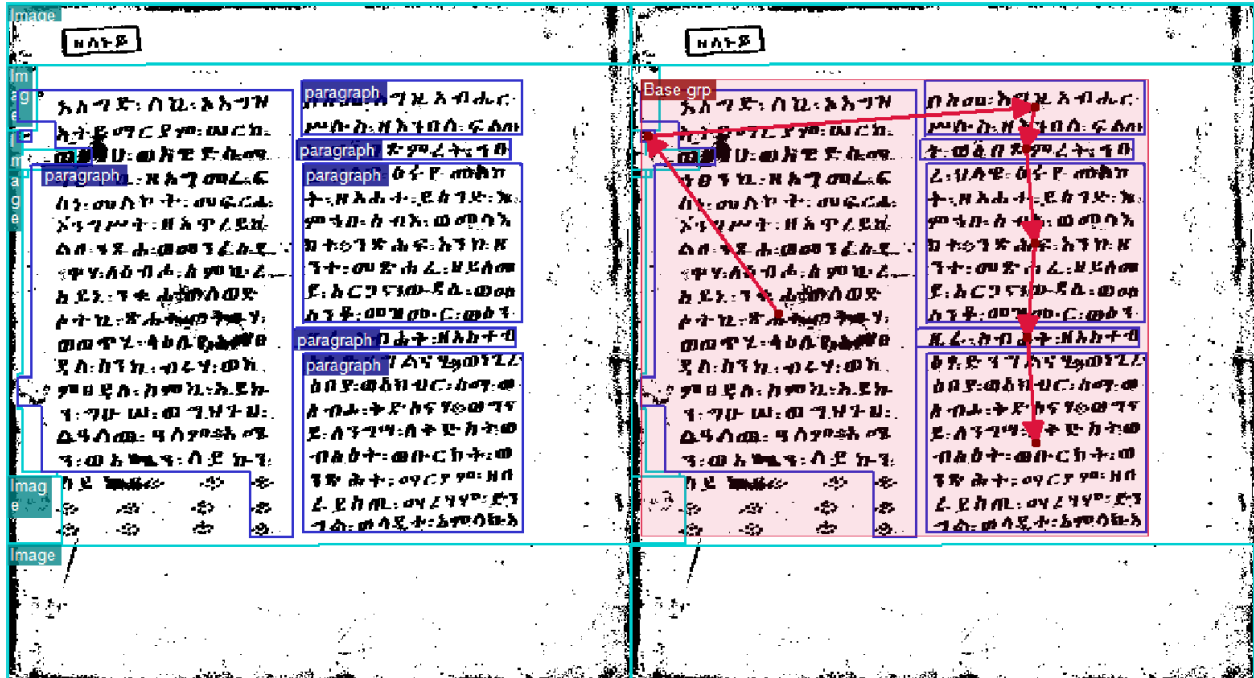
(e)

Figure 4. 4: Page layout description of each document image in the testing set

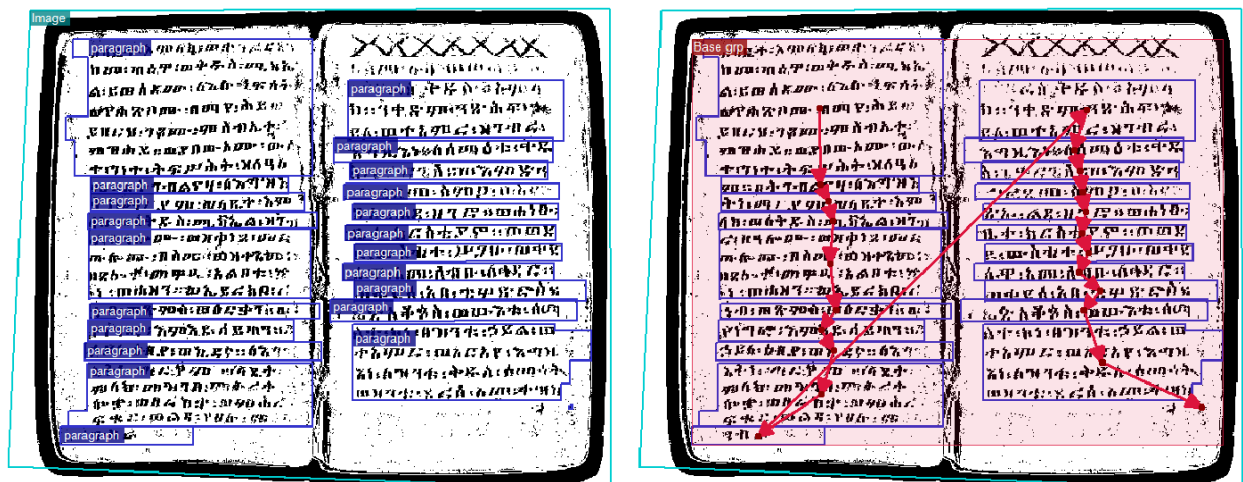
Before performing the page layout analysis, each document image in the testing set were pre-processed. Savoula’s method of binarization and the Hough transform based method of skew detection and correction were performed prior to the page layout analysis task. Both methods are selected based on their effectiveness in the previous experiments, described above in Section 4.1. The detected skew angle of each document image in the testing set were -0.2, -0.65, 2.32, 12.65 and 6.7 degrees correspondingly.

The figure below shows the semantic distinction of region types (image, text, paragraph and caption) and reading order that includes all text regions on the page. When it comes to the region classification, Leptonica performed accurately in Image1, Image2 and Image4 relatively. The worst region classification is manifested in Image3. The whole page is classified as an image region and a short length separator, though it has a text region consists of two columns. The possible reason the text region is not accurately classified is that it is highly decorated with drawings and has interlaced ornaments surrounding the text region. However, the caption is accurately identified. Similarly, most

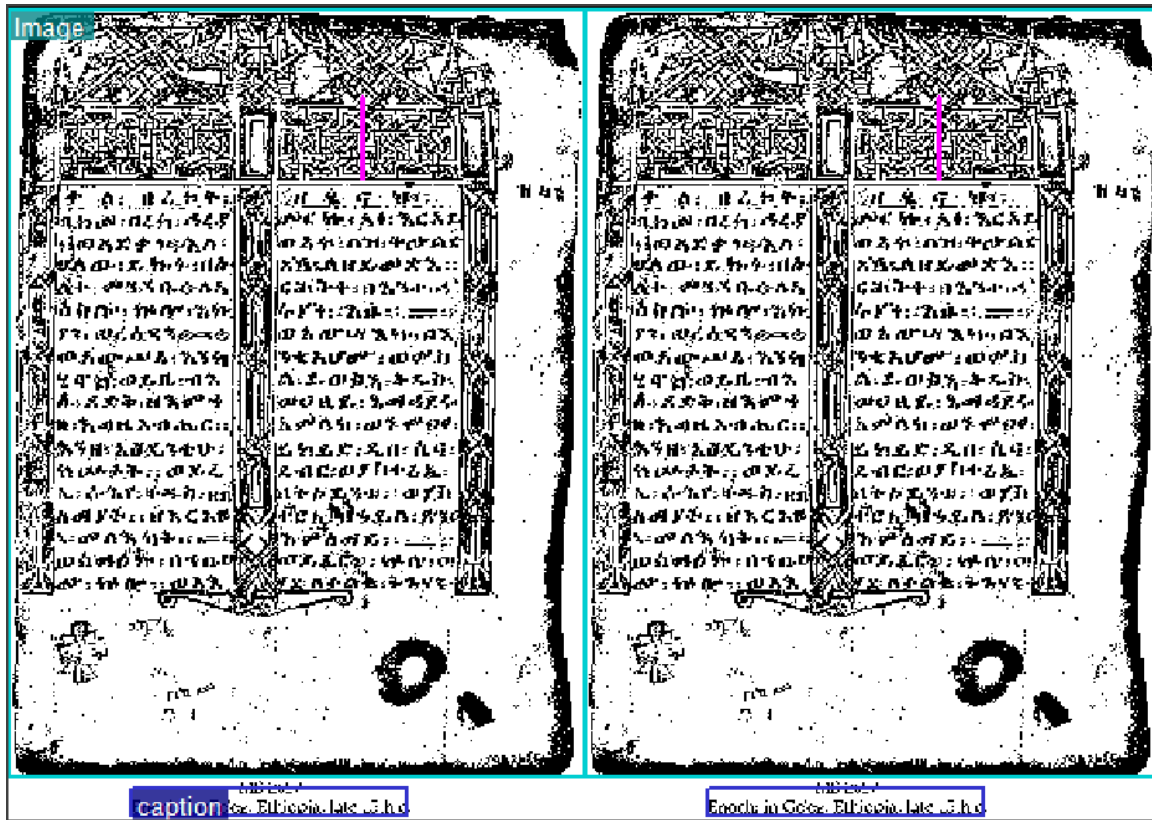
of the text region in Image5 has been misclassified as caption instead of paragraph. One of the possible reason this happens due to the surrounding of the portion of the text region with interlaced decoration on the top.



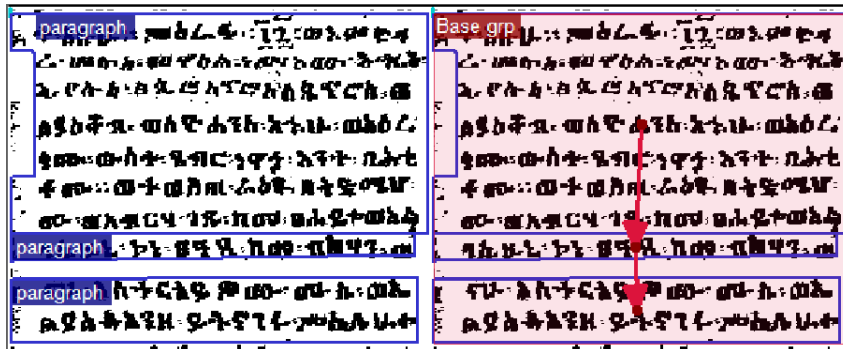
(a) In Image1, Leptonica accurately classified the text region. All the rest part including the marginal note on the top are classified as image zones. The arrows on the right side shows the reading order.



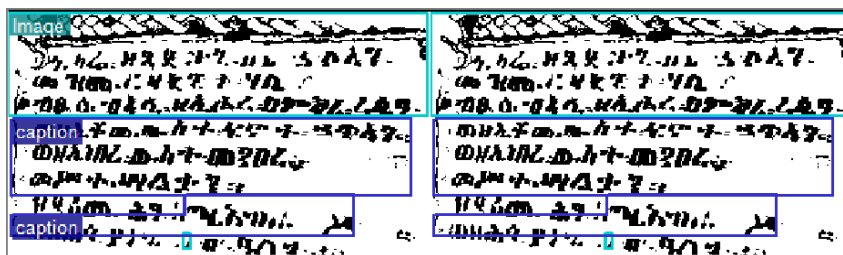
(b) In Image2, Leptonica accurately classified the text region. All the rest part is classified as image zone. The arrows on the right side shows the reading order.



(c) The whole Image3 is misclassified as an image region and a short length separator, though it has a text region consists of two columns.



(d) In Image4, Leptonica accurately classified the text region. The arrows on the right side shows the reading order.



(e) In Image5, most of the text region has been misclassified as caption instead of paragraph.

Figure 4. 5: Region classification and reading order of each image in the testing set

The other important issue that must be noted in page layout analysis task is text line detection in the text region. When it comes to complex layout and degraded historical documents, text line detection is still challenging task. In this experimentation, however, Leptonica has performed text line detection in acceptable manner. The figure below shows how Leptonica has performed text line detection in the tested document images.

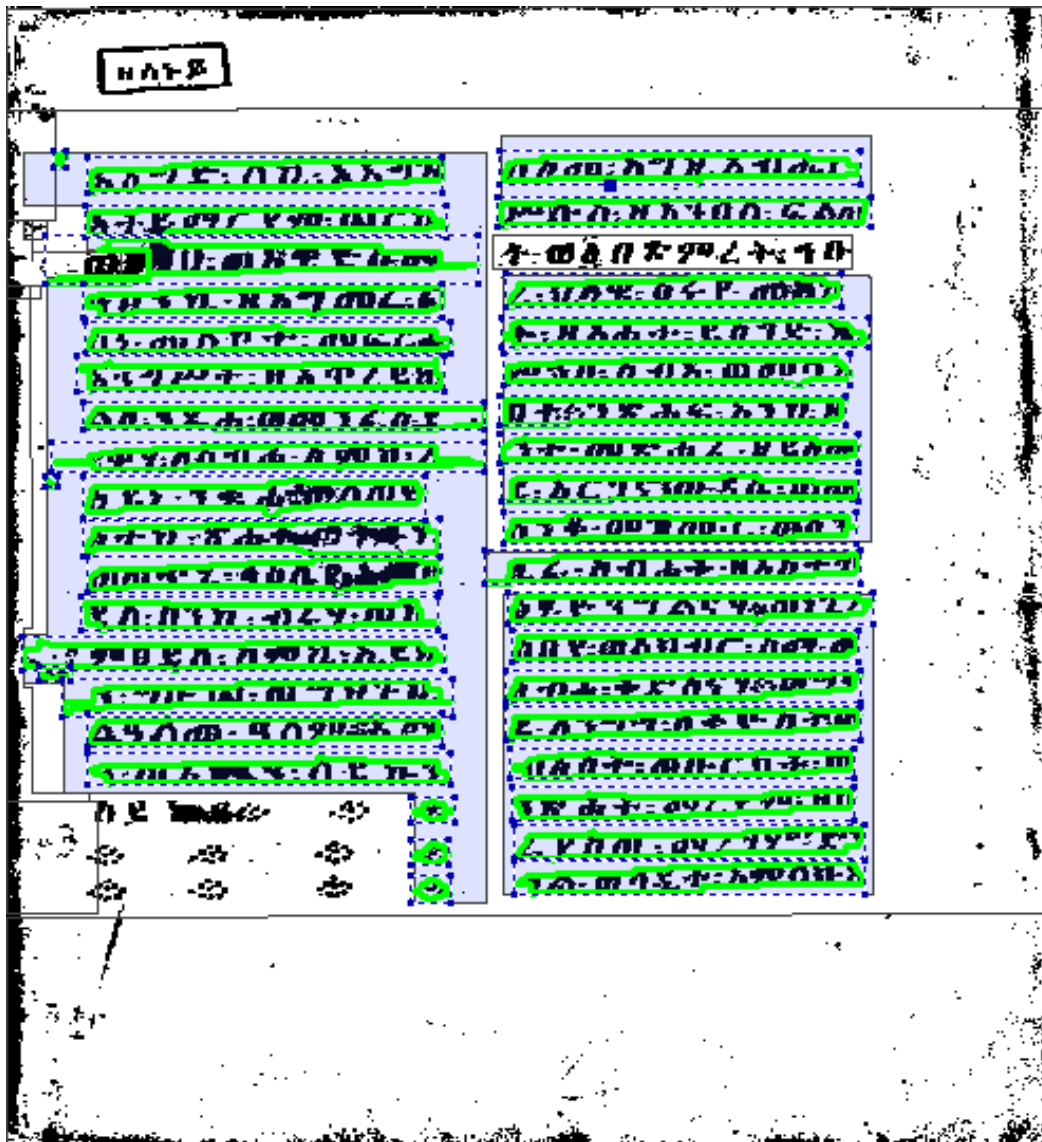


Figure 4. 6: Sample of text line detection in Image1

In addition to the text line detection, further investigation of word and glyph detection were performed. After a careful analysis of the preliminary investigation of the word and glyph detection, however, it has been observed that Leptonica has failed to perform at acceptable success rate in all the tested document images.

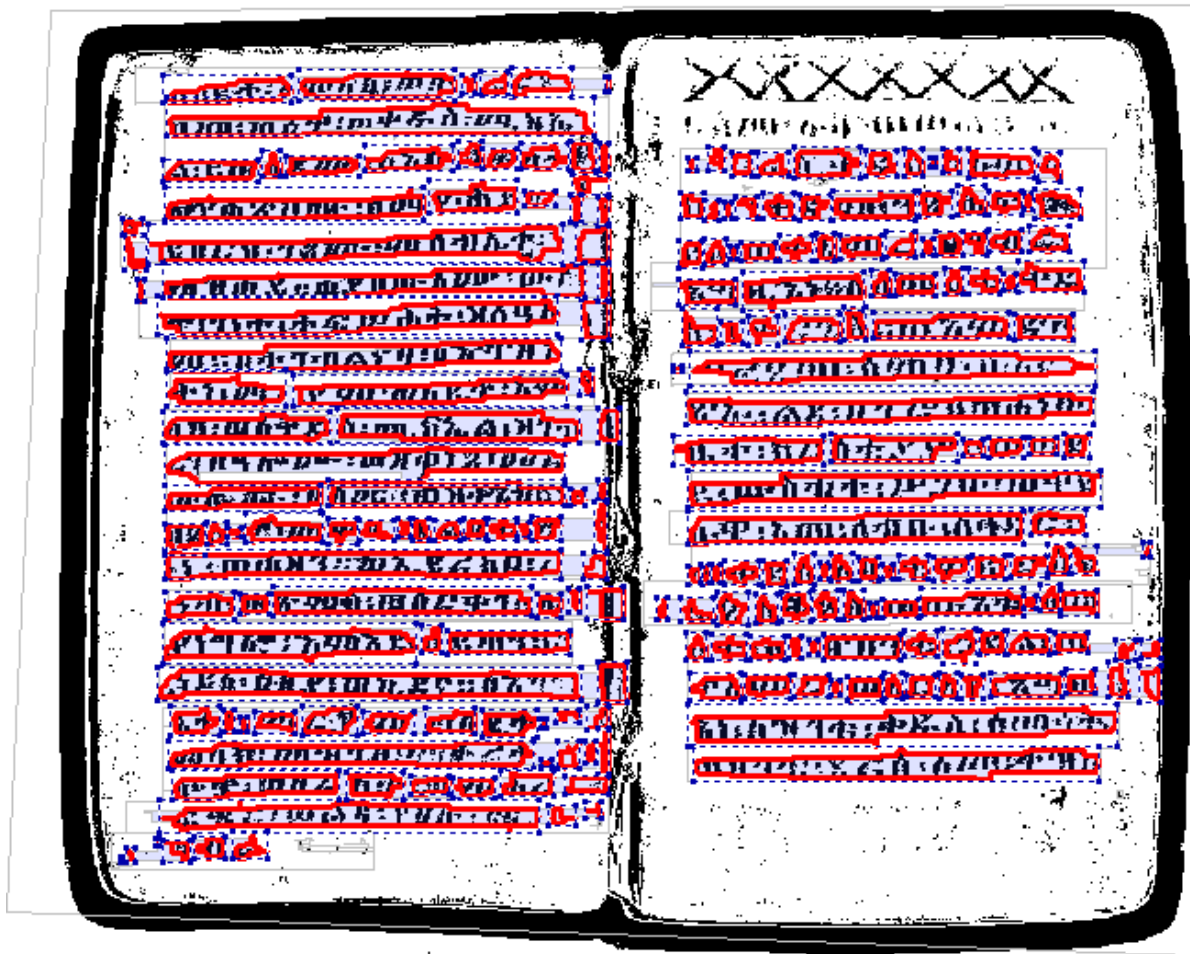


Figure 4. 7: Sample of word detection in Image2

In addition to the above qualitative description of the experimental results, objective evaluation method was also used in the page layout analysis. In order to use the objective evaluation method, each document page got processed using Leptonica and the results were stored using the PageXML standards and compared against the ground truth created manually using Aletheia tool. The ground truth XML file, the segmentation result XML file and the black-and-white document image were used as input for the evaluation. The ground truth regions were compared to the segmentation result regions. Differences were logged as evaluation errors (such as merge, split, miss, partial miss, misclassification, and false detection in terms of success rate). The default settings including weights were considered. The evaluate levels considered were regions, text

lines and groups. 'Plain.evx' option was used as evaluation profile. Finally, success rates of the following evaluation metrics are recorded.

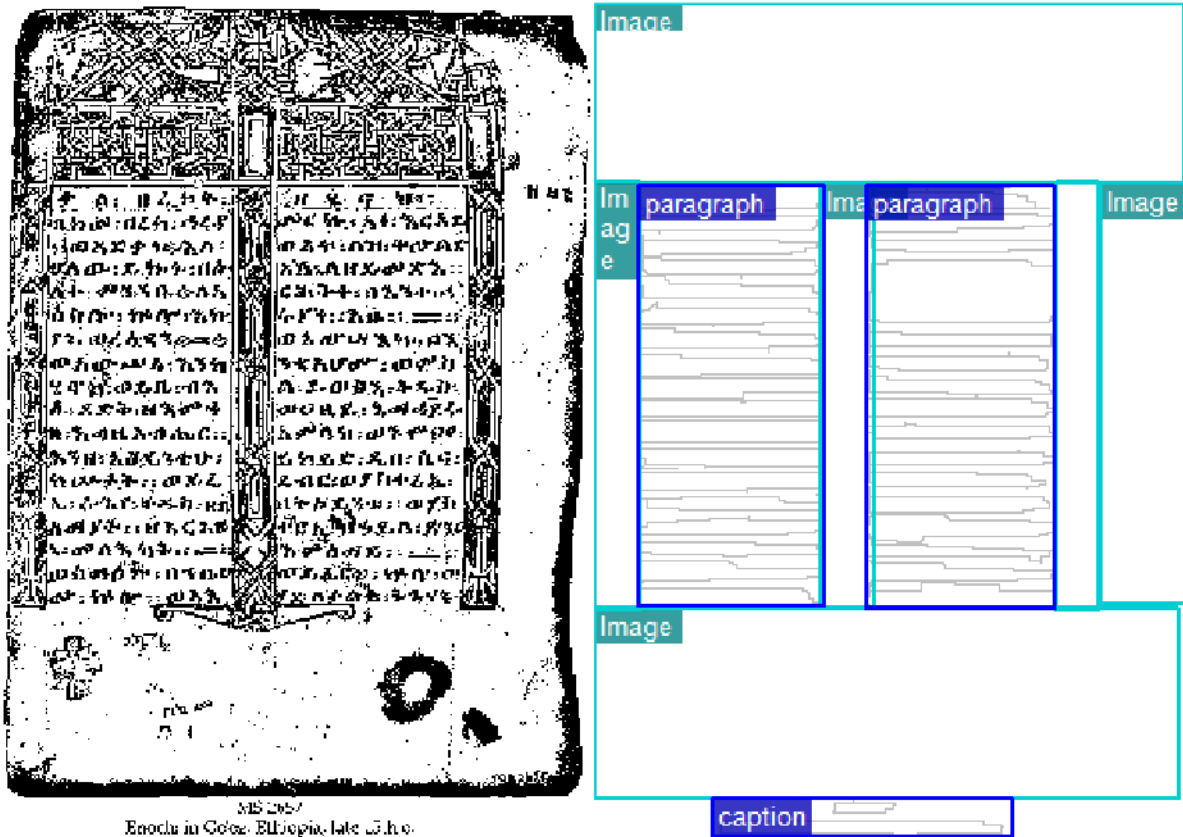


Figure 4. 8: Ground truth of Image3 created manually using Aletheia tool

Merges: Two or more ground truth regions are connected through a segmentation result region.

Splits: Ground truth regions that are overlapped by more than one segmentation result region.

Partial Misses: Ground truth regions that are not fully covered by the segmentation result.

Misclassifications: Segmentation result regions with other types than the corresponding ground truth regions.

False detections: Segmentation result regions with no corresponding ground truth region.

```

-<PcGts xsi:schemaLocation="http://schema.primaresearch.org/PAGE/gts/pagecontent/2018-07-15 http://schema.primaresearch.org/PAGE/gts
-<Metadata>
  <Creator/>
  <Created>2020-12-30T07:34:50</Created>
  <LastChange>2020-12-30T07:50:36</LastChange>
</Metadata>
-<Page imageFilename="1.tif" imageWidth="800" imageHeight="870">
  <ReadingOrder>
    <OrderedGroup id="ro357564684568544579089">
      <RegionRefIndexed regionRef="r0" index="0"/>
      <RegionRefIndexed regionRef="r1" index="1"/>
    </OrderedGroup>
  </ReadingOrder>
  <TextRegion id="r0" type="paragraph">
    <Coords points="66,586 63,586 63,564 62,564 62,549 66,549 66,506 63,506 63,473 60,473 60,458 58,458 58,454 63,454 63,418 60,4
55,276 56,276 56,273 61,273 61,230 60,230 60,207 58,207 58,192 52,192 52,186 58,186 58,157 63,157 63,123 116,123 116,121 140,1
276,108 276,116 278,116 278,117 333,117 333,134 327,134 327,162 333,162 333,165 337,165 337,169 332,169 332,196 352,196 352,
340,261 334,261 334,306 332,306 352,306 352,309 345,309 345,324 331,324 331,359 334,359 334,374 342,374 342,376 323,376 323,
350,442 318,442 318,469 330,469 330,494 335,494 335,499 332,499 332,530 341,530 341,532 333,532 333,557 332,557 332,583 337,
273,617 273,621 266,621 266,624 260,624 260,621 252,621 252,619 245,619 245,613 253,613 253,594 201,594 201,618 195,618 195,
-<ImageRegion id="r8">
  <Coords points="200,595 200,627 341,627 341,595"/>
  </ImageRegion>
  <TextLine id="r9">
    <Coords points="276,117 276,118 332,118 332,132 328,132 328,133 306,133 306,134 287,134 287,135 285,135 285,136 255,136 2
100,139 100,140 89,140 89,141 68,141 68,142 65,142 65,129 64,129 64,127 65,127 65,126 68,126 68,125 94,125 94,124 119,124 11
189,120 189,119 212,119 212,118 275,118 275,117"/>
  </TextLine>
  <TextEquiv>
    <Unicode>
  </TextEquiv>

```

Figure 4. 9: Snapshot of the Ground truth XML file of Image1

Table 4. 6: Page layout evaluation results of success rate for each document images

Evaluation Metrics	Image1		Image2		Image3		Image4		Image5	
	Region level	Text line level	Region level	Text line level	Region level	Text line level	Region level	Text line level	Region level	Text line level
Merge	73.89%	97.44%	100.00 %	26.33%	29.32%	24.37%	90.19%	99.22%	52.42%	35.19%
Split	49.97%	98.20%	40.60%	45.08%	99.72%	100.00 %	52.86%	99.89%	39.34%	85.72%
Miss	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	99.17%
Partial Miss	99.97%	96.89%	100.00 %	100.00 %	99.79%	99.28%	98.74%	99.92%	100.00 %	99.89%
Misclassification	96.82%		62.34%		63.97%		96.67%		41.31%	
False Detection	100.00 %	33.94%	100.00 %	99.90%	100.00 %	98.60%	100.00 %	99.94%	100.00 %	100.00 %
Overall success rate:										
Arithmetic mean	63.06%	66.27%	79.57%	51.82%	75.70%	59.06%	88.19%	99.79%	68.73%	66.64%
Harmonic mean	55.51%	50.63%	69.06%	39.29%	58.11%	37.43%	82.45%	99.79%	57.77%	52.62%

A careful analysis of the results of objective evaluation in Table 4.6 also ratifies the in-depth observational analysis described previously. For instance, in the case of Image4, Leptonica has performed region classification accurately at region and text line level as

shown in Figure 4.5 (d) which is also manifested with the highest success rate as shown in Table 4.6.

4.3. Results and Discussion of Training the OCR Engine

Training Tesseract OCR engine using training data was carried out and reported below, after a brief description of the experimental procedure. The results from the newly recognition model are compared to the original pre-trained or base model in order to assess the impact of the training.

4.3.1. Training Data Preparation

One of the challenges to apply deep learning technique is its requirement of huge amount of training data. In order to cope up with this challenge, however, there are multiple options. One of the multiple options for training is known as fine tuning. Instead of training from the scratch, fine tuning approach enables starting with an existing base model and then train on a specific additional data. Due to a difficulty to prepare large training data with ground truth from actual historical documents, fine tuning approach is getting popular nowadays. To address the problem, in this study, a similar approach is proposed and applied in the context of historical Ge'ez manuscripts.

The LSTM based Tesseract training process requires ground truth on text line level. Hence a training data is prepared from actual historical Ge'ez manuscripts. A total of 257 text line images collected from 15 different pages are prepared using Aletheia tool. The corresponding ground truth is UTF-8 encoded text line with text files.



Figure 4. 10: Sample of scanned pages

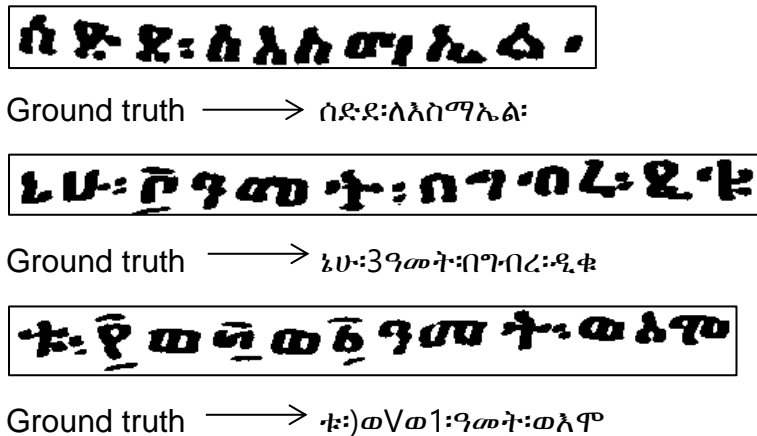


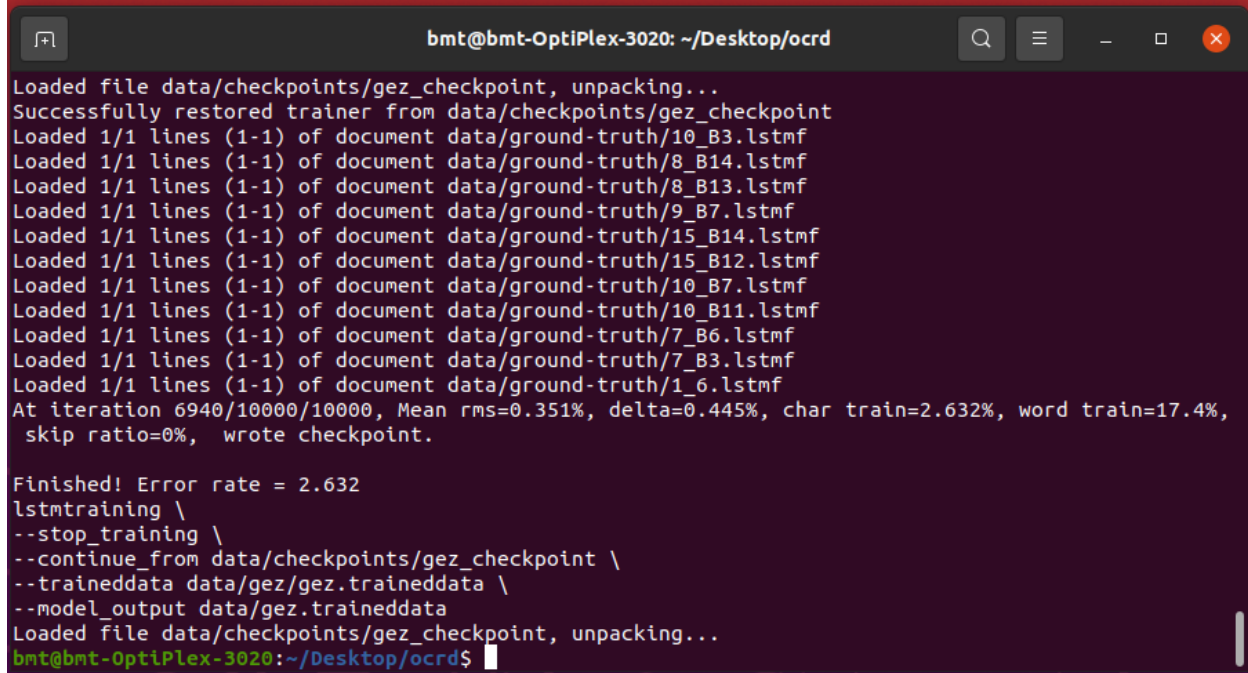
Figure 4. 11: Sample of text line images with their corresponding ground truth

4.3.2. Running the Training Process

The Tesseract training process requires ground truth on text line level. It is trained via a collection of command line tools and Linux shell scripts. Training is made after installing all the required pre-requisite libraries. The procedure is accomplished using OCR-D on Ubuntu 20.04. The procedure is composed of the following steps:

1. Download base model: amh.traineddata data file of tessdata-best type from https://github.com/tesseract-ocr/tessdata_best/blob/master/amh.traineddata.
2. Download OCR-D open source library from <https://github.com/kevinbicycle/ocrd-train>
3. Install the C++ compiler: `$ sudo apt install g++`
4. Install the latest Tesseract: `$ sudo apt-get install tesseract-ocr`
5. Provide training images and their ground truth text labels to OCR-D: Provide images with `.tiff` format and their labels in `.gt.txt` extension. Images should contain only one line of text image, and their ground truth labels also should contain only one line of text. We need to place all these files in data/ground-truth folder inside OCR-D.
6. Install Ubuntu make: `$ sudo apt install make`
7. Open terminal in the OCR-D folder
8. Execute the following command to start the training: `$ sudo make training`

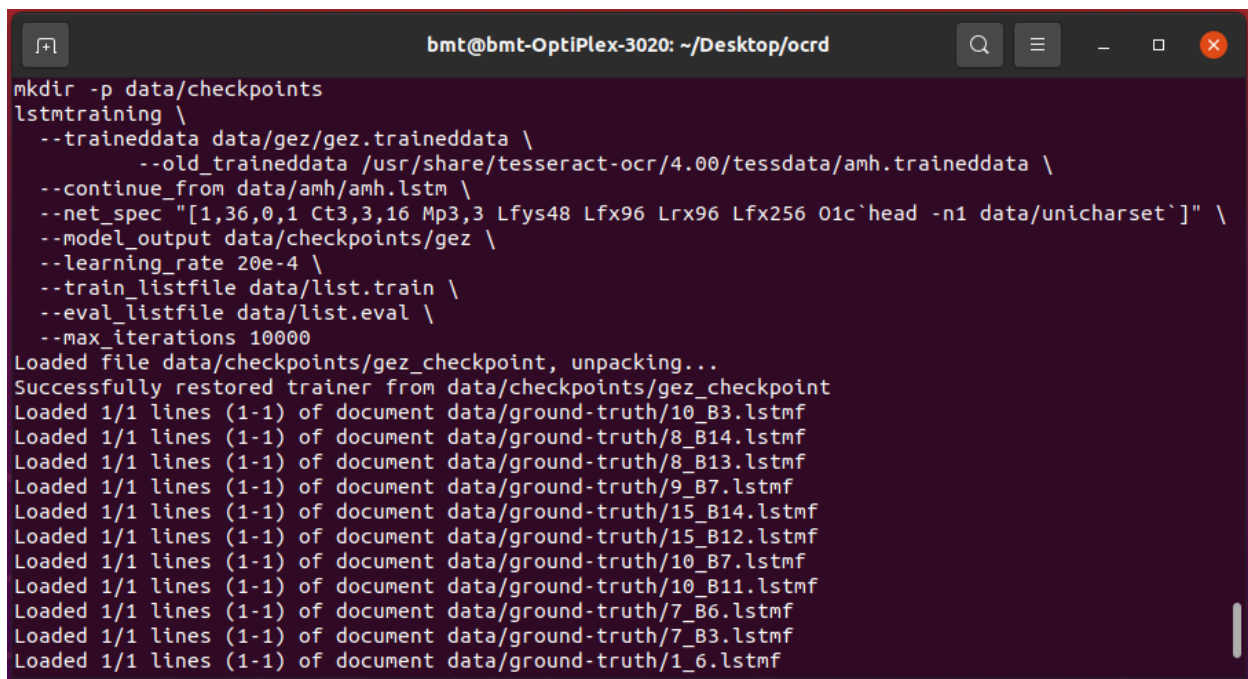
After the training is finished, the output on the terminal looks like the following.

A terminal window titled 'bmt@bmt-OptiPlex-3020: ~/Desktop/ocrd' showing the output of the training process. The output includes: 'Loaded file data/checkpoints/gez_checkpoint, unpacking...', 'Successfully restored trainer from data/checkpoints/gez_checkpoint', a list of 11 ground-truth files loaded, 'At iteration 6940/10000/10000, Mean rms=0.351%, delta=0.445%, char train=2.632%, word train=17.4%, skip ratio=0%, wrote checkpoint.', 'Finished! Error rate = 2.632', and the command 'lstmtraining --stop_training --continue_from data/checkpoints/gez_checkpoint --traineddata data/gez/gez.traineddata --model_output data/gez.traineddata' followed by 'Loaded file data/checkpoints/gez_checkpoint, unpacking...'.

```
bmt@bmt-OptiPlex-3020: ~/Desktop/ocrd
Loaded file data/checkpoints/gez_checkpoint, unpacking...
Successfully restored trainer from data/checkpoints/gez_checkpoint
Loaded 1/1 lines (1-1) of document data/ground-truth/10_B3.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/8_B14.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/8_B13.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/9_B7.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/15_B14.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/15_B12.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/10_B7.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/10_B11.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/7_B6.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/7_B3.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/1_6.lstmf
At iteration 6940/10000/10000, Mean rms=0.351%, delta=0.445%, char train=2.632%, word train=17.4%, skip ratio=0%, wrote checkpoint.

Finished! Error rate = 2.632
lstmtraining \
--stop_training \
--continue_from data/checkpoints/gez_checkpoint \
--traineddata data/gez/gez.traineddata \
--model_output data/gez.traineddata
Loaded file data/checkpoints/gez_checkpoint, unpacking...
bmt@bmt-OptiPlex-3020:~/Desktop/ocrd$
```

Figure 4. 12: The output on the terminal after the training is finished

A terminal window titled 'bmt@bmt-OptiPlex-3020: ~/Desktop/ocrd' showing the command used to start the training process. The command is 'lstmtraining --traineddata data/gez/gez.traineddata --old_traineddata /usr/share/tesseract-ocr/4.00/tessdata/amh.traineddata --continue_from data/amh/amh.lstm --net_spec "[1,36,0,1 Ct3,3,16 Mp3,3 Lfys48 Lfx96 Lrx96 Lfx256 01c`head -n1 data/unicharset`]" --model_output data/checkpoints/gez --learning_rate 20e-4 --train_listfile data/list.train --eval_listfile data/list.eval --max_iterations 10000'. The output is the same as in Figure 4.12, showing the files loaded and the training progress.

```
bmt@bmt-OptiPlex-3020: ~/Desktop/ocrd
mkdir -p data/checkpoints
lstmtraining \
--traineddata data/gez/gez.traineddata \
--old_traineddata /usr/share/tesseract-ocr/4.00/tessdata/amh.traineddata \
--continue_from data/amh/amh.lstm \
--net_spec "[1,36,0,1 Ct3,3,16 Mp3,3 Lfys48 Lfx96 Lrx96 Lfx256 01c`head -n1 data/unicharset`]" \
--model_output data/checkpoints/gez \
--learning_rate 20e-4 \
--train_listfile data/list.train \
--eval_listfile data/list.eval \
--max_iterations 10000
Loaded file data/checkpoints/gez_checkpoint, unpacking...
Successfully restored trainer from data/checkpoints/gez_checkpoint
Loaded 1/1 lines (1-1) of document data/ground-truth/10_B3.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/8_B14.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/8_B13.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/9_B7.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/15_B14.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/15_B12.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/10_B7.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/10_B11.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/7_B6.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/7_B3.lstmf
Loaded 1/1 lines (1-1) of document data/ground-truth/1_6.lstmf
```

Figure 4. 13: The output on the terminal that shows the lstmtraining

The following default parameter values were used during the training process:

MAX_ITERATIONS	Max iterations. Default: 10000
----------------	--------------------------------

```

LEARNING_RATE      Learning rate. Default: 0.0001 with START_MODEL, otherwise 0.002
PSM                 Page segmentation mode. Default: 6
RANDOM_SEED         Random seed for shuffling of the training data. Default: 0
RATIO_TRAIN        Ratio of train / eval training data. Default: 0.90
TARGET_ERROR_RATE  Stop training if the character error rate (CER in percent) gets
below this value. Default: 0.01

```

The output of the training is a “gez.traineddata” file. By convention, Tesseract models use (lowercase) three-letter codes defined in ISO 639 with additional information separated by underscore. Hence the newly trained recognition model for the Ge’ez language named as ‘gez’ as per the convention.

4.3.3. Performance Evaluation

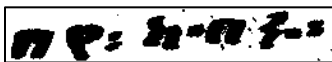
The Character Error rate of the newly trained recognition model, i.e. the ‘gez.traineddata’ is **2.632%**.

4.3.4. Comparison

To assess the impact of the training, OCR results with and without training were compared. Here, without training means using the base model, i.e. ‘amh.traineddata’. In order to use both recognition models, we need to copy them to tessdata directory. On other words, ‘amh.traineddata’ and ‘gez.traineddata’ files need to be copied to the data folder of the Tesseract instance that will be used to perform OCR. The following text line images from the evaluation list are randomly selected and examined to compare the OCR results of both models.



Ground truth → በስመ:አብ:ወወልድ:ወ
 Recognition using the base model → ያፕሊየወር,፤ደ ወ:ወ:ከመመሪሃልውጅ:ው:፤:ና:መጭሯ
 Recognition using the new model → ዘስለመ:አብ:ወወሰይሳድ:ወ
 (a)



Ground truth → በየ:ክብሩ:
 Recognition using the base model → ጋዋጅ:ጃቋዎም ::
 Recognition using the new model → በየ:አብድ:ያ:
 (b)

C: ኢ ወ ደ ቀ

Ground truth → C:ኢ.ወ.ደ.ቀ::
 Recognition using the base model → C::ጹ.ወ.ጅ::
 Recognition using the new model → C:ኢ.ወ.ጅ.ቀ::
 (c)

The comparison is also made on degraded low quality document image as follows.


Document Image	Recognition using the base model	Recognition using the new model
↓	↓	↓
	<p>እ ገብዙሁ፡ 4 8 ክፍሎ-ዳ፡፡ኸ ፲5፡፡ወ3፡ መና።</p> <p>6 ሀወ፡፡ቆ፡ ሞስሐ፡፡ መገ፡፡ በጽ፡ ጁጸ፡፡ማርጅ፡፡</p> <p>ሐላ፡፡የጽ፡፡ስ8 እ እና፡፡ለለቤ፡፡ኖር፡፡ኔ።</p> <p>ጸዩ0ቻ8. ሸለሞ ሐገሌ-እገሁ፡ ወክዕይረዕ</p> <p>ዳሙ-ውስ4-ጸ-በፎ፡፡ነዋታ እንቶ በሐቲ ፡ ቶቆው፡፡ውተወጠሸጠ-ራ5ዩ. ጸዳ%ማ</p> <p>፡" ው፡፡ጩ፡፡እወ፡፡4-4ጁ፡ ሸሀ9ከጩዴቱወኔ.</p> <p>፤ ሻሌ.ሀ.ኔ"ገ፡፡ኔ-ሸማ...ሸሸ09-ሸከግገ- ፡ጩ</p> <p>፪ ናሁ-እኪ-ትርቴዩ ቻ ሀፀ-ሀፀ-እ፡ ወወሊ</p> <p>፤፤ ጸ፡፡ቀ፡፡ወክሌ-ክገክ-9ትኖጊ- ያመሌሊህዮ</p>	<p>እቀለዙሁ፡ ፡መስርዳ.1 እዳ፡፡ወለ1መግ ራ፡፡መሠ፡፡እ፡፡ሦስሐ፡፡1፡፡እመ፡፡እማር፡፡</p> <p>እ፡፡የለ፡፡እለእና፡፡ለለበ.ኖር፡፡ኔ፡፡በ</p> <p>ለእቅብ፡፡ጠለኖሑ.ሐነእእ'ህ፡፡ወኔዕራ፡ ላው፡፡ውስት፡፡ግብርነኖታ፡፡እላት፡፡በሐቲ ፡ ቆ፡፡ውተመወመስጠ፡፡ራዕ፡፡ጸእ፡፡ሚሆ፡ መ፡፡ዘእቁርዛ'ነዜ፡፡ከመ፡፡በእይቀወእ</p> <p>፤ ዳሌ.ሊ.ኪ.ኔ፡፡ነ፡፡በግለከው፡፡በከግሣገ፡፡ው</p> <p>፤ ናሁ፡፡እስ'፡፡ኖር፡፡እ፡፡ፊ፡፡ዋመ፡፡ው፡፡እ፡፡ወሊ ነ እይስእእናዘዘ፡፡ዩ.ዳ.0ገ4ሥእሊህቀ</p>

Figure 4. 14: Comparison of OCR performance by the base and new models

4.3.5. Discussion of Results

The method demonstrates that the character error rate of Tesseract with base model 'amh.traineddata' can be decreased by continuing training using very few training data in the context of historical Ge'ez manuscripts. From a qualitative point of view, the change in the error rate is substantial. For a better accuracy, however, it requires a larger training sample.

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1. Conclusions

The performed experiments have produced encouraging results, which ensure the applicability of the proposed investigation. The study has empirically showed the performance of a set of algorithms and techniques in the document image analysis and recognition on building the handwriting recognition system for historical Ge'ez manuscripts. The proposed investigation has different strengths and a major weakness.

The performed experiments with the prototyping approach have produced encouraging results so that integrated Tesseract OCR engine and Leptonica library can be an ideal solution for a complete OCR system development for historical Ge'ez manuscripts. On other words, Tesseract which is fully implemented in the C++ programming language can be integrated with the Leptonica library which is implemented in the C language.

The major weakness is optimization. Primarily in the task of binarization, Sauvola's method was not optimized. Similarly, the Hough method of skew detection and correction was not optimized. The trained model also was not optimized with large dataset. Therefore, further optimization technique with large training sample is required. Moreover, in all the tasks, Ground truth is the basis for objective performance evaluation methods. Accurate Ground truth, however, is crucial for the evaluation.

In addition to the above mentioned optimization problem which require further investigation, extension works that need further considerations pertaining to the proposed handwriting recognition system are presented in detail in the following section.

5.2. Future Works

Still there is a long way to go in investigating handwriting recognition problem for historical documents. Extension works that need further consideration in the future to advance the current works in the handwriting recognition problem for historical Ge'ez manuscripts are explained as follows.

Handwriting is a form of language representation. Usually, the characters to be recognized are not random sequences, but they are meaningful words. Similarly, sequences of words normally convey a meaning, and form sentences that are syntactically, grammatically, and semantically coherent. On other words, the final transcriptions are required to form sequences of dictionary words. As a future work, investigation needs to consider incorporating Natural Language Processing (NLP) or statistical language model into the recognition process.

The other issue that needs to be considered is that the powerful automatic feature extraction ability of deep learning reduces the need for a separate handcrafted feature extraction process. However, the effect of the combination of *auto-derived features* with some *handcrafted set of features* which have proven good for handwriting recognition needs to be investigated.

Though the state-of-the-art OCR engines have achieved reasonably high accuracy for printed documents, they may not perform well on degraded historical documents. Therefore, it is a wise decision to investigate on another alternative technique for the handwriting recognition problem of historical documents. Thus the other possible extension work is *keyword spotting*. The aim of the keyword spotting is to identify a particular set of handwritten words within the historical document image. The discriminative power of neural networks is interesting for keyword spotting because they are able to concentrate on identifying and distinguishing raw pixels of the keywords, while ignoring the rest.

References

- Abay, T. (2010). *Amharic Character Recognition System for Printed Real-Life Documents*. MSc. thesis, Addis Ababa University.
- Abeto, A. (2018). *Character Recognition of Bilingual Amharic-Latin Printed Documents*. MSc. thesis, Addis Ababa University.
- Acharya, T., & Ray, A. (2005). *Image Processing: Principles and Applications*. John Wiley & Sons, Inc. doi:10.1002/0471745790
- Adak, C. (2019). *A Study on Automated Handwriting Understanding*. PhD thesis, University of Technology Sydney. Retrieved from <https://opus.lib.uts.edu.au/bitstream/10453/134139/2/02whole.pdf>
- Amsalu, T. (2015). *Introduction to Ethiopian philology*. Addis Ababa: Unpublished.
- Bausi et al. (2015). *Comparative Oriental Manuscript Studies: an Introduction*. Hamburg: COMSt. Retrieved from <http://dx.doi.org/10.5281/zenodo.46784>
- Bender, M., Cooper, R., & Ferguson, C. (1972). Language in Ethiopia: Implications of a Survey for Sociolinguistic Theory and Method. *Language in Society*, 1(2), 215-233. Retrieved from <https://www.jstor.org/stable/4166685>
- Bengio et al. (2006). Greedy layer-wise training of deep networks. *NIPS'06: Proceedings of the 19th International Conference on Neural Information Processing Systems*, (pp. 153–160). Retrieved from <https://proceedings.neurips.cc/paper/2006/file/5da713a690c067105aeb2fae32403405-Paper.pdf>
- Berhanu, A. (1999). *Amharic Character Recognition Using Artificial Neural Networks*. MSc. thesis, Addis Ababa University.
- Betselot, Y., Rana, D., & Bhalerao, G. (2018). Amharic Handwritten Character Recognition using Combined Features and Support Vector Machine. *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE. doi:10.1109/ICOEI.2018.8553947
- Birhanu et al. (2019a). Factored Convolutional Neural Network for Amharic Character Image Recognition. *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. doi:10.1109/ICIP.2019.8804407
- Birhanu et al. (2019b). Amharic Text Image Recognition: Database, Algorithm, and Analysis. *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. doi:10.1109/ICDAR.2019.00205
- Birhanu et al. (2020). Amharic OCR: An End-to-End Learning. *Applied Sciences*, 10(3). Retrieved from <https://doi.org/10.3390/app10031117>

- Birhanu, H., Tewodros, A., & Stricker, D. (2018). Amharic Character Image Recognition. *18th IEEE International Conference on Communication Technology (ICCT)*. IEEE. doi:10.1109/ICCT.2018.8599888
- Bluche, T. (2015). *Deep Neural Networks for Large Vocabulary Handwritten Text Recognition*. PhD thesis, Université Paris-Sud. Retrieved from <https://tel.archives-ouvertes.fr/tel-01249405/document>
- Boudraa, O., Hidouci, W., & Michelucci, D. (2020). Using skeleton and Hough transform variant to correct skew in historical documents. *Mathematics and Computers in Simulation*, 389-403. doi:10.1016/j.matcom.2019.05.009
- Bourlard, H., & Wellekens, C. (1988). Links between Markov models and multilayer perceptrons. *NIPS'88: Proceedings of the 1st International Conference on Neural Information Processing Systems*, (pp. 502–510). Retrieved from <https://proceedings.neurips.cc/paper/1988/file/0777d5c17d4066b82ab86dff8a46af6f-Paper.pdf>
- Burger, W., & Burge, M. (2016). *Digital Image Processing*. Springer-Verlag London. doi:10.1007/978-1-4471-6684-9
- Cowell, J., & Hussain, F. (2003). Amharic Character Recognition using a fast signature based algorithm. *Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003*. (pp. 384-389). London, UK: IEEE. doi:10.1109/IV.2003.1218014
- Dereje, T. (1999). *Optical Character Recognition of Typewritten Amharic Text*. MSc. thesis, Addis Ababa University.
- Direselign, A., Liu, C.-M., & Ta, V.-D. (2018). Printed Ethiopic Script Recognition by Using LSTM Networks. *2018 International Conference on System Science and Engineering (ICSSE)*. IEEE. doi:10.1109/ICSSE.2018.8519972
- Drobac, S., & Lindén, K. (2020). Optical character recognition with neural networks and post-correction with finite state methods. *International Journal on Document Analysis and Recognition (IJ DAR)*, 23, 279–295. Retrieved from <https://link.springer.com/article/10.1007/s10032-020-00359-9>
- Duda, R., & Hart, P. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1). doi:10.1145/361237.361242
- Dumoulin, V., & Visin, F. (2018). *A guide to convolution arithmetic for deep learning*. arXiv preprint. Retrieved from <https://arxiv.org/abs/1603.07285v2>
- Ermias, A. (1998). *Recognition of Formatted Amharic Text Using Optical Character Recognition (OCR) Techniques*. MSc. thesis, Addis Ababa University.

- Fetulhak, A. (2019). Handwritten Amharic Character Recognition System Using Convolutional Neural Networks. *Engineering Sciences (NWSAENS)*, 14(2), 71-87.
doi:10.12739/NWSA.2019.14.2.1A0433
- Fischer, A. (2012). *Handwriting Recognition in Historical Documents*. PhD thesis, Universität Bern.
Retrieved from
https://www.researchgate.net/publication/259346163_Handwriting_recognition_in_historical_documents
- Fitehalew, A. (2019). *Ancient Geez Script Recognition Using Deep Convolutional Neural Network*. MSc. thesis, Near East University. Retrieved from <http://docs.neu.edu.tr/library/6801590343.pdf>
- Fitsum, D. (2011). *Developing Optical Character Recognition for Ethiopic Scripts*. MSc. thesis, Dalarna University. Retrieved from <http://du.diva-portal.org/smash/get/diva2:519067/FULLTEXT01.pdf>
- Gatos, B., Ntirogiannis, K., & Pratikakis, I. (2009). ICDAR2009 Document Image Binarization Contest (DIBCO 2009). *International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1375–82). IEEE.
- Gatos, B., Pratikakis, I., & Peranto, S. (2004). An Adaptive Binarization Technique for Low Quality Historical Documents. *DAS 2004: International Workshop on Document Analysis Systems VI* (pp. 102-113). Springer, Berlin, Heidelberg. Retrieved from https://doi.org/10.1007/978-3-540-28640-0_10
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Graves et al. (2006). Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *ICML '06: Proceedings of the 23rd international conference on Machine learning* (pp. 369–376). ACM. Retrieved from <https://doi.org/10.1145/1143844.1143891>
- Graves, A. (2008). *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Technische Universität München. Retrieved from <https://www.cs.toronto.edu/~graves/phd.pdf>
- Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6), 602-610. Retrieved from <https://doi.org/10.1016/j.neunet.2005.06.042>
- Graves, A., & Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in neural information processing systems*, (pp. 545–552).
- Graves, A., Fernandez, S., & Schmidhub, J. (2007). Multi-Dimensional Recurrent Neural Networks. Retrieved from <https://arxiv.org/abs/0705.2011v1>
- Hailemariam, M. (2003). *Handwritten Amharic Character Recognition: The Case of Postal Addresses*. Masters Thesis, School of Information Studies for Africa, Addis Ababa University.

- Halefom et al. (2019). A New Hybrid Convolutional Neural Network and eXtreme Gradient Boosting Classifier for Recognizing Handwritten Ethiopian Characters. *IEEE Access*, 8, 17804 - 17818. doi:10.1109/ACCESS.2019.2960161
- Hashizume, A., Yeh, P.-S., & Rosenfeld, A. (1986). A method of detecting the orientation of aligned components. *Pattern Recognition Letters*, 4(2), 125-132. doi:10.1016/0167-8655(86)90034-6
- Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Pearson Education, Inc.
- Hinton et al. (2012). Improving neural networks by preventing co-adaptation of feature detectors. Retrieved from <https://arxiv.org/abs/1207.0580v1>
- Hinton, G., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18, 1527–1554. Retrieved from <http://www.cs.toronto.edu/~fritz/absps/ncfast.pdf>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8). Retrieved from <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hopfield, J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, 79(8), pp. 2554-8. doi:10.1073/pnas.79.8.2554
- Huang et al. (2019). An Efficient Document Skew Detection Method Using Probability Model and Q Test. *Electronics*, 9(55). Retrieved from <https://doi.org/10.3390/electronics9010055>
- Jain, A., Duin, R., & Mao, J. (2000). Statistical Pattern Recognition: A Review. *Transactions on pattern analysis and machine intelligence*, 22(1). Retrieved from http://ssg.mit.edu/cal/abs/2000_spring/np_dens/classification/jain_pami_1_00.pdf
- Khan et al. (2020). *A Survey of the Recent Architectures of Deep Convolutional Neural Networks*. arXiv preprint. Retrieved from <https://arxiv.org/abs/1901.06032v7>
- Khurshid, K. (2009). *Analysis and Retrieval of Historical Document Images*. PhD thesis, Université Paris Descartes. Retrieved from https://www.researchgate.net/profile/Khurram_Khurshid2/publication/305238729_Analysis_and_retrieval_of_historical_documents_images/links/5785cc4c08aec5c2e4e1238b/Analysis-and-retrieval-of-historical-documents-images.pdf?origin=publication_detail
- Kim, G., Govindaraju, V., & Sriha, S. (1999). An architecture for handwritten text recognition systems. 2, 37–44. doi:10.1007/s100320050035
- Kitchenham et al. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology*, 52(8), 792-805. Retrieved from <https://doi.org/10.1016/j.infsof.2010.03.006>

- LeCun et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4). doi:10.1162/neco.1989.1.4.541
- Lins et al. (2017). Assessing Binarization Techniques for Document Images. *Proceedings of the 2017 ACM Symposium on Document Engineering*, (pp. 183-192). doi:10.1145/3103010.3103021
- Lu, H., Kot, A., & Shi, Y. (2004). Distance-reciprocal distortion measure for binary document images. *IEEE Signal Processing Letters*, 11(2), 228 - 231. doi:10.1109/LSP.2003.821748
- McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics volume*, 5, 115–133. doi:10.1007/BF02478259
- Memon et al. (2020). Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). *IEEE Access*, 8. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9151144>
- Mesay et al. (2019). Handwritten Amharic Character Recognition Using a Convolutional Neural Network. Retrieved from <https://arxiv.org/abs/1909.12943v1>
- Mesay, H. (2003). *Line Fitting to Amharic OCR: The Case Of Postal Address*. MSc. thesis, Addis Ababa University.
- Million, M. (2000). *A Generalized Approach to Optical Character Recognition (OCR) of Amharic Texts*. MSc. thesis, Addis Ababa University.
- Million, M., & Jawahar, C. (2005). Recognition of Printed Amharic Documents. *ICDAR Proceedings of the Eighth International Conference on Document Analysis and Recognition* (pp. 784–788). ACM. Retrieved from <https://doi.org/10.1109/ICDAR.2005.198>
- Niblack, W. (1985). *An introduction to digital image processing*. Standberg Publishing.
- Nielsen, M. (2019). *Neural Networks and Deep Learning*. Retrieved from <http://neuralnetworksanddeeplearning.com>
- Nigussie, T. (2000). *Handwritten Amharic Text Recognition Applied To The Processing Of Bank Checks*. MSc. thesis, Addis Ababa University.
- Nosnitsin, D. (2012). *Ethiopian Manuscripts and Ethiopian Manuscript Studies: a brief overview and evaluation*. Comparative Oriental Manuscript Studies.
- Nosnitsin, D. (2013). *Churches and Monasteries of Tagray: A Survey of Manuscript Collections*. Harrassowitz Verlag. Retrieved from https://www.harrassowitz-verlag.de/title_1086.ahtml
- Ntirogiannis, K., Gatos, B., & Pratikakis, I. (2013). Performance evaluation methodology for historical document image binarization. *Trans Image Process*, 22(2), 595–609. doi:10.1109/TIP.2012.2219550

- O'Gorman, L., & Kasturi, R. (1997). *Document Image Analysis*. IEEE Computer Society Executive Briefings. Retrieved from <https://www.cse.usf.edu/~r1k/DocumentImageAnalysis/DIA.pdf>
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* (pp. 62-66). IEEE. doi:10.1109/TSMC.1979.4310076
- Papandreou, A., Gatos, B., Louloudis, G., & Stamatopoulos, N. (2013). ICDAR 2013 Document Image Skew Estimation Contest (DISEC 2013). *2013 12th International Conference on Document Analysis and Recognition*. IEEE. doi:10.1109/ICDAR.2013.291
- Plamondon, R., & Srihari, S. N. (2000). On-line and Off-line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, No.1. Retrieved from https://cedar.buffalo.edu/papers/articles/Online_Offline_2000.pdf
- Postl, W. (1986). Detection of Linear Oblique Structures and Skew Scan in Digitized Documents. *8th ICPR*, (pp. 687-689). Paris.
- Pratikakis, I., Zagoris, K., Barlas, G., & Gatos, B. (2017). ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1395-1403). IEEE. doi:10.1109/ICDAR.2017.228
- Pratikakis, I., Zagoris, K., Karagiannis, X., Tsochatzid, L., & Mondal, T. (2019). ICDAR Marthot-Santaniello - Competition on Document Image Binarization (DIBCO 2019). *International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1547-56). IEEE.
- PRIMA Research Lab. (2019). *Aletheia User Guide*. University of Salford, UK. Retrieved from <https://www.primaresearch.org/www/assets/tools/Aletheia%20User%20Guide.pdf>
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408. doi:10.1037/h0042519
- Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning Representations by Back-Propagating Errors. *Nature*, 323(6088), 533-536. doi:10.1038/323533a0
- Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern Recognition*, 33(2), 225-236. doi:10.1016/S0031-3203(99)00055-2
- Scelta, G. (2001). *The Comparative Origin and Usage of the Ge'ez writing system of Ethiopia*. Retrieved from http://www.thisisgabes.com/documents/paper_gabriella.pdf
- Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673 - 2681. doi:10.1109/78.650093
- Sergew, H. (1982). Bookmaking in Ethiopia. *Bulletin of the School of Oriental and African Studies*, 45(1). Retrieved from <https://doi.org/10.1017/S0041977X00055154>

- Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13. Retrieved from <https://pequan.lip6.fr/~bereziat/pima/2012/seuillage/sezgin04.pdf>
- Shafii, M. (2014). *Optical Character Recognition of Printed Persian/Arabic Documents*. PhD thesis, University of Windsor. Retrieved from <https://scholar.uwindsor.ca/etd/5179/>
- Shiferaw, T. (2017). *Optical Character Recognition For Ge'ez Scripts*. MSc. thesis, University of Gonder.
- Siranesh, G. (2016). *Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Networks*. MSc. thesis, Addis Ababa University.
- Sutskever, I. (2013). *Training Recurrent Neural Networks*. PhD thesis, University of Toronto. Retrieved from https://www.cs.utoronto.ca/~ilya/pubs/ilya_sutskever_phd_thesis.pdf
- Taddesse, T. (1972). *Church and State in Ethiopia, 1270-1527*. Oxford University Press.
- Tallec, C., & Ollivier, Y. (2017). Unbiasing Truncated Backpropagation Through Time. Retrieved from <https://arxiv.org/abs/1705.08209v1>
- Tensmeyer, C., & Martinez, T. (2020). Historical Document Image Binarization: A Review. *SN Computer Science*. doi:10.1007/s42979-020-00176-1
- Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science*. PhD thesis, Harvard University. Retrieved from https://www.researchgate.net/publication/279233597_Beyond_Regression_New_Tools_for_Prediction_and_Analysis_in_the_Behavioral_Science_Thesis_Ph_D_Appl_Math_Harvard_University
- Werbos, P. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78, pp. 1550 - 1560. IEEE. doi:10.1109/5.58337
- Wondwossen, M. (2004). *OCR For Special Type Of Handwritten Amharic Text ("Yekum Tsifet"): Neural Network Approach*. MSc. thesis, Addis Ababa University.
- Worku, A. (1997). *The Application of OCR Techniques to the Amharic Script*. MSc. thesis, Addis Ababa University.
- Worku, A., & Fuchs, S. (2003). Handwritten Amharic Bank Check Recognition Using Hidden Markov Random Field. *Computer Vision and Pattern Recognition Workshop (CVPRW'03)*. doi:10.1109/CVPRW.2003.10027
- Yaregal, A. (2002). *Optical Character Recognition of Amharic Text: An Integerated Approach*. MSc. thesis, Addis Ababa University.

Yaregal, A., & Bigun, J. (2006). Ethiopic Character Recognition Using Direction Field Tensor. *18th International Conference on Pattern Recognition (ICPR'06)*. Hong Kong, China: IEEE. Retrieved from <https://doi.org/10.1109/ICPR.2006.507>

Yaregal, A., & Bigun, J. (2007a). A Neural Network Approach for Multifont and Size-Independent Recognition of Ethiopic Characters. *Proc. of the IWAPR2007*. Springer.

Yaregal, A., & Bigun, J. (2007b). A Hybrid System for Robust Recognition of Ethiopic Script. *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*. IEEE.

Yaregal, A., & Bigun, J. (2008). Writer-independent Offline Recognition of Handwritten Ethiopic Characters. *ICFHR/2008*. Retrieved from <http://www.cenparmi.concordia.ca/ICFHR2008/Proceedings/papers/cr1160.pdf>

Yaregal, A., & Bigun, J. (2009). HMM-Based Handwritten Amharic Word Recognition with Feature Concatenation. *2009 10th International Conference on Document Analysis and Recognition*. Barcelona, Spain: IEEE. doi:10.1109/ICDAR.2009.50

ተግባሩ አዳኒ. (2008 ዓ.ም). የግእዝ ቋንቋ ትምህርት በአራቱ አድባራት ወገኖች. *ሐቃፊ አገር ካልላይ ጉባኤ ግእዝ ጥንቅር*.

አቤሴሎም ነቃጥቡ. (2012 ዓ.ም). መሰረታዊ ግእዝ፣ የግእዝ ቋንቋ ማስተማሪያ ጽሑፍ (ያልታተመ ጥራት). ጎንደር ዩኒቨርሲቲ.

አዲሴ ያለው. (2012 ዓ.ም). መሰረታዊ ግእዝ (ለማስተማሪያ የተዘጋጀ ሞጁል). ጎንደር ዩኒቨርሲቲ.

ዳንኤል ክብረት. (2008 ዓ.ም). ጥንታውያን የብራና መጻሕፍት እንደ ሰነዶች ምዝገባና ማረጋገጫ ጽ/ቤት. *ሐቃፊ አገር ካልላይ ጉባኤ ግእዝ ጥንቅር*.