DSpace Institution

DSpace Repository **http://dspace.org**

Computer Science thesis

2021-02-02

OFFLINE HANDWRITTEN AWNGI CHARACTER RECOGNITION USING DEEP LEARNING TECHNIQUE

HAILEYSUS, ABATIE

http://ir.bdu.edu.et/handle/123456789/12398

Downloaded from DSpace Repository, DSpace Institution's institutional repository

Wisdom at the source of the Blue Nile

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING AND GRADUATE STUDIES

FACULTY OF COMPUTING

OFFLINE HANDWRITTEN AWNGI CHARACTER RECOGNITION USING DEEP LEARNING TECHNIQUE

BY

Name: HAILEYSUS ABATIE YISMAW

Bahirdar, Ethiopia

February 2, 2021

OFFLINE HANDWRITTEN AWNGI CHARACTER RECOGNITION USING DEEP LEARNING TECHNIQUE

BY

HAILEYSUS ABATIE

A THESIS SUBMITTED TO THE SCHOOL OF RESEARCH AND GRADUATE STUDIES OF BAHIR DAR INSTITUTE OF TECHNOLOGY, BDU IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE IN SOFTWARE ENGINEERING IN THE FACULTY OF COMPUTING

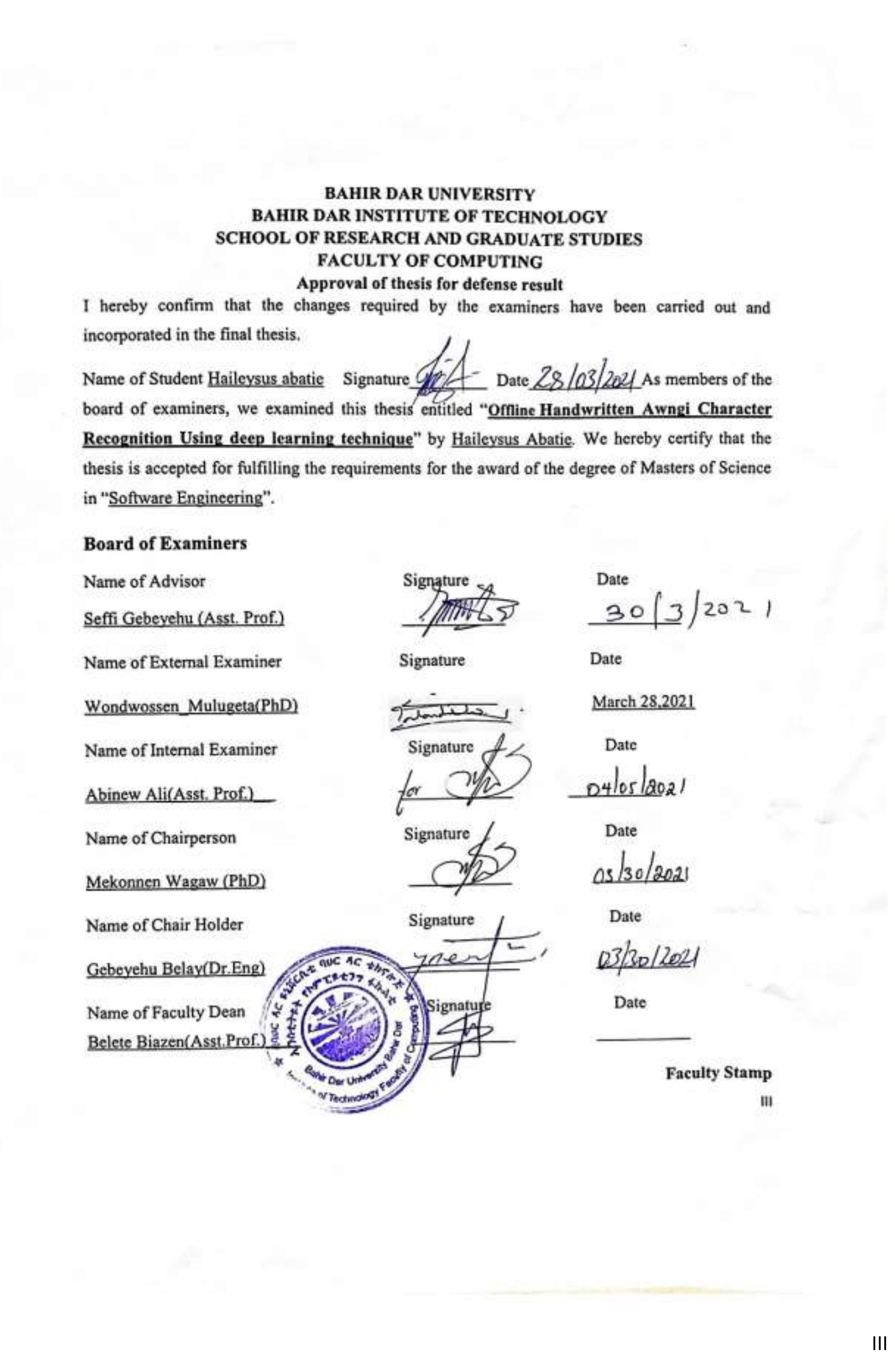
Bahir Dar, Ethiopia

February 2, 2021

Declaration

This is to certify that the thesis entitled **"Offline Handwritten Awngi Character Recognition Using deep learning technique"**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering under Computing, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during the course of this investigation have been duly acknowledged.

Haileysus abatie		28/03/2021
Name of the candidate	signature	Date



BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF RESEARCH AND GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student Haileysus abatie Signature Date 28/03/2021 As members of the board of examiners, we examined this thesis entitled "**Offline Handwritten Awngi Character Recognition Using deep learning technique**" by Haileysus Abatie. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science in "Software Engineering".

Board of Examiners

Name of Advisor — Signature — Date
Seffi Gebeyehu (Asst. Prof.) — 30/3/2021

Name of External Examiner — Signature — Date
Wondwossen Mulugeta(PhD) — March 28,2021

Name of Internal Examiner — Signature — Date
Abinew Ali(Asst. Prof.) — for — 04/05/2021

Name of Chairperson — Signature — Date
Mekonnen Wagaw (PhD) — 03/30/2021

Name of Chair Holder — Signature — Date
Gebeyehu Belay(Dr.Eng) — 03/30/2021

Name of Faculty Dean — Signature — Date
Belete Biazen(Asst.Prof.)

Bahir Dar University Bahir Dar Institute of Technology Faculty of Computing

Faculty Stamp

© 2021

HAILEYSUS ABATIE

ALL RIGHTS RESERVED

Acknowledgments

First of all, I would like to thank the Almighty God, for He is the Source of Wisdom and the Beginning of Knowledge. I praise and thank Him for His great help throughout this work. I would also thank the Ever-Virgin, St. Marry, Mother of our Lord.

My deepest gratitude is to my advisor, **Mr. Seffi Gebeyehu**(Ass. prof). I am deeply thankful to him for giving his insights and constructive comments. My sincere thanks also go to **Mr. Abraham Debasu(Ass. prof)** as well for his continuous support, motivation, and immense knowledge.

My deepest gratitude also goes to **Dr. Milion Mengesha and Abraham Debasu**(Ass. prof) for their guidance at the time of problem formulation and proposal writing.

I am deeply grateful for carefully reading and commenting on revisions of this document. I would also thank **Dr. Gebeyehu Belay**, for his teaching during Research Design and Methodology course.

Last, but not least, I would like to offer my regards and blessings to all my families, friends, and those who supported me in any respect during the completion of the thesis

Abstract

Handwritten recognition systems take input from the paper documents scanner, images; touch screen, and, electronic pen. After processing it offers the output as a electronic document suitable for subsequent access and manipulation. There are a lot of handwritten documents. Which are damaged due to dust, flooding, fire, and natural disaster like an earthquake. Nowadays there are a lot of documents putting on shelves for many years and it takes more space it is bulky to manage and edit a majority of data available in a handwritten form. This requires an approach to change the documents into electronic form for easy of searching and retrieval as per users' need.

In this research work, we have presented the development of offline handwritten character recognition model for Awngi documents. We focus on the 26 base characters Awngi language. The proposed character recognition model contains all of the essential steps that are obligatory for developing an efficient recognition system. The designed model includes modules like preprocessing, segmentation, feature extraction, and classification. In the preprocessing stage, image resizes grayscale, noise reduction, morphological transformation, and binarization. In any handwritten character recognition system is separating individual characters from the document. Better character segmentation phase, has achieved using dual thresholding criterion to minimize the character segmentation error. A CNN is used for feature extraction character classification purposes.

Furthermore, In this paper, we prepared a new public image dataset for Awngi handwritten characters a dataset containing a total of 30,115 out of the total dataset images,80%of dataset is used for training, 20% dataset is used for testing. We collected data from injibara teacher's education, injibara baunk primary school students and Amhara mass media from Awngi media. We are interested in the new success of end-to-end learning in pattern recognition we propose a new trained end-to-end fashion model. The experimental results on handwritten Awngi character recognition that our model achieves a training accuracy of 96.6% and 92.6% of the testing accuracy of the proposed model.

Keywords -- Convolutional neural network, Handwritten, Character, Recognition, end-to-end learning, Awngi

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Algorithms

LIST OF ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
CNN	Convolutional neural network
CPU	Central processing unit
GB	Gigabyte
HOG	histogram of oriented gradients
MSE	Mean Square Error
OCR	optical character recognition
RGB	Red Green Blue
SVM	Support vector machine
PCA	Principal Component Analysis
RMSE	Root mean square error
ReLU	Rectified linear unit
HWR	Handwritten recognition
AHCD	Awngi handwritten character dataset

CHAPTER ONE

1.1. Introduction

Handwritten character recognition is the most challenging and attractive research area in the pattern recognition and computer vision. Due to the serious factors of variances in writing patterns and cursive text, and the similarity of numerous characters in the form, recognition research is laborious and challenging. Handwriting recognition systems take input from the paper, images, touch screen, documents scanner and, electronic pen. After processing it offers the output as a electronic document suitable for subsequent access and manipulation.

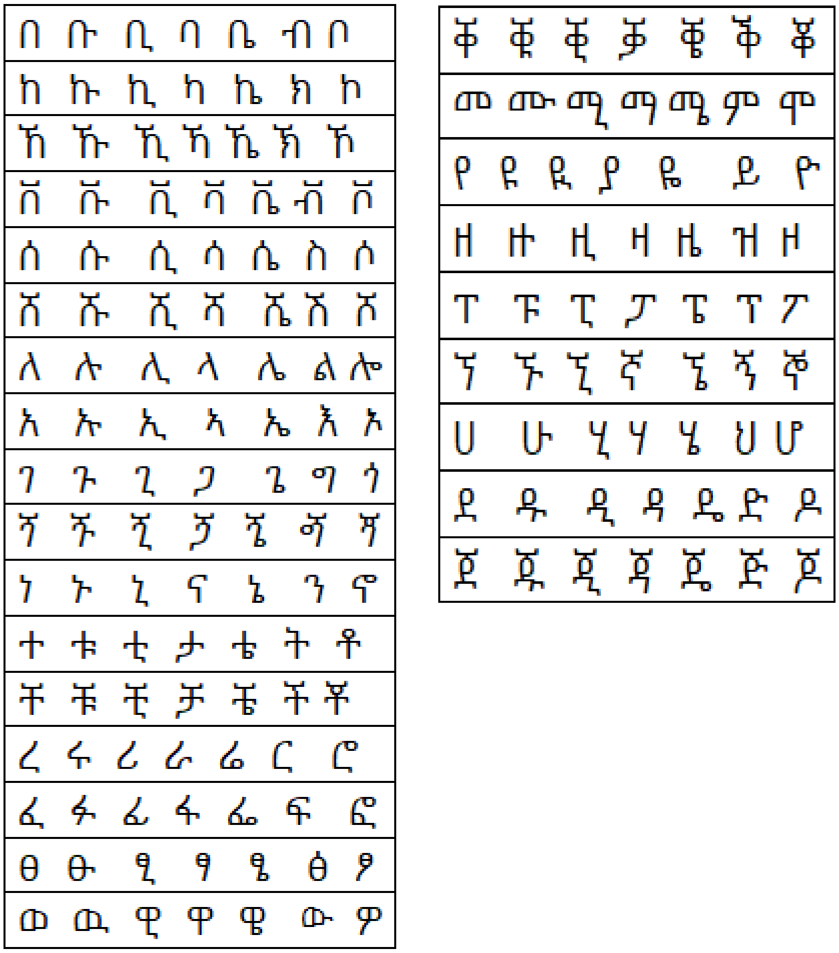
There is a large amount of information being produced every single bit of our lifetime, and most of these are in printed as well as handwritten format. Reaching the level of making these information items accessible and computable is one wing of information science research. This wing is called Optical Character Recognition (Birhanu, Sethuraman, & Research). Optical Character Recognition (Birhanu et al.) Is a process that allows the handwritten document to be recognized optically and converted into a digital format that can be accepted by a computer for further processing. Nowadays Optical Character Recognition (Birhanu et al.) The authors only considered 231 basic Amharic characters. The most important technology can able to convert a handwritten document or handwritten into computer format by capturing the image of the documents. Nowadays, Americans, Indians, Europeans, Chinese, Arabic researchers have implemented Optical Character Recognition (Birhanu et al.) Technologies for their languages.

As a result, these OCR technologies help to read different documents written in English, Chinese, Hindu, Arabic, Russian, and the like but do not read documents written in Awngi. In Ethiopia, there are over 80 languages spoken all 80 languages have their writing and lettering style so Awngi is one of them, which is spoken in the Amhara region which is found in the Awi zone around Injibara Ethiopia. researchers have implemented Optical Character Recognition (Birhanu et al.) Technologies for their languages. As a result, these OCR technologies aid to read different documents written in different languages.

Nowadays there is a huge amount of information being produced every single bit of our lifetime, and most of these are in handwritten as well as printed format large amount of information is found in churches, governmental and nongovernmental and private institutions including information centers, libraries, museums, etc. Thus, they can take a share of these advantages by developing the Awngi OCR system, as the country is endowed with countless historical, cultural, and other documents written using Awngi characters.

In the market many automatic systems of handwriting recognition are available. Yet, these systems provide solutions mainly for major world Languages scripts such as Arabic, Chinese, Japanese, and English.in Addison, that the recognition problems of the scripts are not solved. There are numerous characters used in varied parts of the sphere for which there is not found any automated model able to recognizing handwritten character images. Due to the huge difference in handwriting styles, the handwriting recognition state-of-the-art technologies fail to deliver satisfactory performance on different types of handwriting systems. In the recent past, handwriting recognition technologies have advanced fast and significant developments have been accomplished by using different algorithms, such as Support Vector Machines (Fitsum & Patel), Convolutional Neural Networks(CNN), and Artificial Neural Networks(Arcelli, Di Baja, & Intelligence).

There is a vast of information available in handwritten documents that essential to be converted into electronic form for easy and better searching and retrieval. the massive amount of documents found in museums, libraries, government, and private organizational offices is found in the form of magazines, letters, pamphlets, books, newspapers, etc. That Converting these documents into electronic format is a must to protect historical documents, save storage space, and make them available via the Internet or to share over the internet .some applications of OCR include a governmental and private office, bank check processing, Postal automation, and libraries and publisher. Utmost different handwritten recognition models are motivated by recognition numbers as compared with the alphabetic characters. Recognizing handwritings is an attractive a vital part of many applications peoples are regularly exposed to and use technology, such as tablet devices, smartphones, both for educational and entertaining purposes. Handwriting using the stylus or a finger is attractive one of the favorite user input choices. To the best of ours understanding



በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጘ	ጙ	ጚ	ጛ	ጜ	ጝ	ጞ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ

ቐ	ቑ	ቒ	ቓ	ቔ	ቕ	ቖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ

Figure 1. 1 Awngi alphabet (MAND) with their seven order row

Special characters ጡ ጤ ጣ ጠ

1.2. Problem statement

There are a lot of handwritten documents in Primary school, governmental institute, non-governmental institute, museum, and personal files. Those documents are damaged due to dust, flooding, fire, and natural disaster like an earthquake.

Nowadays there are a lot of documents putting on shelves for many years and it takes more space it is bulky to search and edit a majority of data available in a handwritten form that requires to be changed into electronic form for easy searching and retrieval as per users' need.

Due to the reasons mentioned earlier, converting handwritten Awngi documents into electronic format is needed. To convert the text of these documents the conventional way is typing through the keyboard, which is not only time-consuming, error-prone, and tedious. The problem of typing into computers is even worse for Awngi characters where typing each character needs two keystrokes on average. This emphasizes the importance and tremendous need for Awngi handwritten OCR that is capable of recognizing characters. Thus, if the automation of documents is needed the OCR software is the preferred means for converting existing documents into machine-readable form. There are Awngi characters ቐ("ቐ","ቑ","ቒ"ቓ","ቔ","ቕ","ቖ“,ቚ,ቛ,ቜ,ቝ)ጘ("ጘ","ጙ","ጚ","ጛ","ጜ","ጝ","ጞ"ጟ,),that differ from Amharic characters ቐ and ጘ with their variants are frequently typed character in awigni writing system, that need to be recognized by Optical character Optical character recognition is an essential part of document scanners. OCR is used in different applications such as banking, script recognition, postal processing, security, and language identification. Many researchers are done on Amharic and Geez OCR in both printed and handwritten but it needs improvement of the recognition performance.

The cost of accessing, cost of storage, cost of searching, and cost of converting handwritten documents are the main problems. Retyping handwritten documents is tedious, boring, consuming time, and unmanageable. And also, there are practical challenges such as tax form processing, bank check reading, book.

1.3.Research Questions

- Which optimization technique is appropriate for offline Awngi character recognition?
- What preprocessing methods are suitable for Awigni characters?

1.4.The objective of the study

The objective of this study can be described into two categories: General objective and specific objective. Both categories are described as follows.

1.4.1. General objective

The general objective of the study is to develop offline handwritten Awngi character recognition model.

1.4.2. Specific objectives

To achieve the general objectives of the study, the following specific objectives are:

- To identify optimization technique for Awngi character recognizer
- To prepare a training and testing dataset for Awngi character recognition systems
- To evaluate the performance of our model

1.5.Scope of the study

The scope of the study is to design and develop an isolated handwritten character recognition system for Awngi basic characters. Since the recognition of isolated handwritten characters is our aim and it is one of the challenges in the OCR system. Because isolated characters are the building block of words, sentences, and paragraphs. We are going to design medium size new data set for Awngi handwritten characters.

Because of the time, we are limited to develop only basic characters.

- Awngi digits left as future work.

1.6. Methods

In this research work, from different research design's we follow design science research methodology. Following is the description of each phase of design science research methodology: motivation and problem identification, solution for objectives, design and development, evaluation.

1.6.1. Motivation and problem identification

Some factors motivate us to undertake a study. First, there are a lot of handwritten recognition researches in other languages such as English, and non-English languages, however, there is no researches have been done in only Awngi characters and,

- A majority of handwritten documents available that requires to be altered into form of electronic for easy retrieval and searching as per user need.
- To prevent different kinds of damage like fire, flooding, dust, etc.
- There are a lot of large documents written in Awngi language like historical books, letters, etc but there are no OCR technologies to convert to computer format (digital format)
- Converting handwritten documents used to reduce searching time, cost of storage, retyping of documents, and cost of accessing the written document.
- Nowadays deep learning is the most attractive and better approach, so we are motivated to apply the deep learning approach to Awngi handwritten characters.

In this stage, the thesis problem is well-defined and the importance of a solution is justified. Problem definition is used to develop an artifact that provides a solution. Explaining the value of a solution aids us to motivate the researcher and understand the thought associated with the researcher's understanding of the problem (Ken Peffers,, Tuure Tuunanen, & , Marcus A. Rothenber, 2007) . Different kinds of literature are reviewed to obtain knowledge about the state of the problem and the importance of the solution. The gaps in related research works are analyzed and how we fill in the gaps is presented.

1.6.2. Solutions for Objectives

The objectives of the study that are inferred from the problem identification are explained. Various resources have been reviewed to know the state of the problem, the state of current solutions, and their efficacy.

1.6.3. Design and development

In this phase, the art factual solution is created for the problem stated this task includes defining the artifact's preferred functionality and its architecture and then creating the actual artifact or model. Keras is used for designing the CNN model Keras uses Tensor Flow as a backend. Python programing language is used for writing the required source codes.

We collected handwritten character images from different participants in collages with different handwriting styles handwritten Awngi character image datasets are divided into two parts: training dataset used to train the model and to increase the performance of the model through different parameters; testing dataset to evaluate the model. We are collected About 36,000 handwritten characters were collected. We divided it into 80/20 format 80% used for training and 20% used for testing the trained model.

1.6.4. Evaluation

The developed model is evaluated to measure how well it supports a solution to the problem. To evaluate the model, testing datasets were fed into the developed model. the model was evaluated by comparing its output against the observed data using precision, recall, and f1-score values for evaluating recognition accuracy.

1.6.5. Communication

In this section, the problems, the artifacts of the designed solution, the effectiveness and other

Related information is communicated to relevant audiences when appropriate.

1.7.Significance of the study

Introducing this paradigm to solve this challenging problem would have a significant impact in terms of technology transfer. The outcome of this research can inspire other researchers to try different algorithms to solve these challenging problems which can improve the quality of day-to-day life. The research would show the role of modern artificial intelligence in solving challenging visual recognition tasks. Likely applications in support technology for blind and visually impaired users, human-robot interaction, automatic data entry for business documents Traffic number plate recognition, Postal code identification, Reading of bank cheques, Extracting data from hardcopy forms, and Digitization of documents ID card reading

1.8.Organization of the Thesis

This section presents the overview contents of the remaining chapters. The rest of this thesis is organized as follows

In Chapter Two, the literature reviewed on the concept of handwritten and printed character recognition is presented. Besides, we have given a detailed description of Awngi language characteristics a detailed analysis of different works related to Handwritten character recognition is presented. Only those works whose contributions are related to our work are reviewed. Also, the common gaps of the reviewed works and the way how we fill in the gaps are described exhaustively.

In Chapter Three methodology materials are going to be presented also the components that compose the system (preprocessing, segmentation, feature extraction, feature learning, and classification using CNN), and the responsibility of each component is described in detail. , and dataset used and the implementation of the current models is described thoroughly.

In Chapter Four experimental result discussion of the proposed model for Handwritten Awngi, character recognition is described in detail. Finally, the test results and discussion is presented.

In Chapter Five, we would conclude the major findings in this research work. Also, the major Contributions of the proposed CNN model and future works would be outlined.

CHAPTER TWO

2. LITERATURE REVIEW

2.1.Introduction

In this chapter, a thorough review of the literature and analysis of related works are presented. Handwritten character recognition (HCR) takes an active and interesting area for researchers. Many research papers, published journals, and other studies have been done by researchers to improve the efficiency and accuracy of the recognition process. In Handwritten character recognition many relevant phases or stages like data acquisition, preprocessing, classification. And each phase has its objectives; their efficiency outlines the accuracy of the complete recognition process. As we have tried to explain before those processes need to be improved to come up with a better and efficient character recognition system.

Many researchers have tried to improve each of those phases at different times, and still many papers are being published presenting new and efficient ideas about preprocessing post-processing, and other relevant techniques. However, most of the researches which is done on the recognition process are especially it is difficult to find enough researches on Awngi or Amharic character recognition for both handwritten or printed document recognition process in deep learning. We can say that it's almost impossible to find researchers done on any processes of character recognition for Awngi scripts or manuscripts.

The overall process of handwritten character recognition is much difficult than printed character recognition which is applied on scanned printed documents. There is a different variant of the handwritten styles among the different people. Many people have their writing style even it's difficult for a person to use the same handwriting style every time he/she writes. Also, poor and cursive handwriting is difficult to recognize, this makes the segmentation and classification process difficult. If the paper is written in cursive handwritten style then the process would not be handwritten character recognition (HCR) most likely it would be handwritten word recognition (HWR). This and many other reasons make the handwritten character recognition process especially the segmentation phase much harder than printed document recognition. Phases like line segmentation, word segmentation, and character segmentation are the most relevant and advanced phases of any character recognition system, especially for handwritten character recognition. As we have discussed it HWR is difficult than printed document recognition. Moreover, early document character recognition is much difficult and advanced than even the handwritten character recognition process.

Early documents are very noisy so, they require an advanced noise removal technique. Since the noise removal method affects the overall process of the recognition system, it needs to be done carefully. The background of most of the early documents is not white; we can even say they don't have a common background color. As a result, the noise removal process would be very difficult and if this phase cannot be achieved in a quality manner then the other phases would be much more difficult than expected. Most of the ancient documents are aged beyond expectation. As a result, the letters, words, and also statements are not fully clear even for human eyes.

There are many different languages and documents which are written using their languages. Nowadays, those documents are being converted into machine-editable texts by many researchers**.** As we observed from different papers, some of the studies used a deep learning approach. The deep learning approach is the modern methodology used to extract features automatically. However, deep learning needs a huge amount of data set for training and testing the system. The deep learning approach produces a promising result in the accuracy of recognizing characters. Some of the algorithm, such as convolutional neural network, deep belief neural network, and other. The other approach is using the different machine learning techniques, first, extract features of the character and finally fed extracted features into the classification algorithm. There is human intervention. The second approach doesn't require huge data set than the first approach. The deep learning approach is difficult to train than another approach.

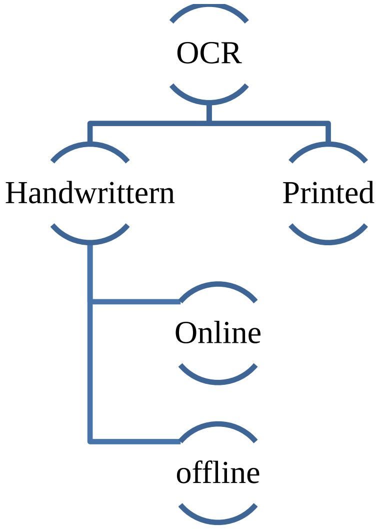


Figure2. 1 Review of optical character recognition

In this chapter, a thorough review of works of literature and analysis of related works are presented. We reviewed literature related to this study and understanding the basic concepts of optical character recognition; especially we are focused on handwritten character recognition. We reviewed the paper in two different ways, such as based on different language and different approaches in which the author employed. Review literature is one of the methodologies to gather the required information for this study.

2.2.Handwritten Recognition

(Reta, Rana, & Bhalerao, 2018). the Author presented "Amharic handwritten character recognition using combined features and support vector machine," The algorithm is trained and tested on Amharic handwritten characters data set and Chars74K benchmark numeric data set. The model was validated using a 10-fold cross-validation technique. From the experiment, they observed that some of the characters were recognized correctly. With the limitation characters which have a visual similarity of shape are still challenges and also the accuracy of recognition can be improved using different feature extraction techniques.

(B. Belay, Habtegebrial, Liwicki, Belay, & Stricker, 2019)They briefly explain how can recognize Amharic character from image data the algorithm is trained and tested on data set of Amharic image characters The proposed method achieves 94.97% overall general character recognition accuracy introduced convolutional neural-based technique, for Amharic printed character image recognition. The architecture consists of two classifiers: the first is responsible to detect a row component and then the second is responsible to detect the column component of a character in 'Fidel Gebeta' the performance of the proposed framework.

In the research work of (Demilew & Sekeroglu, 2019) the researchers conducted ancient Ethiopic manuscript recognition based on a deep network. The proposed system includes all of the necessary steps that are useful for the recognition system to be as efficient as possible. After the neural network is designed it's trained based on an unsupervised learning algorithm of a greedy layer. The model first converts input images into a grayscale image. RGB images are stored as three-dimensional matrices, but grayscaled images are stored as two-dimensional matrices making the recognition process much easier. As a result of the conversion of the RBG images into grayscale images, it saves memory usage and the processing time required to perform any kinds of operations on the images. Then the grayscale images are binarized using a hybrid binarizing technique which is derived from the two most common methods which are global and local thresholding. Then, the binarization is followed by skew correction processes. Also, the segmentation process is divided into two sections which are the line and character segmentation. For the process of the line and character segmentation, the researchers used horizontal and vertical projections respectively. the researchers prepared a dataset for training and testing the proposed system The dataset prepared contains 2400 they take only 24 base characters of the Geez alphabet and each base alphabet has a repetition of 100 images. The characters were collected from a total of 200 pages. The dataset contains a total of 7065 characters which are normalized into a scale of 28*28 pixels. Among the total of 7065 characters, 70% of the dataset was used for training and the rest were used for testing purposes. The researchers designed three kinds of DNNs in which the first model contains 2 hidden layers. Finally, for the last model having four hidden layers having 28 units and with 150 epochs the classification error 0.077778 which was not as expected. Therefore, after analyzing the result obtained in research they conclude that a deep neural network approach with three hidden layers shows a better performance for ancient Ge'ez recognition, according to the proposed research.

Also, in research (Birhanu et al., 2015) the researchers conducted Amharic character recognition for real-time document recognition and they proposed an artificial neural network. The proposed system follows the three most common parts which are preprocessing, segmentation and classification. Even though they have shadowed a standard and advanced character recognition process their result is not that much satisfying. After the proposed neural network is trained with the prepared dataset, the classification accuracy for the training image was an average of 96.87%. However, the accuracy of the system for additional testing images was 11.40% which is not satisfying and expected results.

In the paper Recognition of printed Amharic documents,(Meshesha & Jawahar, 2005) authors used multiclass DAGSVM for recognizing a printed Amharic character (Ethiopian script). They did feature extraction from the whole image to create a single contiguous vector. They concatenating all rows of the printed images. The extracted feature consists of 0s for representing background and 1s for representing foreground pixels of the character image. For dimensional reduction of features, Principal Component Analysis (PCA) followed by Linear Discriminant Analysis (LDA) have been used. Finally, Multiclass DAGSVM is applied to the given feature.

(Fitsum & Patel, 2018)In research, optical character recognition for Tigrigna printed documents researchers are present an Optical Character Recognition (OCR) to convert printed documents into computer format using SVM classifiers. They convert input image into Grayscale and binary (black and white) and then removes noise and skew detection and correction. Segmentation is the most significant matter to segment the character by word, line, and character. They have used HOG (histogram of the oriented gradient) as feature extraction to extract information from the character. Researchers are using techniques to extract the features of each segmented images and SVM classifier used to classify based on the feature extraction similarity of the shape of Tigrigna characters was a challenge for them to recognize printed Tigrigna characters they used noise detection and removal techniques for the recognition of real-life documents did not produce satisfactory result Skewed and slant correction for the problem domain in this work was done using Microsoft paint rotation toolbox.

Abdurahman, F., (2019). Authors worked on the title of “Handwritten Amharic Character Recognition System Using Convolutional Neural Networks” CNN feature extraction and classification technique is proposed result of the proposed techniques CNN model gives an accuracy of 91.83% on training data and 90.47% on validation data special characters of

Amharic letters are put as future work The proposed system requires large data set, and it consumes time and Increasing performance visual similarity of shape are still challenges Also the accuracy of recognition can be improved using different feature extraction techniques
The authors performed preprocessing of character images such as binarization, slant correction, smoothing, and noise removal, and normalization. In feature extraction, the authors used four sets of feature extraction techniques such as using boxing approach

In research (Pal, Wakabayashi, & Kimura, 2007)a handwritten Bangla character recognition was proposed by the researchers. They used bidirectional arc tangents through the application of a feature extraction method on the grayscale images for the recognition method. Researchers have applied a “two by two” mean filtering method and researchers applied this mean filtering technique for four iterations. After the mean filtering is done, they conducted a non-linear normalization of size on the images. Gradient image was relevant at this stage so, to get the gradient image, a Roberts filter was applied to the images. Finally, the quadratic classifier was used by the researchers to classify the characters. After all those processes and methods were applied to the images the result was not that vast and outstanding even if it is great progress. They have accomplished 85.9% percent of accuracy on the Bangla dataset containing 20,543 test samples they have applied a five-fold cross-validation technique

In research (Desai & Singh, 2016) Authors proposed deep learning, training the classifier using a convolutional neural network model. The author applied the center of the loss function to alleviate inter and intraclass differences simultaneously. They combine center loss function and softmax function. Center loss function shows good results. The network is composed of four conventional layers; four max-pooling layers, two fully connected layers, and ReLu. The input image data contains 96x96 sized grayscale images and they have used a dataset with a bulky number of images. They used a deep learning tool called Caffe for train and Stochastic Gradient Descent (SGD) algorithm for fine-tune the network parameters. The proposed model achieved an accuracy rate of 92.75%.

In the research paper (Nair, James, & Saravanan, 2017)authors proposed a deep learning approach, the CNN model. CNN delivers the greatest accuracy rate and wants a large dataset to train the model. The researchers used different techniques to increase the size of the data set. In this system, the authors perform different tasks such as collecting the dataset, enlarge the data set, augmenting data then preprocess the data set, develop the CNN model, and finally classifying the character image using the softmax layer. And, they have used the dropout layer to reduce the over fitting problem.

In research(Acharya, Pant, & Gyawali, 2015) Authors employed a deep learning approach. In this research paper, they have used 92,000 dataset images with different 46 classes of characters. Authors have used dropout and extending the dataset method to produce better accuracy. The authors train deep neural networks on the dataset. In this model, they implemented a Back propagation algorithm using stochastic gradient descent with a momentum of 0.9, the batch size is 200, and the network trained for 50 epochs. The researchers experimented on the two modes; they conclude that increasing the number of training samples is effective to improve performance. Established along with the experimental result, CNN with dropout layer and extending the dataset method can result in high test accuracy. The accuracy of the proposed system is 98.47%

In this paper researchers (Afroge, Ahmed, & Hossain, 2017) presented an optical character recognition approach especially for Bangla Language offline printed characters. It has been studied that there is no specific algorithm is establish for well-organized feature extraction. The Feature selection is very important or an essential step of handwritten character recognition. Accurate and distinguishable feature extraction plays a significant role to influence the performance of a classifier. A feature extraction scheme based on “Dynamic Time wrapping” and “Discrete Frechet Distance” is proposed. Chance of incidence of the pixels on the given character of the dissimilar font is considered to train Multilayer perceptron Neural Network. Investigational results display 100% accuracy for trained characters and overall 90-95% accuracy for all elementary characters of the established method on training and the test datasets respectively. (Elleuch, Tagougui, & Kherallah, 2015) In this paper, the authors employed RBF multiclass SVM supervised classification algorithm and Gabor filter as feature extraction techniques. After preprocessing the dataset, Gabor features are extracted. Gabor flirters are used in image processing for extracting good features such as optimal joint spatial frequency location and fitness to simulate the receptive fields of simple cells in the visual cortex. It represents an image in different levels of frequency. Then different features are extracted, based on everyone’s response to the filter and its frequency. In this study, one versus one multi-class SVM is used for the classification of 66 classes of characters. The experiment was performed on the HACDB database. The model produced an error classification rate of 12.23% with RBF kernel multiclass SVM. And also, the experiment shows that multiclass SVM with RBF kernel produces good results than polynomial SVM on HACDB with 66 classes. (Poornima, Kavitha, & Sornam, 2016) The authors proposed the Morlet activation function-based Adaptive Backpropagation (ABP) model. The authors compared ABP with the BP network model based on the three activation function such as sigmoid, Shannon, and Morlet functions. The proposed method shows Morlet function achieves the best performance than other activation functions. From the experiment, the result shows Mean Square Error (MSE) of Morlet function is 0.00021 with 56 m/s CPU time and 45 epochs. It shows that in the Morlet activation function, the network model performed better. (Desai & Singh, 2016) The authors describe two implementations in character recognition using the template matching method and feature extraction method followed by support vector machine classification. Image pre-processing the image is converting into binary black and white and remove the noise. The image is segmented into sub image characters ,and the associations between single character ,and a given set of patterns or templates are computed to find the similarities and then identify the input character. After the segmentation Features extracted from the segmented sub-image characters are used to train the SVM classifiers.

(Nakkach, Hichri, Haboubi, & Amiri, 2016)This paper presents a new approach for the feature extraction step of online handwritten Arabic characters using global and local features. The proposed system was tested with 2000 Characters written by different writers and the best rate of recognition obtained was 92.43%. The system has used the Fourier Descriptor algorithm for global features extraction and, chain code of freeman for local feature extraction. The achievable aim of future experiments is to add some structural features such as a number of the dot(s) and their positional information, character’s position within a word (first, last, middle, or isolated character) to enrich the database of Online Arabic OCR to improve recognition accuracy.

In the paper Handwritten Gurmukhi character recognition (Aggarwal & Singh, 2015) author describes. The two features of the characters gradient feature vectors and curvature feature vectors are calculated on the character image. to create a single feature vector they fused two features. They have used steps to extract features like the Computation of gradient, computation of curvature, composite feature vector generation and at the end reducing the dimension of the feature using Principal Component Analysis (PCA). They used two feature combination techniques are used: by concatenation and forming composite feature by the cross product they forming composite feature .at the last, the combined feature is used by the classification algorithm. SVM with RBF kernel used as a classifier. The accuracy of the recognition rate is 98.56%

In the paper,(Katiyar & Mehfuz, 2015) proposed SVM-based offline handwritten digit recognition. In this paper, SVM is used as a classifier. The authors performed preprocessing of character images such as binarization, slant correction, smoothing, and noise removal, and normalization. In feature extraction, the authors used four sets of feature extraction techniques such as using boxing approach, using diagonal distance approach, mean and gradient operations. Enormous features are extracted using those techniques. The extracted features are concatenated to form a single feature vector. After obtaining these features, SVM is applied to the final Feature vector. The accuracy of the recognition rate is 97.16%.

In Paper (Bautista et al., 2015) the author proposed handwritten alphanumeric character recognition using a support vector machine and projection histogram. The authors follow basic optical character recognition steps: preprocessing feature extraction, and classification. In preprocessing, basic preprocessing operations are performed such as gray scaling, noise removal, binarization, and others. Similarly, in feature extraction, project histogram is counting pixel distribution in horizontal and vertical directions of character images. after transforming two -dimensional image into a one-dimensional signal via a projection histogram, then Principal Component Analysis (PCA) is used to distinguish which of the histogram are unique for the particular character, and also PCA is used to reduce the dimension of features. After the feature extracted using a projection histogram and features are reduced using PCA. Finally, this reduced feature fed to the classification algorithm. In this paper, SVM is used as a classifier. The different kernels of SVM are used. Hence, the RBF kernel achieved a good result. The authors

(Hussien, Elkhidir, & Elnourani, 2015) employed a special model of neural network called Hopfield neural network. Hopfield neural network is used as a classifier. Hopfield neural network has some advantages over other models. It is simple to implement using optical devices. Hopfield neural network is characterized by inputs are feedback to inputs, no self-feedback, the weight matrix is symmetry and the state is updated one at each time. The Hebbian learning algorithm is used to find the weight. In this study, seven features are extracted from each character, such as the number of connected components, number of holes, number of dots, the position of dots, upper and lower density, aspect ratio, and extrema.

(B. H. Belay, Habtegebrial, & Stricker, 2018) Recognition of character image has been studied and solved for multiple scripts. They have been complete lots of work to solve the problems in most Latin and non-Latin scripts and even most of the scripts, now, have commercially off-the-shelf optical character recognition applications. Yet, there are still untouched scripts in the area of optical character recognition

A paper (Assabie & Bigun, 2009)proposed handwritten character recognition. It is based on structural features of primitive character strokes, and the type of characters categorized based on their silent primitives. The structural and syntactic model grasps the orientation, relative length, structure, and spatial position of primitive strokes by using a directional field tensor. In this method, the authors build a primitive tree to hand the relationship, and the tree traversed to generate sequences of primitives the arrangement of strokes is compared against the knowledge base. The knowledge base stores sequences of primitive strokes and their connections to Ethiopic script.

In the Paper (Bautista et al., 2015), the researchers proposed handwritten alphanumeric character recognition using projection histogram and support vector machine. They have followed basic steps for optical character recognition: like preprocessing feature extraction, and classification. In preprocessing, basic preprocessing operations are done such as gray scaling, noise removal, binarization, and others. Similarly, in feature extraction, project histogram is counting pixel distribution in horizontal and vertical directions of character images. after transforming two -dimensional image into a one-dimensional signal via a projection histogram, then Principal Component Analysis (PCA) is used to distinguish which of the histogram are unique for the particular character, and also PCA is used to reduce the dimension of features. After the feature extracted using a projection histogram and features are reduced using PCA. Finally, this reduced feature fed to the classification algorithm. In this paper, SVM is used as a classifier. The different kernels of SVM are used. Hence, the RBF kernel achieved a good result.

Paper (Kaleka & Verma, 2016)describes Handwritten Gurmukhi character recognition. In this thesis, researchers have calculated two features of characters: curvature feature and gradient feature of the character image. The two features are fused to create a single feature vector. in feature extraction, there are steps: Computation of gradient, computation of curvature, composite

feature vector generation, and at the end reducing the dimension of the feature used Principal Component Analysis(PCA). There are two feature fusion techniques are used to fuse the two features: forming composite feature by simple concatenation and forming composite feature by cross product. Finally, the combined feature is used by the classification algorithm. SVM with RBF kernel used as a classifier. The accuracy of the recognition rate is 98.56%.

In the paper(Katiyar & Mehfuz, 2015) proposed SVM-based offline handwritten digit recognition. In this paper, SVM is used as a classifier. The authors performed preprocessing of character images such as binarization, slant correction, smoothing, and noise removal, and normalization. In feature extraction, the authors used four sets of feature extraction techniques such as using boxing approach, using diagonal distance approach, mean and gradient operations. Enormous features are extracted using those techniques. The extracted features are concatenated to form a single feature vector. After obtaining these features, SVM is applied to the final feature vector. The accuracy of the recognition rate is 97.16%.

In the research paper(Retsinas, Gatos, Stamatopoulos, & Louloudis, 2015)they have used feature extraction techniques of projection-oriented gradients followed by a support vector machine classifier. The proposed model by researchers evaluated data sets GRPOLY-DB, CIL-Greek, and CEDAR-English. Projection of oriented gradients consists of two steps:

computation of oriented gradients & got a projection-based description for each character gradients. Gradients orientation elaborate computation of gradients of the pixel in the character image along horizontal and vertical direction using different kernels. The projection-based descriptor is used to divide the binary character image into different projections on the selected angles, by copying the Radon transform. Fourier transform computed to decrease needless variations.

2.3.Characteristics of Awingi Alphabet

Awngi language is one of the four (Xamtanga, Bilen, Kemant, and Awngi) central Cushitic Agew language groups. Among these Xamtanga is spoken by peoples which are live in waghimera, Kemant is spoken by peoples which are live in Northwestern Ethiopia (Gondar), Bilen is spoken by peoples which are live in Eritrea, and Awngi is spoken by the bulk of speakers who are located in the Agew Awi zone of Amhara Regional state, and in the Metekel zone of the Benishangul –Gumuz Regional State. But the language which is spoken in Benishangul –Gumuz and Northwestern Ethiopia (Gondar) is different from the language which is spoken in the Awi zone (Yaregal, 2005) The use of a large number of scripts in writing and the presence of a large number of visually similar character or scripts are the difficulties of research in the identification of the Ethiopian indigenous scripts.

It is understood that Awngi alphabets are derived from Geez, but Awngi has its special character that differs from Amharic Although, (Yaregal, 2005) Awngi alphabets contain 26 base characters with each base character having its own different 7 kinds of shapes. Some experts agree about Awngi letters which are not a part of Amharic alphabets ቐ ፣ ኸ and ጘ each base character having their own different 7 kinds of shapes but character ኸ is present in Amharic alphabet. so character **ቐ("ቐ"," ቑ ","ቒ "ቓ", "ቔ" ,"ቕ", "ቖ") ጘ("ጘ","ጙ","ጚ","ጛ","ጜ","ጝ","ጞ",)**, **ኸ ("ኸ","ኹ","ኺ","ኻ","ኼ","ኽ","ኾ",)** are frequently typed character in Awngi writing system.

2.4.Summary

In this chapter, we have reviewed the literature on the handwritten and printed character recognition methods for Amharic, Geez, and other languages. We have discussed researches that propose different kinds of recognition methods. Such as Hidden Markov Model, Convolutional neural network, deep neural network, and other models. Also, we have understood numerous kinds of papers that are done on Bangla handwritten character recognition, Handwritten Latin Character recognition, Amharic handwritten character recognition, and Amharic handwritten word recognition. Some scholars propose binarization free technique results in efficient recognition accuracy. There are also other research works suggesting that binarization makes the recognition of ancient documents as efficient as possible as that of the binarization free approach. Furthermore, we give a detailed analysis and characteristics of the origin language that differs from Amharic. Finally, we have also reviewed and presented different CNN architectures, with results in pattern or image recognition.

Table 2. 1 Summary of related work

Author (s)	Title	Technique	Result	Limitation and future work
(Reta, Rana, & Bhalerao, 2018).	Amharic handwritten character recognition using combined features and support vector machine	features are extracted using Haar wavelet transform followed by HOG support vector machine as a classifier	The result of the paper shows performance of the classifier achieved an accuracy of recognition 86.30% on AHCD data set	✓ They didn't have worked in all Amharic alphabets or "Fidel" Amharic characters which have a ✓ the visual similarity of shape are still challenges ✓ Also, the accuracy of recognition can be improved using different feature extraction techniques.
Abdurahman, F., (2019).	Handwritten Amharic Character Recognition System Using Convolutional Neural Networks,	CNN feature extraction and classification technique is Proposed.	The result of the proposed techniques CNN model gives an accuracy of 91.83% on training data and 90.47% on validation data	Special characters of Amharic letters are put as future work The proposed the system requires large data set, and it consumes time and Increasing performance ✓ the visual similarity of shape are still challenges ✓ Also, the accuracy of

				recognition can be improved using different feature extraction techniques
(B. Belay, Habtegebrial, Liwicki, Belay, & Stricker, 2019)	Amharic Character Image Recognition	The architecture consists of two classifiers: the first is responsible to detect a row component and then the second is responsible to detect the column component of a character in ’Fidel Gebeta’	The proposed method achieves 94.97% overall general character recognition accuracy	✓ Yet, there are still untouched scripts or character in the area of optical character recognition ✓ Visually similar characters are a challenge even when the character is printed And also the accuracy of recognition needs to be improved
(Demilew & Sekeroglu, 2019)	Ancient Geez manuscript recognition based on a deep network	They have proposed deep neural network DNNs in which the first model contains 2 hidden layers. Finally, for the last model having four hidden layers having 24 units and with	The result of the paper shows performance of the classifier achieved an accuracy of recognition 90.30% on Geez character datasets	✓ As a limitation, they have only taken only 24 base characters of the Geez alphabet ✓ still untouched scripts or character in the area of optical character recognition ✓ the accuracy of recognition needs to be improved

CHAPTER THREE

3. METHODOLOGY

3.1. Research Methodology Overview

A general handwritten character recognition application starts from dataset preparation and is followed by a training model. In this unit, we are present the dataset that is used for training and model evaluation, the proposed model architecture, and the training schemes In this thesis we have divided the research methodologies into two groups which are used while directing the design and implementation of the proposed system. We have used for the proposed system are grouped as data collection and system development methodologies. In phase of system development we have used python programing language used for developing the proposed system. Some of the frequently used libraries used during the employment of the proposed system are Tensor Flow for normalizing image vectors and as a supporting library for Keras. Keras**:** For implanting the CNN architecture in a much easier way. Open CV**:** For preprocessing and other relevant tasks. NumPy**:** For converting the images into integers and storing them as a 2D vector.

3.2. Data collection methodologies

Datasets are an essential part of any computer vision system. A standard dataset permits researchers to equate different machine learning approaches and methods quickly and properly(Ciresan, Meier, Gambardella, & Schmidhuber, 2011). In handwritten character recognition, the primary task is collecting required data and preparing it for further processing. Awngi handwritten character benchmark dataset is still not available for the research community in public and this was the major challenge during this research work.

To collect the data from peoples, white A4 paper was used for data collection, and for the ancient documents we have scanned 30 pages of a document. The awngi letter contains 26 unique characters; we were asked injibara teachers collage Awngi department students to fill the empty paper in Figure 3.1 following the order shown in the reference matrix. After making an agreement from the school participating, teachers were provided with copies of the data collection papers that were spread among school students, and the reference data collection matrix. We prepared 170 data collection A4 papers and 155 data collection papers were obtained from participants 155 datasets for each character were collected and having (155x182) 28,212 datasets in total.

We have used A4 paper as a data collection paper we were collected from 155 persons who write each character in the A4 paper data collection. The selected individuals can write the Awngi alphabet randomly with different educational backgrounds and different age ranges. Sample filled character in A4 papers following 155 obtained matrices was scanned with a resolution of 332 dpi Samsung galaxy S7 CamScanner mobile application. After scanning the image the scanned image before the image is segmented will be prepossessed, since the research is dealing with pattern recognition, or more specifically character recognition, the data collected are of two types. The first being data for training the convolutional neural network (recognition engine), the second will be for testing the performance of the CNN model.

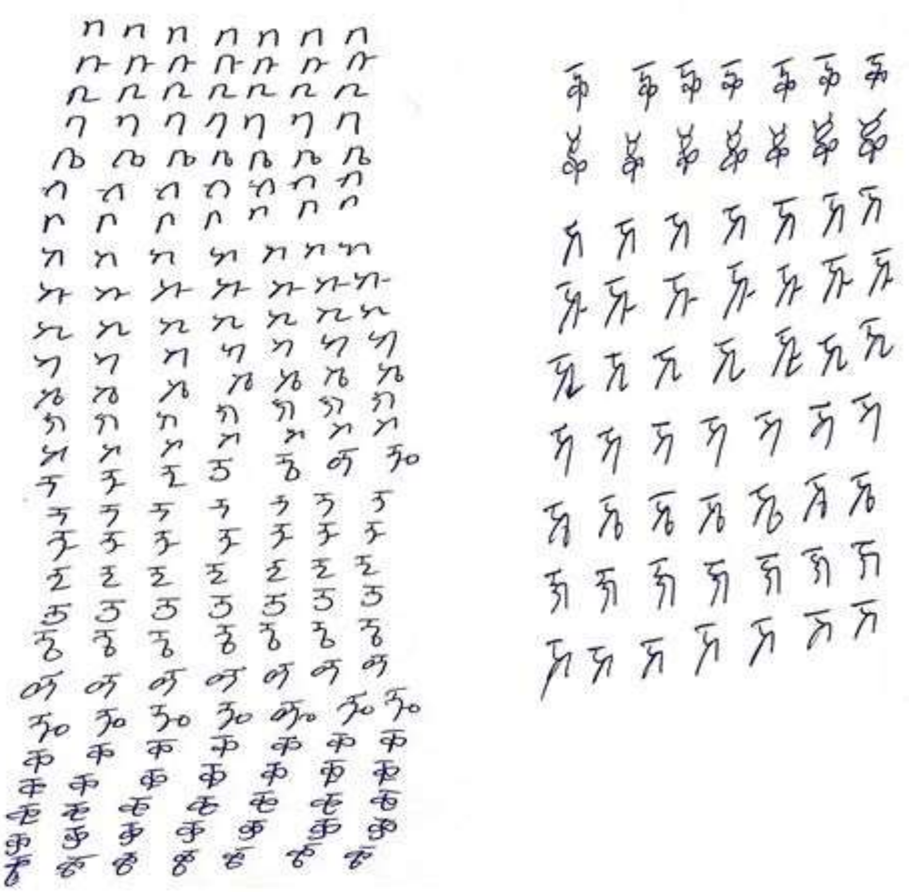


Figure 3. 1 sample written character in A4 paper

As we have seen from the above figure 3.1 is the sample of handwritten Awngi characters whose images are captured by the Samsung galaxys7 cam scanner device. Characters that are ready for prepossess

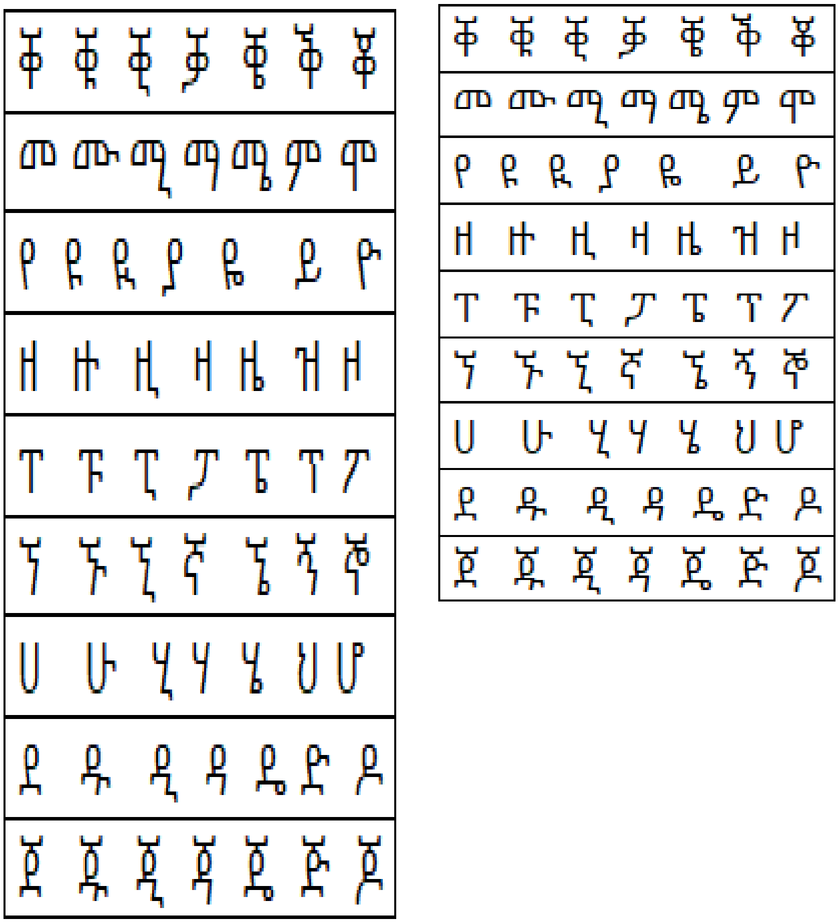


Figure 3. 2 Data reference matrixes

3.2.1. Image capturing

Scanning document is the normal and recommended way of collecting data for image processing and related fields such as handwritten character recognition. We have used a Samsung galaxys7 cam scanner device for collecting images whenever possible. Although, every image used for training and testing the proposed system is gathered through a cam scanner. We have collected about 30 pages of images from all sources. More than 35%of the collected images are gathered from documents which are found in injibara teacher's education awigni department and the rest are collected from awi museum libraries (historical documents), private books, and the Amhara mass media awigni media. We have collected 30 pages from the total of 20 pages are different poems and Awngi film scrip documents found in injibara teachers education. Also, 10 pages of the images are captured or took from handwritten books in libraries and private data or personal files. But most of the handwritten books found in the libraries are that much aged and degraded documents. This might affect the recognition system. Below a sample of a scanned document written in awigni.

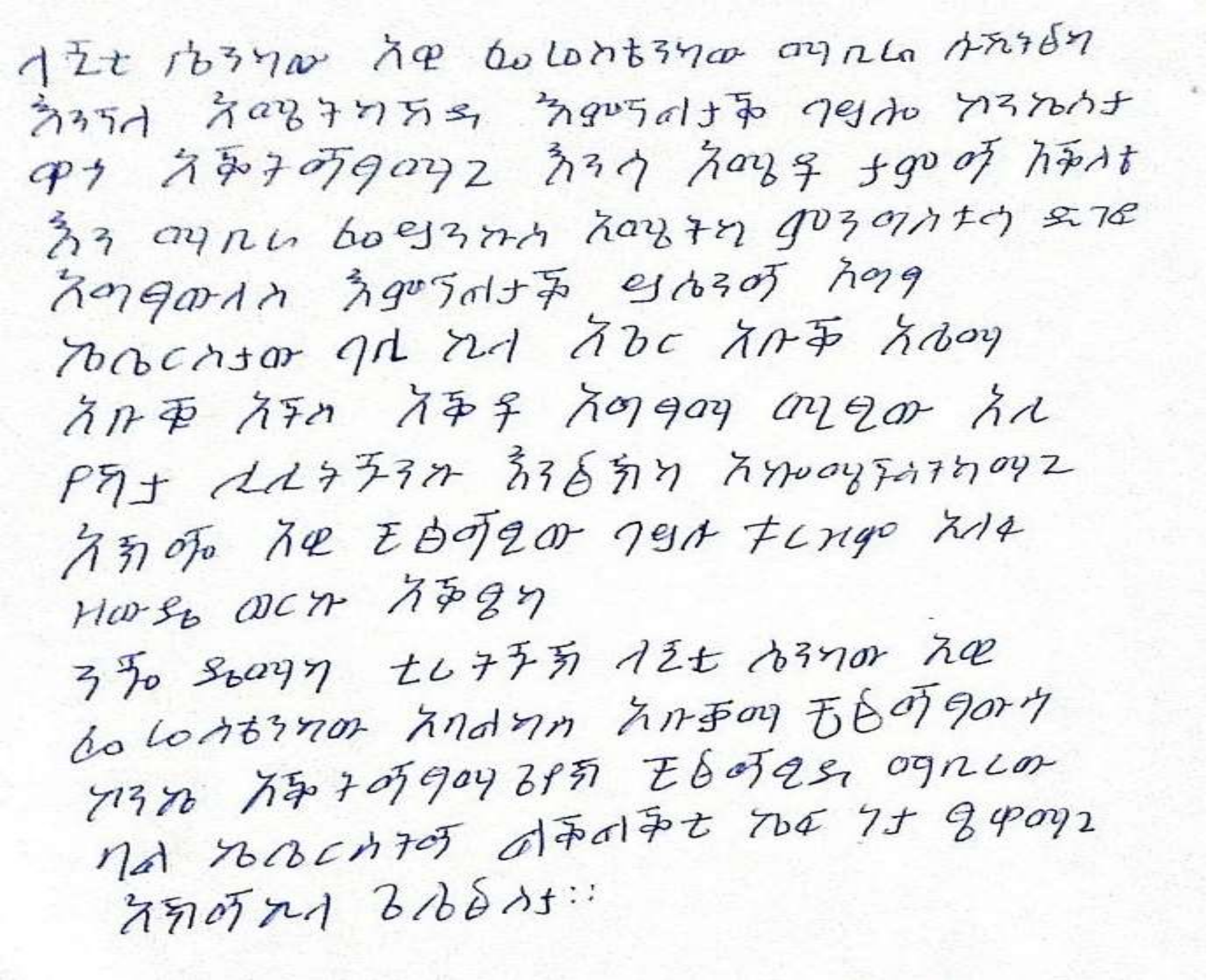


Figure 3. 3 Sample Handwritten Awngi document

3.3. Design and Development Methodology

Explains the methodologies used to design and implement this proposed handwritten Awngi character recognition model.

3.1.3. System Design Methodologies

In this section, the techniques used to implement the prototype of the system are described with their benefits. This section covers all the preprocessing techniques starting from the grayscale to binarization and noise removal methods. Also, the approaches used as a feature extractor and character classifier for the proposed system are presented in this section.

3.1.4. Preprocessing Methodologies

These preprocessing techniques include smoothing, noise removal, skew detection and correction, and morphological operations. Almost we can say all old documents are highly noisy and require some advanced level noise removal and smoothing techniques. We have used a Gaussian kernel of 5*5 for blurring and smoothing the images to remove edges and unwanted noises. After applying the Gaussian blurring technique, we have used non-local mean denoising for removing the rest of the noises that are left on the images. a skewed document can happen during scanning and copying the documents. Skew can be visualized as a slope of the text lines in the x-axis, this should be correct otherwise it will affect the accuracy of the character recognition system.

3.1.5. Binarization methodology

In general, there are three kinds of binarization techniques which are global, local, and adaptive thresholding methods. The global thresholding technique uses only one threshold value for the entire image, however; the local thresholding method divides the entire image into patches and uses one threshold value per patch. Additionally, the adaptive thresholding uses a different threshold value for each pixel in the image. We have used a global thresholding method called Otsu thresholding to separate foreground and background of images which is sometimes referred to as “Optimal thresholding” and considered to be the best and fastest method according to researches made for degraded documents. Otsu binarization automatically computes a threshold value from the image histogram for the bimodal image.It uses cv2.threshold() function with an extra flag, cv2.THRESH_OTSU. For threshold value, just pass zero. Then an algorithm discovers the optimal threshold value and returns us the second output, retVal. If Otsu thresholding is not used, the retVal remains the same as the threshold value we used.

3.1.6. Segmentation methodologies

Segmentation is the critical part of character recognition which involves segmentation of lines, words, and characters respectively from the inputted image and making the image ready for feature extraction. According to the paper (Choudhary, 2014), there are three kinds of character segmentation techniques they are listed as follows:

- Explicit segmentation,
- Implicit segmentation,
- Holistic segmentation,
- Hybrid segmentation.

Explicit segmentation: - Method is based on a vertical projection; it first scans the image from the top to the bottom and saves the position of each column. By using those columns, the algorithms divide the word images into individual character images and make them ready for the feature extractor stage. Also, this technique is sometimes called a partition because; the method follows a character cutting technique from the documents.

The implicit Segmentation approach carries the process of segmentation with the recognition side by side. It just finds the character from the image directly and it serves as a replacement for the explicit method. It is usually used in HMMs based recognition systems. Moreover, the **holistic** approach directly deals with the words rather than the character. It's also called a segmentation-free technique since it doesn't care about segmenting the characters from the words. This holistic method is highly dependent on the predefined set of lexicons of words. Therefore, mostly this approach is not suitable for character recognition systems rather it's suitable for word recognition systems. Lastly, the

A hybrid approach has been proposed and presented by many researchers to improve the segmentation process through the application of methods like linear searching and contextual knowledge. Also, some researchers introduce a new **hybrid** segmentation method in their research papers and improved the accuracy of the segmentation(Khan & Mohammad, 2008).

We have used counter tracing for each segmentation process. Contours tracing is one of the difficult and relevant tasks of image processing. It involves tracing and getting the border or boundaries of the characters. Hence, if we can locate and trace the boundaries of each character indirectly it means we exactly know where the characters are located on the images and the extraction process will become possible and easier. We have used counter-finding algorithms that trace the contours of each character from an image. This algorithm was originally designed by Satoshi Suzuki and Keiichi Abe(Suzuki, Abe, & intelligence, 1985)

3.1.7. Normalization

Especially in handwritten the size of character differs from one person to another and also from time to time even when the person is the same .therefore normalization helps us to in equating the size of handwritten character image that is call binary matrix so feature extraction can be extracted on the same size 28 x28 pixel image or 784 binary matrix.

3.2 Classification and feature extraction methodologies

There are different kinds of character classification approaches that are used by many researchers such as Support Vector Machines (SVM), structural approaches, statistical models, and Artificial Neural Networks (ANNs). Among those classifications, we have used a deep neural network-based character classification approach. The proposed system’s prototype is implemented using convolutional neural networks (CNNs) as a feature extraction method. CNN's have a special type of layer called the convolutional layer and it’s usually followed by a pooling layer. The main goal of the convolutional layer is to extract the relevant features of the characters and provide the features to the classification layers. Usually, after the last layer of the convolution model, a dense or fully connect layer will be placed to perform the classification process, using the extracted features.

Nowadays, CNN's approach is becoming the most efficient feature extraction approach in pattern recognition when it is mixed with dense layers of classifiers. The dense layers use the features which are extracted from the convolutional layers after a convolution operation is performed with the kernel or filter vector. This classification approach shows a greater improvement in recognition accuracy of a MINIST English handwritten dataset which hasn’t been achieved by any other classifiers so far. Furthermore, the CNN models are much efficient than the other approaches based on the computational power required to compute the weights of each node Also, a DNN model is designed that doesn’t have any CNN layers for comparing the results obtained by the proposed system.

3.2.1 Artificial Neural Network

An artificial neural network is used by many researchers for character classification approaches in any pattern recognition method usually includes three steps: preprocessing, feature extraction, and classification Preprocessing the data works cleaning noises of image data in some way, then it to be passed into the next stage, feature extraction, where relevant information that aids in classification is extracted. Feature extraction is a difficult process, and is usually time-consuming, and cannot process raw visual data. The third step is classification, where either the manually extracted features are used to classify the image into a specific class using an appropriate classifier. In this part, we discuss the convolutional neural network and the methods employed to performance improvement of the recognition on the given dataset.

3.2.2 Input layer

The input layer is an H*W*D pixel image, where H is the height, W is the width, and D is the depth, in pixels. A colored RGB image has D=3, while a grayscale image has D=1. Since we have fixed size of image input a 28*28*1 grayscale image.

3.2.3 Hidden layers

CNN is comprised of convolutional layers, pooling layers, and fully connected layers. Convolutional layers perform feature extraction on the input image, extracting features such as corners, edges, or endpoints, and then applying an appropriate non-linear function that allows us to avoid collapsing the network into one linear function. the system is composed of different convolutional layers, where each layer makes a batch normalization operation after the ReLU activation function. Found to work well. As we want to preserve the network's representational capacity we increase the number of feature maps as the network deep after each layer of pooling by apply padding in all of the convolutional layers to protect the problems of image reduction and information loss around the perimeter of the input character image.

$$\mathrm{ReLU}(x) = \max(0,x) \tag{3.1}$$

3.2.4 Output layer

The output layer is a layer (one class for each letter, plus the special character) on the dataset. On the dataset, the output layer is a neural output a probability-like forecast for each character class; see Equation below, where N is the number of output classes:

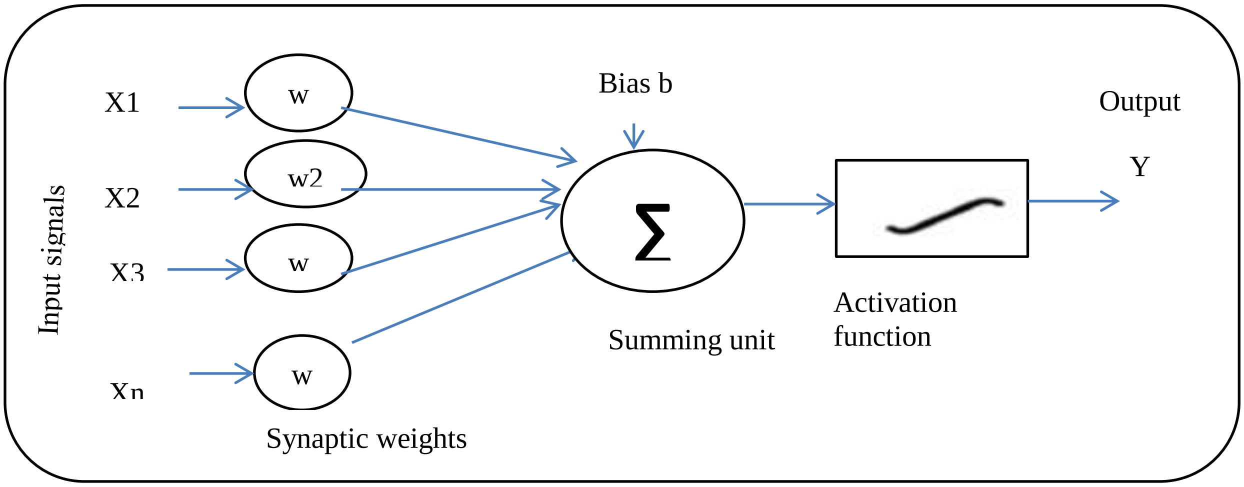


Figure 3. 4 neural network models

As the above image defines it clearly a neural network is composed of input signal values represented by X= {x1, x2, ..., xn}. Then the input signals are processed with the weights of the input node which are represented by W= {w1, w2, ..., wn}. Earlier passing the result to the activation function the input signals are summed with the summing function ∑, the mathematical formula of the summing function is expressed where U is the result of the summing. Furthermore, the activation function is responsible for defining or expressing the output of the neural network. In which the activation function decides that the neuron should fire or not. Finally, the result, or in our case the character class of the inputted image is returned by y after passing through the activation function.

$$U = \sum_{i=0}^{n} WiXi + b \tag{3.1}$$

Activation functions serve as an indicator for the neural network to make a decision based on the inputted sum. There are four most commonly used activation functions of a neural network which are linear, logistic (sigmoid), hyperbolic tangent, and rectified linear units (ReLU). From The most commonly used activation functions of a neural network three activation functions, the ReLU function is the most commonly used and considered as an efficient one for deep neural networks (Eckle & Schmidt, ,2019). There is another activation function called leaky ReLU which is a variant of the rectified linear unit (ReLU). In which its function can be represented using (y) = max $(\beta y, y)$. Also, their variants of the above-listed activations have been proposed by new researchers (Gomes et al., 2011). We have shown the ReLU activation function followed by the leaky ReLu function in which it adds a smaller constant value β to the equation (y) = max$(\beta y, y)$.

$$\text{g(x)= x} \tag{3.2}$$

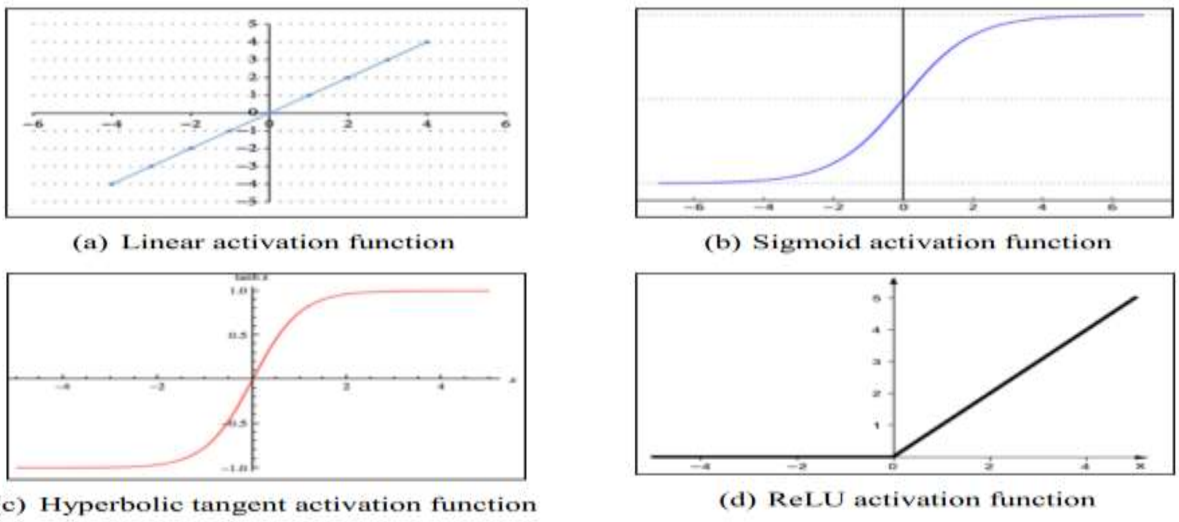
$$\text{g(x)=} \quad = \frac{1}{1+\exp(-x)} \dots \dots \dots \dots \tag{3.3}$$

$$\text{g(x)} \quad = \frac{\exp(2x)-1}{\exp(2x)+1} \dots \dots \dots \dots \tag{3.4}$$

$$g(x)= \max(0,x) \tag{3.5}$$

$$g(x)= \max(ax,x) \tag{3.6}$$

Whereas α is the smallest constant number usually 0.001. However, we can’t assign 1 for α because the model will be linear. The α constant is used to fix the neuron dying problem of the ReLU activation function. (Yingying , Yibin , & Yong, 2020). The Influence of the Activation Function in activation function a convolution neural network model of recognition.



(a) Linear activation function

(b) Sigmoid activation function

(c) Hyperbolic tangent activation function

(d) ReLU activation function

Figure 3. 5 Neural networks most commonly used activation functions

The graphs of four common different activation functions. The names of the above curve functions are the Linear activation function, the Sigmoid activation function, the hyperbolic activation function, and the ReLU activation function.

3.2.5 Support vector machines

Due to its accuracy and small usage of computational power a support vector machine is chosen by many scholars like (Nasien, Haron, & Yuhaniz, 2010) and (Malanker & Patel, 2014). The main objective of the support vector machine is to find the hyperplane from the entire number of input features that can uniquely classify the features. The hyperplane is used for a decision-making line path that will be used to classify the input features based on their location wherever the data lies on the plane using the line. The length of the hyperplane highly depends based on the amount of the input features. Fundamentally, the dimension of the hyperplanes is three because most of the time the number of the input features is more than two. However, in some cases the number of the input features is less than three then the hyperplane will be a two-dimensional plane. The decision line is highly affected and depends on the support vectors which are close to the hyperplane. Also, the margin between the support vectors and the hyperplane should be maximized to meet the main goals of the SVM model(Cheriet, Kharma, Liu, & Suen, 2007).

3.2.6 Structural pattern recognition

Structural pattern recognition is usually used for online handwritten recognition systems rather than offline recognition systems (Cheriet et al., 2007). These structural techniques use syntactic grammar for uniquely identifying the structure symbols or objects (Ajmire et al., 2012). For this reason, they are sometimes referred to as a syntactic classification approach. In general, this structural or syntonic approach requires deeper analysis to generate the morphological structure of the objects or the characters. The structure technique records the sequence of the character stroke and assembles these strokes to create the correct structure of each character. The classes that are used to classify the characters are structural templates. In which each character can be represented with one or more templates and the inputted pattern is compared with the previously obtained structural templates. Furthermore, this syntactic technique doesn’t provide a general similarity pattern that can identify the character as a whole but it rather provides a similarity index for each stroke or the components that make a character. But pattern or feature extraction in this approach might be difficult.

3.3 Convolutional Neural Network

In fully connected layers, each unit (neuron) is connected to all of the units in the previous layer. In CNN, however, each unit is linked to a minor number of elements in the previous layer. In addition, all elements are linked to the earlier layer in the same way, with the same weights and structure. CNN uses the convolution process in its place of overall matrix multiplication in at least one of its layers. Convolution is an element-wise multiplication of two matrices followed by the sum. The basic CNN architecture is given in figure 3.6 below.

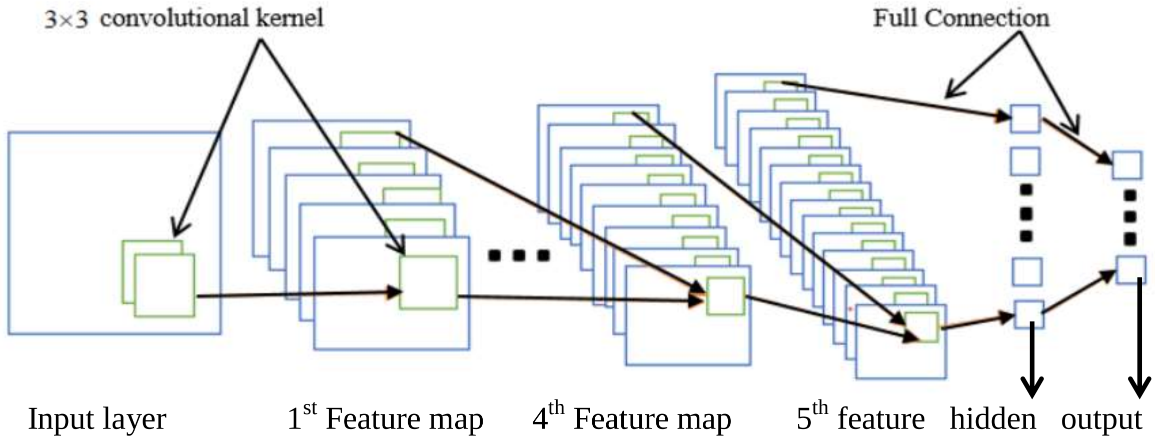


Figure 3. 6 Basic CNN architecture

In deep learning libraries use the simplified cross-correlation function as a convolution operation (Adrian, 2017). Convolution (denoted by * operator) over a two-dimensional input image i and two-dimensional kernel K is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n K(i+m, j+n)\, i(m,n) \qquad (3.7)$$

The kernel K (tiny matrix) sits on the top of the input image I (image matrix) and slides from left-to-right and top-to-bottom, applying a mathematical convolution operation at (x, y)-coordinate of the input image I. CNNs can learn filters (kernels) that can identify edges in the lower-level layers of the convolutional network and then convolutional neural network uses the edges as building blocks to detect high-level objects (features).

CNN gives us three key benefits: local invariance (via pooling layers), compositionality, and shared weights(Adrian, 2017). (Gulli & Pal, 2017). Local invariance is used to detect an image containing a particular object regardless of its spatial location. When designing CNNs, it should be invariant of translation, rotation, scaling, and any other sorts. A CNN computes the same features of an image across all spatial areas(Hope, Resheff, & Lieder, 2017) Compositionality is used to learn deeper features in the model network. Each filter comprises a local patch of the lower-level features into the higher-level representation. A network may learn edges from pixels, shapes from edges, and then complex features from shapes. This property gives CNNs so powerful in computer vision.

When we want to change away from pixel illustration in a row by in advance the ability to detect the same feature individually from the position where it is placed in the input image, a simple intuition is to use the same weights for all neurons in the hidden layers. Each layer will then learn a set of location-independent latent features derived from the input image (Gulli & Pal, 2017).

3.3 Building Blocks of Convolutional neural network

Many layers are used to build CNNs such as convolutional, activation, pooling, fully-connected, and dropout layers. Following is the detailed description of each layer.

Convolutional layers: considered as the core building blocks of CNN architectures. Convolution takes input (raw data or feature map output from another convolution), applies a convolutional kernel (filter), and outputs a feature map. Filters are small in terms of their spatial dimension (but extend throughout the full depth of the volume) as compared to the input image and are nearly always square. On images with large spatial dimensions, it is often impractical to connect a neuron to all other neurons in the previous layer. When implementing CNNs, we choose to join each neuron only connected a local area of the input volume – we call this local connectivity and the size of this local region the receptive field of the neuron.

If the receptive field is of size 5 x 5, then each neuron in the convolution layer will connect to a 5 x 5 local region of the image for a total of 5 x 5 x 3 = 75 weights (if the image has a depth of 3 (one for each RGB channel) 5 x 5 x 1=25 weights (if the image has a depth of 1 (one for each grayscale channel).

There are three parameters that control the size of an output volume: ***depth***, ***stride***, and ***zero-padding*** size(Akal et al., 2018). The set of filters that are looking at the same (x, y) location of the input is called the depth. The depth controls the number of filters (neurons) in the convolution layer that join to a local region of the input volume. If the volume size is 8 x 8 x 32 and the receptive field size is 5 x 5, then each neuron in the convolution layer will connect to a total of 5 x 5 local region of the image for the total of 5 x 5 x 32 = 800 connections to the input volume. Stride defines by how much we skip pixels in the x- and y-coordinates. It controls the spatial movement of the filter across the image or feature map. It introduces the size of skips in the application of the filter(Hope et al., 2017). Smaller strides lead to overlapping receptive fields and larger output volumes, while larger strides lead to smaller output volumes. the size of output volume is reduced much when using a stride size of 2.

Zero-padding means padding borders of the original image with zero to keep the original image size when applying a convolution. Without zero-padding, the spatial dimensions of the input volume would decrease too quickly. The output of applying a 3 x 3 filter on zero-padded 5 x 5 image left while the output of applying 3 x 3 filter without zero-padding the original image is shown in the right. As it is clearly seen, zero-padding helps to preserve the spatial dimensions of the original 5 x 5 image.

Activation layers: since no parameters are learned in activation layers, they are not technical layers. The output of the activation function is always the same as the input dimension due to the fact that the activation function is applied in an element-wise manner(Adrian, 2017) After linear layers, it is well-known practice to apply nonlinear activation functions. Rectifying linear unit (ReLU) is the most widely used activation function in CNNs. CNN's that apply ReLU train several times faster as compared to applying other activation such as tanh. ReLUs do not require input normalization to prevent them from saturating (Krizhevsky, Sutskever, & Hinton, 2017). Activations are done in-place so there is no need to create a separate output volume. ReLU outputs the maximum of zero, and the number (image pixel). i.e., max (0, x). rights below show the output of applying ReLU to a 3 x 3 input volume.

Pooling layers: used to progressively reduce the dimensionality of the input data and help reduce overfitting. This, in turn, helps to reduce the number of parameters and the amount of computation in the network. It takes a local receptive field and replaces the nonlinear activation function at each portion of the field with the max or min or average function(Ramsundar & Zadeh, 2018)   The most common type of pooling layer is max pooling, which is used in the middle of the CNN architecture to reduce the spatial size. If our pool size is 2 x 2 and we apply max pooling, then we keep only the maximum or the largest value in each 2 x 2 block regions and the height and width of the input volume will be reduced by a factor of 2. However, we can further reduce the dimensionality of the input volume by increasing the stride size. the effect of max pooling with a stride size of 1 (top-right) and 2 (bottom-right) while applied on a 4x4 input volume (left). Applying a max pooling with a large stride size (such as 2) will intensely reduce the spatial dimensions of the input volume, discarding 75 % of activations from the previous layer.

Fully-connected layers: neurons in this layer are fully connected to all neurons in the previous layer. They have normal parameters for the layer and hyperparameters. Fully connected layers are used to calculate class scores that will be used as the output of the network (Josh & Adam, 2017). They are applied at the end of the network before applying the classifier (usually Softmax classifier).

Dropout: is a technique of turning off some fraction of units in a layer, by setting their values to zero during training and turning on these units by setting their values to one during testing. It is a regularization form used to force the network to distribute the learned representations across all neurons and hence force the network to learn a representation that works after the dropout(Hope et al., 2017). Dropout reduces overfitting by explicitly altering the network architecture at training time. It ensures every single node to be activated when presented with a given pattern. Dropout avoids dependency of the network on small units of neurons (Adrian, 2017).

3.4. Proposed model of the Awngi handwritten character recognition

In our proposed model we have used convolutional neural networks for the recognition of the handwritten Awngi characters. We have major units in the model: preprocessing of collected data and segmentation of characters and the third one is extracting features from character images and using those features for classification and recognition. Resizing, noise removal, grayscale binarization, morphological transformation, and segmentation belong to the preprocessing step in the proposed system. The two essential components in recognition, feature extraction, and classification, both are done in our CNN model.

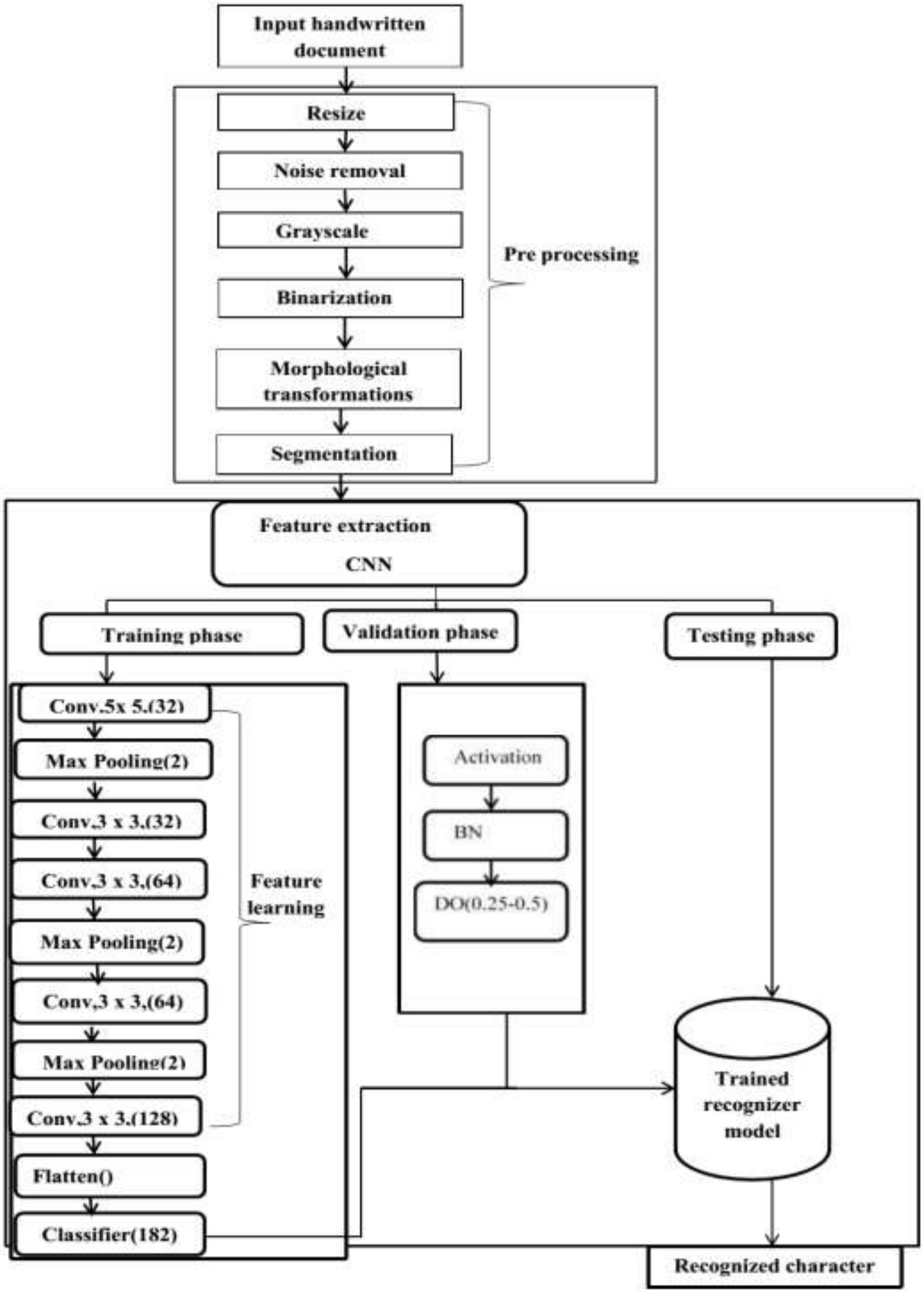


Figure 3. 7 proposed model architecture

3.4.1. Image Acquisition

Image acquisition is the first and most important step of any printed or handwritten character recognition system. Generally, for any kind of image processing system image acquisition is the primary and fundamental task before conducting any kind of activities. Note that each step of the character recognition process is highly dependent on the previous. Most documents that are used for image recognition are captured using a scanner. However, sometimes there can be documents that are difficult to scan using a scanner because of the materials used. Thus, Samsung Galaxy S7 mobile camera was used to capture the documents

3.4.2. Image Preprocessing

Once the relevant images are collected the images will pass through the preprocessing stages. The purpose of the preprocessing phase is to eliminate irrelevant pixels to form an enhanced representation of the input images and make them ready for the following stages. The preprocessing stage involves different step-by-step processes such as grayscale conversion, binarization, and smoothing skew correction, and noise removal.

1. Scale down: Before starting any kind of process on the images, each image should be scaled down or normalized to a common size. This will improve the speed and accuracy of the recognition process. Images taken with a camera are bigger and when we start processing these input images directly then the process will take a longer time to process. Therefore, before processing the images we need to scale down the images to a common size of mostly between 300dpi and 600dpi. This will improve the recognition process by reducing the memory load and execution time of the system.

```
Input: A sample document image I
Output: An array of resized  document image (to 600, 600 dpi)
Begin
  Image = original document image I
  Image = resize (Image, 600, 600 dpi)
  return Image
End
```

Algorithm 4 1 applying document scale down

2. Noise reduction: We then apply a suitable Gaussian filter to the noise character image to eliminate any kind of salt and pepper noise, since our algorithm contracts with the calculation of gradient vectors to sense character boundaries. The noise of Salt and pepper can reason the algorithm to crash false positives and result in the generation of noisy data. Some unusual cases arise when the lines in the images are very thin, in which case the Gaussian filter destroys large chunks of lines. We used RMSE (Root mean square error) as a measure to make sure this does not happen. While dealing with handwritten document recognition especially while conducting old document recognition since the documents are highly noisy and degrade. In the proposed system we have used a noise reduction algorithm called non-local means denoising which was originally proposed in a research paper by(Buades, Coll, & Morel, 2011). It works by finding the average of similar pixels and updating the pixel value with the calculated average pixel value. The entire image is scanned for discovery pixels having similar intensity within the pixel to be denoised. Denoising is done by computing this average value of the pixels having similar values.

```
Input: scale down document image I
Output: de noised document image
Begin
   Image = original scale down document image I
   Image = GaussianBlur (Image ,(3,3),0)
   return Image
End
```

Algorithm 4 2 Algorithm for noise reduction

3. Grayscale: Usually, the images which are collected using a scanner or camera are RGB or colored images. And if we are dealing with an image processing approach, which can’t be affected by the color of the images then, it’s a good practice to convert each image into a grayscale image. The input images are stored in a three-dimensional (RGB) array or vector-matrix but we should convert the images into a grayscale image to reduce the computational cost. Then the grayscale image is stored as a two-dimensional matrix making the recognition processes much easier. Commonly, in colored images, every pixel or point is denoted with three values of RGB, in which each value is represented as an 8-bit integer by the range of 0 to 255. However, in grayscale images, every point is expressed with a single value that varies between0 to 255. Each pixel value in grayscale images represents how much that point is white. A pure white is represented with 255 and a pure black with 0.

```
Input: de noised document image I
Output: grayscale document image
Begin
  Image = original de noised document image I
  Image = cvtColor(Image, cv2.COLOR_BGR2GRAY)
  return Image
End
```

Algorithm 4 3 Algorithm for grayscale

4. Binarization: Once the images are converted to grayscale and the irrelevant noises are removed from the images then the binarization phase takes place. In which it converted the grayscale image into binary images so that the image is composed of pixels having a color value of either black or white. In the proposed system we have used an inverted Otsu thresholding algorithm to convert the images into binary format. The inverted method converts the background with black color and foreground or characters with white color. The Otsu method is intended to find a threshold value that can separate the foreground of the image with background i.e. the black pixels with white pixels. So that it can minimize the overlap that occurs between the white and black pixels. The Otsu algorithm uses the threshold value and increases the distribution of either of the pixels and decreases the other left pixel. Simply it calculates the thresholding value and separates the entire image pixels into two groups and then it finds the mean of each group. Then it squares the result of the subtraction of the mean of the two groups and finally multiplies the number of pixels in the two groups. The result we have after applying the inverted Otsu thresholding algorithm is in the opposite format. In which the pixels of the character are represented with white color and the background is represented with black color. As a recommendation, there is no efficient thresholding algorithm that can be suitable for every problem. Therefore, it's recommended to include more than one thresholding procedure and leave the choice for the system users to apply the efficient algorithm based on their choice at the run time.

```
Input: grayscale document image I
Output: binary document image
Begin
  Image = grayscale document image I
  Thresh = threshold(grayscaled, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
  return Thresh
End
```

Algorithm 4 4 algorithm for Binarization

5. Morphological transformations Furthermore, we have also used some other transformations that are used to remove the irrelevant pixels that happen during the task of binarization The first transformation operation applied is opening that will remove the boundaries of the foreground or the character so that the character becomes more visible. Secondly, we have applied a transformation operation called dilation. The dilation technique Dilation is used for filling gaps of the given image. This dilation transformation is useful to improve the characters that are too thin by adding pixels on the boundaries of the character. Then finally closing transformation is applied, which will help in filling the openings that happen inside the foreground of the images. All of these morphological operations provide us with the ability to reduce noises that are missed in the noise removal stage and noises that happen during the execution of the other stages. Thus, through thinning and filling some gaps that occur between characters we have improved the preprocessing phase the noise-reducing stage. However, these operations have drawbacks; they might end up with resulting additional noises. Morphological transformations are some simple operations based on the input image shape. It is normally performed on binary images. Therefore transformation needs two inputs, one is our original input image, the second one is called the structuring element or kernel which selects the nature of the operation. basic morphological operators are Erosion and Dilation we have used for developing the proposed model.

```
Input: binary document image I
Output: Morphological image
Begin
   Thresh = binary document image I
Dilate = dilate(Thresh, None, iterations = 5)
   return Dilate
End
```

Algorithm 4 5 Algorithm for Morphological transformations

3.4.3. Segmentation

In the handwritten recognition process Segmentation is a serious step and it vastly affects the recognition process. For each segmentation process, contour analysis plays a greater role in the planned system. Essentially, segmentation of line and character segmentation, a special method of adding extra foreground colors horizontally is used. In overall satisfying additional pixel which is also called dilation, the technique is used for line segmentation. However, the word segmentation method is not shown in the segmentation process because the aim is character recognition.

3.4.4. Character segmentation

In the proposed model for the character segmentation. To segment or crop the character, we have used a technique of contour analysis and bounding box. The steps tried for character segmentation is defined under

1. Accept the list of lines that are returned from the line segmentation stage.
2. Find all contours/characters on the line and return a list of coordinates for each contour.
3. Crop the characters from the threshold text line images using the returned coordinates.
4. Iterate through each line.

3.4.5. Character Blob detection

To recognize the regions where the character is present on the handwritten paper documents, we have used morphological operations such as dilation for character region detection.

Different kernels are used to segment the character region. The character contours over the dilated image give X, Y coordinates used to segment characters from the original image document.

3.4.6. Character Contour segmentation

The area where the character is present is identified, and then contour and bounding box are performed (as shown Fig below). The bounding box is then separated, into a character. The contours are not in the order in which the character is written, so they are arranged based on their X and Y coordinates of the identified contours, to get the contour images in their sequence in which the character appears on the image.

Fundamentally, the contour analysis algorithm uses the edge or border finding method to find all objects on the image. The algorithm finds the borders of each contour and returns their location coordinates. The coordinates returned are the points of the top-left, top-right, bottom-left, and bottom-right corners of each contour. Then a coordinate is used to process the contours of objects. The steps that how the algorithm follows to find contours are stepped below.

1 After the image is binarized will scan the image and update *P* is a calculated successive number of the lastly found outside most border.

2. Characterize each hole by decrease the pixels into a single pixel-based on a given threshold value *t*. (where *t* is the smallest perimeter of a character).

3. Characterize the outside border by decrease its pixels into a single pixel founded on a given perimeter or threshold value *t*.

4. Place the shrank outer border pixel character to the correct of the shrank hole pixel.

5. Cutting the surrounding relatives connected components i.e. holes and outer borders.

Bounded Box -To obtain a bounded box covering the only character, removing any black border around it. Then we resize all of the cute images to get a character with the images of size 28 X 28. This gives us an identical set of images, of the same sizes. To detect and find all shapes using contour function. For each contour found, bounding boxes are drawn with additional height threshold h. This height helps each contour to connect with another contour above or below. Then, another contour function is performed to extract the height and width information of each character.

```
Input: image I
Output: Segmented character  image
Begin
  image = image I
For i in range (len_x)// binary length of horizontal x
for j in range(len_y)// binary length of vertical y
for i, counter  in enumerate
boundingRect(counter )
 [y:y+h, x:x+w] // Getting ROI of characters from the document
          Resize(28,28)
End
```

Algorithm 4 6 Segmentation Algorithms

3.5. Feature Extraction and Classification

Once the segmentation process is completed the model has all of the isolated characters and it's arranged to extract the features that will uniquely represent the characters. Convolutional Neural Networks are composed of two parts. The main part of CNN contains several convolutional layers which is a distinctive kind of neural network that is accountable for extracting features from the inputted handwritten character image. The second part of CNN is dense neural layers which are obligatory to classify the inputted handwritten character based on the features mined from the convolution layers. On the proposed system we have used 3 convolutional layers for feature extraction and 2 dense layers for classification purposes. The main goal of any neural network is to decrease the loss function which can be given either by cross-entropy or means squared error.

3.5.1. Training phase

In the training phase feature learning each phase to learn the characteristic features that characterize the patterns of the handwritten Awngi character.

Convolution layer: The input image is accepted through a stack of convolution layers which is the main component of a CNN whose job is to detect important features parts of the input document image of pixels. The input of the first convolution layer is a 28 x 28 x 1 handwritten character image. The convolution process needs four parameters. Filters are the first parameter of the convolutional operation used to control the depth of the output volume. In our model, we have used 32, 64, and 128, filters. The number of filters we have used is bigger as we go down to fully connected layers and Softmax classifier. A dissimilar number of convolution layers and a dissimilar number of filters are tested and those that achieve higher accuracy are selected.

The second parameter Convolution layer is the field size, which controls the size of each kernel (filter) and is nearly always square. We have used 5 x 5, 3 x 3 filter size, or kernel size in our model.

The third parameter is **stride** size that controls the number of pixels skipped both vertically and horizontally we make a convolution process each time. We have used stride sizes of two (2, 2) and one (1, 1). When stride size is 2, the image length is reduced by half vertically and horizontally.

The fourth or the last parameter is the zero-padding, which is used to manage the size of the output. In this paper, we have used “same” padding, which means that the size of the output is equal to the size of the input when the stride size is one. When we do not want to change the size of an input volume through convolution operation, we have used stride size of one and padding “same”, otherwise we apply stride size of two. We have done convolution operations repeatedly before the input image is down-sampled with the pooling layers (operation). Functioning multiple convolution layers before applying a pooling layer allows the model to develop more complex features before the destructive pooling operation is done.

Selection of filter size

Many scholars use 3 x 3 or 5 x 5 receptive field filters with a small (which is the smallest size to capture the notion of left/right, up/down, and center) all over the CNN or sometimes can be changed in between the architecture. In this paper, we selected the filter size scientifically based on the characteristic features in the convolution module, we have used 3 x 3, the filter size is considered and slide over the whole image.

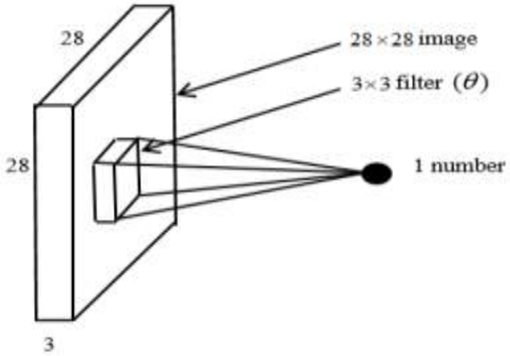


Figure 3. 8 convolving an image of 28 x 28 with a filter of 3 x 3

Activation layer: The output of the activation function is always the same as the dimension (size) of the input. The activation function is applied in an element-wise way. Later, the width, height, and depth of the output layer are the same as the width, height, and depth of the input layer respectively. In this thesis, we used ratified linear unit (ReLU) as an activation function in the activation layer throughout our model. ReLU activation function returns zero if the values in the input layer are negative, otherwise, it returns the existed value. Mathematically, it is defined as:

$$y = \max(0, x) \qquad (3.8)$$

Pooling layer: We use pooling operation to progressively reduce the width and height of the input volume. We have applied max pooling to change the size of the input volume. Pooling operation requires two parameters. The first parameter is pool size, which controls by how much we want to reduce the spatial size of the input volume. Since our input image size is (28 x 28 pixels), we have applied a pooling size of two (2 x 2) after the consecutive convolution layers. The second parameter is the stride size, which determines the number of pixels we skipped while doing the pooling operation. We have used a stride size of two each time we do a pooling operation. The output of the pooling layer after each pooling operation is then W output x H output x D output (Adrian, 2017)

Where:

W output = ((W input - F) / S) + 1 (3.9)

H output = ((H input - F) / S) + 1 (3.10)

D output = K

Fully connected layer: We have used only one fully connected layer to compute the final output probabilities for each class before applying them to the Softmax classifier. It is due to the move from a fully connected layer to average pooling result in a 0.6% increment of accuracy(Abadi et al., 2016). It holds 182 nodes (equal to the number of classes), which are directly applied to the Softmax classifier.

Dropout layer: is used to reduce overfitting by randomly disconnecting inputs from the previous layer to the next layer in the network architecture. Random disconnection ensures that there is no single node is responsible for "activation" when random disconnection presented with a pattern. It enables multiple, redundant nodes to activate when given with similar patterns (inputs), which also helps our model to generalize. We have applied dropout layers with p (dropping probability) = 0.3 immediately before the fully connected layers, which is followed by the Softmax classifier.

3.5.2. Softmax classifier

The output of the final fully-connected layer is given as input to the Softmax classifier. A 182-way Softmax is used to classifying input patterns into a specific class. Features of the input sample are extracted through a series of convolution and pooling layers. Then, some linear layers and Softmax classifier are used to determine the prediction probability that the input image belongs to each category the loss at this epoch is obtained by calculating the cross-entropy loss, the network's parameter is optimized by back propagation algorithm to minimize the loss of network output (Qiuyu & Zikuang) loss mainly depends on the corresponding probability of the network output of the correct category through softmax.

3.5.3. Validation Phase

In Feature Learning for Validation Phase, our goal is to improve the performance of the model The performance is improved, when the accuracy of the model is improved or the loss function is reduced. In addition to the layers or operations in the training phase, we apply the following techniques to increase the performance of our model.

Batch normalization: before applied to the next layer in the network, Batch normalization normalizes the activations of the given input volume. Batch normalization is used to decrease the number of epochs taken to train a neural network and become stable training; it slows down the training time of our network **(**Adrian**,** 2017**)** however, outweighs the negatives. Also, it makes learning rate and regularization less volatile to tune. If we consider x to be our mini-batch of activations, then we can compute the normalized via the following equation(Adrian, 2017)

$$xi = \frac{xi - \mu\beta}{\sqrt{a\beta^2 + £}} \qquad (3.11)$$

During training, we compute the μβ and σβ over each mini-batch β, where:

$$\mu\beta == \frac{1}{m}\sum_{n=1}^{m} Xi \qquad \mu\beta = \frac{1}{m}\sum_{n=1}^{m}(Xi - \mu\beta)^2 \qquad (3.12)$$

During testing time, the mini-batch μβ and σβ are replaced with running averages of μβ and σβ computed during the training process.

Dropout: we applied dropout at the final stage during the training phase. Now we also apply dropout at the initial stages after the downsampling operation via pooling and improve the performance of our model. In this time, however, we apply dropout layers with smaller probability, p = 0.1 following convolution layers. At the early stages, the dropping probability should be small since dropout is less effective here due to the local connectivity of convolution layers (Adrian, 2017). However, it is helpful to combat overfitting.

Learning rate: we use a smaller learning rate to reduce the weight update. To reduce overfitting we have used decaying the learning rate helps us to reduce overfitting. We have divided the initial learning rate by the number of epochs (1-50) each time the network is trained. In our case, the network at the first epoch is trained with the learning rate of 1e-3, while at the last epoch (epoch 50) the network is trained with 1e-3 / 50 = 1e-5. As learning started to slow dramatically around epoch 10, we stopped training at epoch 50. Keeping the learning rate high will lead to overshooting of areas of the low loss since we are taking large steps to descend into these areas. It is better to decrease the learning rate progressively, and hence taking smaller steps. Reduced rate enables the network to descend into a lower loss landscape (without missing it) effectively. According to (Adrian, 2017), the learning rate scheduler generally has two main objectives. These are: (1) finding a set of learning weights early in the training process with a higher learning rate. (2) Tuning these (learned) weights later in the process to find more optimum weights using the smaller learning rate. We have applied learning rate schedulers that reduce gradually based on the epoch number.

3.5.4. Optimization Algorithms to our model

Optimizers are used for weight updates of the network and they have their behaviors. There are a lot of optimizers algorithms used to change attributes of your neural network like learning rate and weights to reduce the losses. So Optimization algorithms have responsible for reducing the losses and provide the most optimal results possible.

Stochastic gradient descent (SGD): is an optimization algorithm that computes the gradient of the network loss function for each weight in the network. Every forward pass through the network results in a certain parameterized is the loss function, and each of the gradients we've created for each of the weights, multiplied by a certain learning rate, to move our weights in whatever direction its gradient is pointing. In SGD, while we run through the training example and we encounter the training example, we update the parameters according to the error for a single training example (Wu & Goodman, 2018). The algorithm is given in the equation below.

For i=1 to m ,{ (3.13)

}

$\theta_J = \theta_J + \alpha\ (y^{(i)} - h_\theta(x^{(i)}))xj^{(i)}$ (for every j)

}

It is called stochastic since samples are selected randomly from the training data in each run to update parameters during optimization. In SGD, unlike the standard gradient descent, we only use one or a subset of training samples from the training set to do the update for a parameter in a particular iteration. SGD also updates weights more frequently than the standard gradient descent. the optimizer is configured with the default hyperparameters. Stochastic gradient descent has a default learning rate of 0.01

Adam optimizer (adam): Adam (Adaptive Moment Estimation) works with momentums of the first and second instruction. The perception behind the Adam is that we do not want to roll so fast just since we can jump over the minimum, we want to reduction of the velocity a little bit for a careful search. In addition to maintains an exponentially decaying average of the past squared gradients like AdaDelta, Adam also keeps an exponentially decaying average of past gradients M(t).M(t) and V(t) are values of the first moment which is the Mean and the second moment which is the uncentered variance of the gradients respectively. The main advantage of using Adam optimizer in the convolutional neural networks is Adam optimizer method is too fast and converges rapidly learning rate. Rectifies vanishing learning rate, with high variance. However, the Adam optimizer has its disadvantage is computationally costly. Adam optimizer uses default a learning rate of 0.001

RMSprop Optimizer:- RmsProp (Root Mean Square Propagation**)** is one of the training optimization algorithms that use the magnitude of new gradients to normalize the gradients. Optimizers like AdaGrad have a problem of the learning rates become low after a few batches, when the learning rate becomes low the result is a long training time. This problem is solved by Root Mean Square Propagation (RMSProp), by decaying the learning rates. This makes RMSProp more volatile. Root Mean Square Propagation uses a learning rate of 0.001, and epsilon is 1e-7

Adagrad(adaptive gradient algorithm): is an adaptive learning rate method that means we adopt learning rate to the parameters we compute or perform large updates for rare parameters and small updates for regular parameters. When we have a thin dataset degrade is a well-performed optimizer

3.5.5. Testing Phase

In this phase, we will follow the same procedure as the training phase We have to preprocess (resize, noise removal, binarization, morphological transformation, and segmentation) the handwritten character image in the same manner as the training phase all steps of preprocessing is done. If we follow any other way, it will lead us to the wrong classification since the network may be presented with patterns (inputs) it cannot classify patterns. Similarly, feature learning is also done in the same manner as in the training phase by using the learning model constructed from the training and validation phase. Input handwritten character images that are different from the training datasets are used, which are called testing datasets.

3.6. Model Evaluation Technique

Different performance metrics have been used to evaluate the performance of the proposed solution or model. Among these, accuracy, precision, recall and f1-score are used extensively for measuring the performance of proposed solutions.

Accuracy is the ratio of true positives (include both true positives and true negatives) against the whole population. Accuracy may mislead the quality of the model if the class is not balanced.

$$\text{Accuracy} = (TP + TN) / (P + N) \qquad (3.14)$$

Precision is the proportion of true positives against the whole positives. Mathematically, it is expressed as:

$$\text{Precision} = TP / P \qquad (3.15)$$

Recall or sensitivity is the proportion of true positives against the whole true or correct data. It quantifies how well the model avoids false negatives [26]. It is also known as true positive rate or hit rate.

Recall = TP / (TP + FN) (3.16)

F1-score is the weighted average precision and recall. The relative contribution of precision and recall to the F1-score are equal.

F1-score = 2 * (precision * recall) / (precision + recall) (3.17)

Micro-average, macro-average, and weighted-average for all the aforementioned performance metrics can also be calculated and used for additional analysis of results.

Macro-average precision or recall is just the average of the precision and recall (respectively) of the model on different classes.

Macro-average precision = (P1 + P2 + … + PN) / N (3.18)

Macro-average recall = (R1 + R2 + … + RN) / N

Micro-average precision or recall is calculated by summing up the individual true positives, false positives, and false negatives for each class

$$\text{Micro-average precision} = \frac{TP1 + TP2 + \ldots + TPN}{(TP1 + TP2 + \ldots + TPN) + (FP1 + FP2 + \ldots + FPN)} \quad (3.19)$$

$$\text{Micro-average recall} = \frac{(TP1 + TP2 + \cdots + TPN)}{(TP1 + TP2 + \cdots + TPN) + (TN1 + TN2 + \ldots + TNN)}$$

3.7. Recognition

The basic thing in pattern recognition is the design of a classifier, a mechanism that takes features of objects as its input and which results in classification or label or value indicating to which class the object belongs. The performance of a classifier is ultimately limited by the overlap of the classes in feature space. Classifiers can be categorized into one of the following categories, Decision theoretic and Structural method

Recognition is a process to decide whether a new object belongs to a particular pattern group or not, depending on whether its features fall inside the domain of that group or outside of the domain of that group, respectively. The performance of the recognition system was evaluated by using the collected dataset after the classifications we test the model by randomly generate characters images from the test data set.

CHAPTER FOUR

4. EXPERIMENT, RESULT AND DISCUSSION

4.1. Introduction

In this chapter, we discuss the datasets used, simulation tools, and the experiments carried out to train, and test the effectiveness of proposed character recognition.

4.2. Prototype Implementation

The proposed system of the prototype was designed using different supportive libraries on the python Jupiter notebook, which is a python editing tool. Between those helpful libraries Keras, OpenCV, and TensorFlow are the best import libraries used. Almost all methods used for preprocessing of the composed image to eliminate noise, image reading, and other relevant tasks are done using functions which are provided by OpenCV. We have used Keras API for developing the convolutional neural network architecture and training. Also, Keras has its graphical representation of the processes of training the neural network. The specifications of the programming tools, supportive libraries, and execution environment are discussed below.

Programming language: Python 3.7 used for developing the proposed system.

Libraries: Some of the frequently used libraries used during the employment of the proposed system are listed below.

1. **TensorFlow:** For normalizing image vectors and as a supporting library for Keras.

2. **Keras:** For implanting the CNN architecture in a much easier way.

3. OpenCV: For preprocessing and other relevant tasks.

4. **NumPy:** For converting the images into integers and storing them as a 2D vector.

Execution environment: The conditions of the execution environment used while applying, training, and testing the proposed system are shown in the following table.

Table 4. 1 Execution environment specifications

No.	Type	Specification
1	Processor	Intel (R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80GHz 2GB GPU
2	RAM	4 GB DDR4
3	Storage	1TB HDD
4	Architecture	64bit
5	Operating System	Windows 10 professional

4.3. Prepossessing result

Before the classification procedure, a prepossessing procedure will be performed from the image captured handwritten Awngi character to the last of dataset preparation for training our model. prepossessing starts from image resize, noise removal, grayscale, binarization morphological transformation, and segmentation and normalization algorithms result of Awngi characters were showed clearly.

4.3.1. Noise removal

In handwritten character recognition, the first step is preprocessing stage in our proposed model after the image document is scanned then the noise from the document is removed after the removal of noise grayscale conversions were performed on the document. We have used from interpolation resizing technique interpolation=cv2.INTER_LINEAR cv2.resize (800,800) method because we are deal with the large document image. Then we compute noise removal techniques

As we depicted in Figures 4.1(a) and 4.1(b) The noise reduction process accepts the scanned document and performs noise reduction from the image. The OpenCV method is named cv2.GaussianBlur is used for denoising the image document. We have applied different kinds of noise removal techniques like median filter Gaussian blur as many authors stated that Gaussian blur is the optimal noise reduction method for degraded and ancient documents in this paper we have used Gaussian blur for noise reduction method.

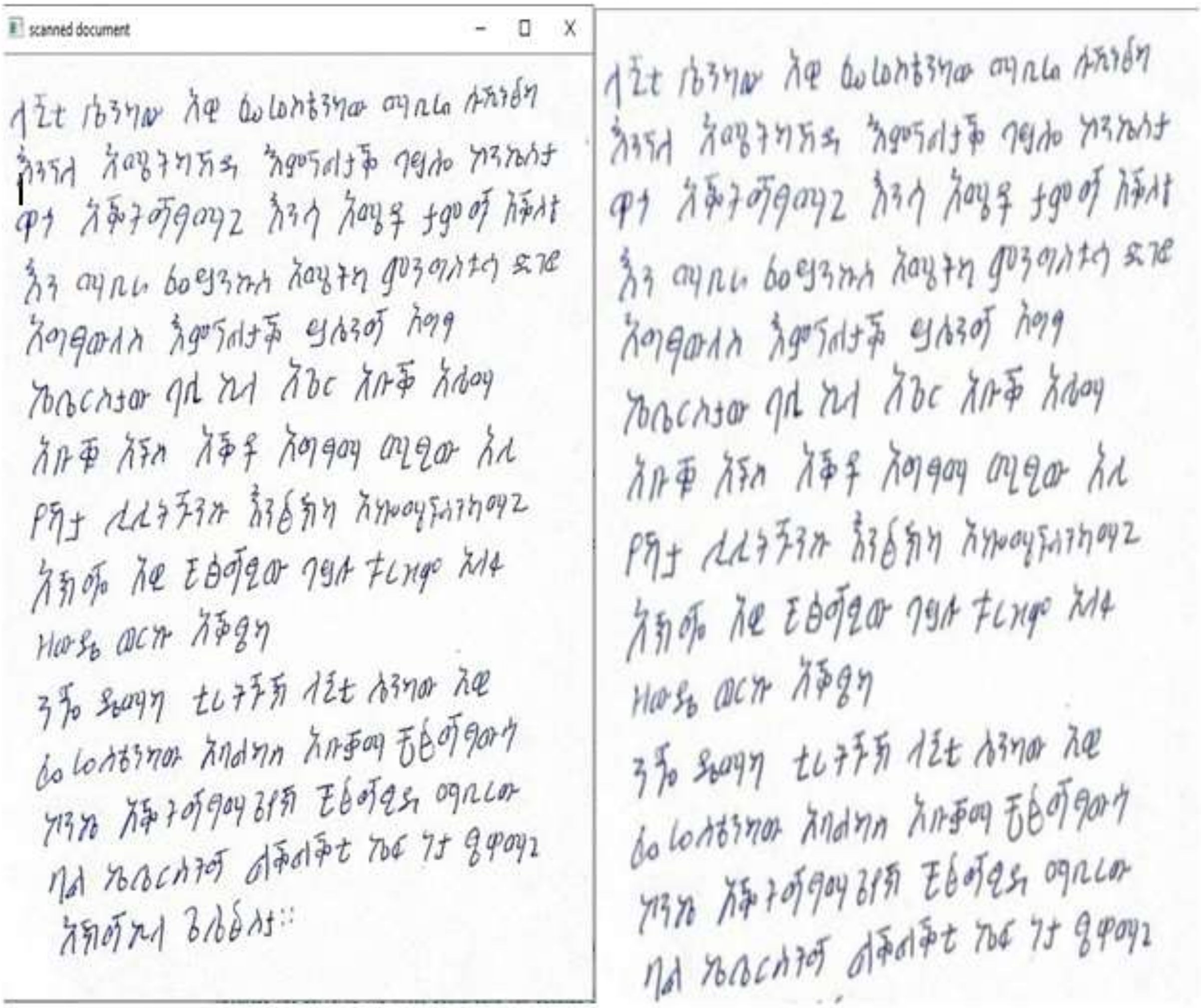


Figure 4. 1 (a) Awngi document (b) Document after noise removal

4.3.2. Grayscale conversion result

Then the next step is the conversion of noise-removed images into grayscale images because to reduce the computational cost of the training process image should be converted from a three-channel image to one channel. To do this, we have used the Open CV method name

cv2.cvtcolor with parameter cv2.COLOR_BGR2GRAY python built-in function used to convert colored image to grayscale image. The conversion result as shown in figure firstly dataset collected from participants was scanned using Samsung galaxy mobile scanner app then Image with RGB from three-channel color was converted to a grayscale image conversion method is shown in Figure 4.2(a) and 4.2(b)

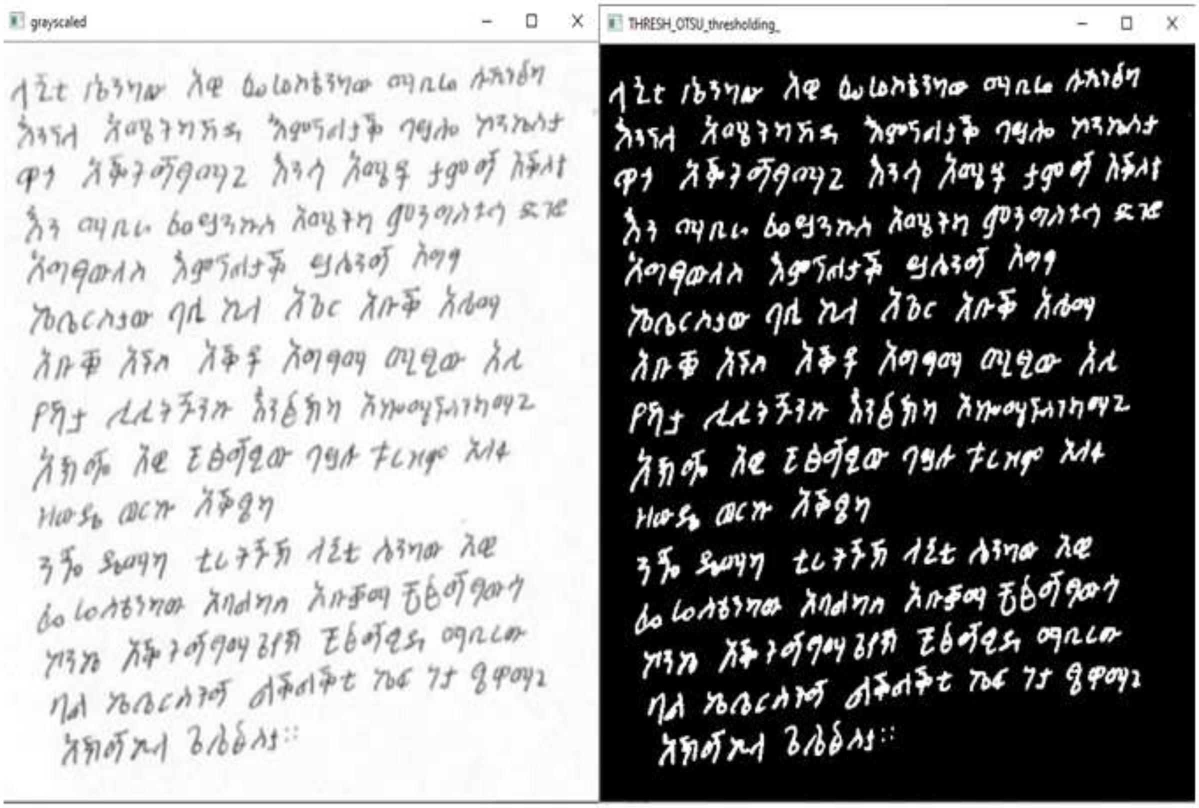


Figure 4. 2 (a) grayscaled Awngi document (b) threshold Document after grayscaled

4.3.3. Binarization result

After the grayscale change and noise decline or reduce stages are done the image is passed to the binarization we have used Otsu threshold binarization methods the image is binarized. By using OpenCV techniques named cv2.threshold with parameters specifying the binarized image is inverted. threshold = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1] as we understand from the code after the document image is denoised next step is binarization separating background from foreground by using threshold value range from 0,255. The results of the Otsu binarization method is shown in the following Figure 4.2(b)

4.3.4. Morphological transformations Result

We have implemented morphological operation on most image documents. These operations are opening and closing using the methods provided by the OpenCV API. Whereas opening is done through apply erosion first and the dilation next also, the reverse is for closing first dilation then erosion. Some of the results of these morphological operations are shown in bellows in Figure 4.3

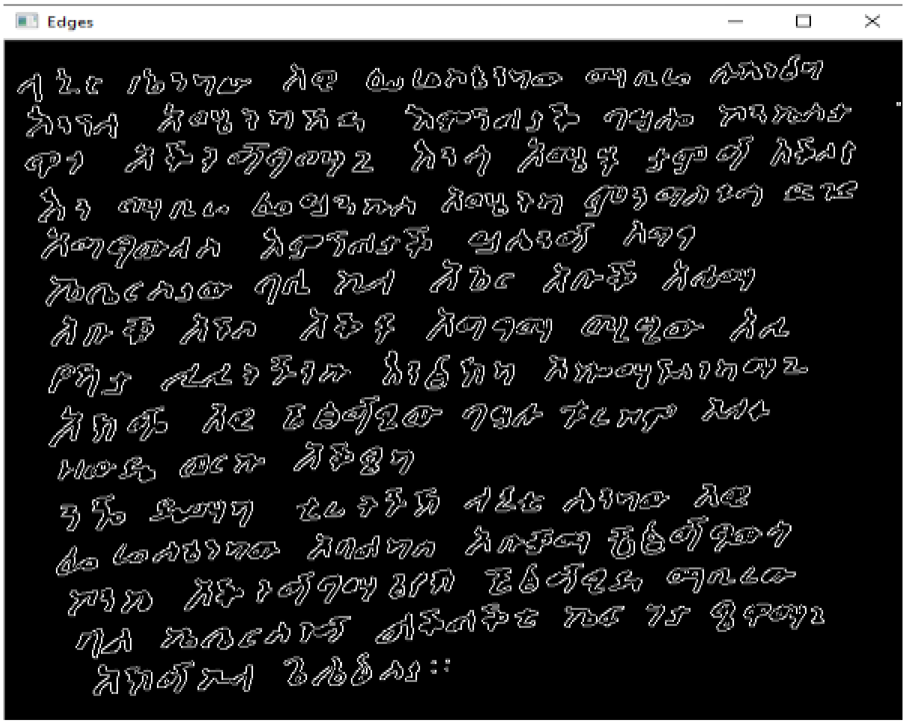


Figure 4. 3 morphological transformation results

As we have seen from the figure 4.3 we have performed morphological transformation

in some case when documents with there a higher degree of noise some of the preprocessing methods have failed. Some of the faults that have occurred during the preprocessing stage of the proposed system.

4.3.5. Segmentation result

Segmentation error happens when if there is an overlapping of characters is found in a written document. To measure character segmentation, the technique is based on a model of human vision measure for the segmentation error of characters is proposed using knowledge-based segmentation procedure is implemented. if we feed the handwritten image document with several 182 characters segmented characters are saved in the folder for feature extraction then the accuracy of characters segmented correctly will be calculated

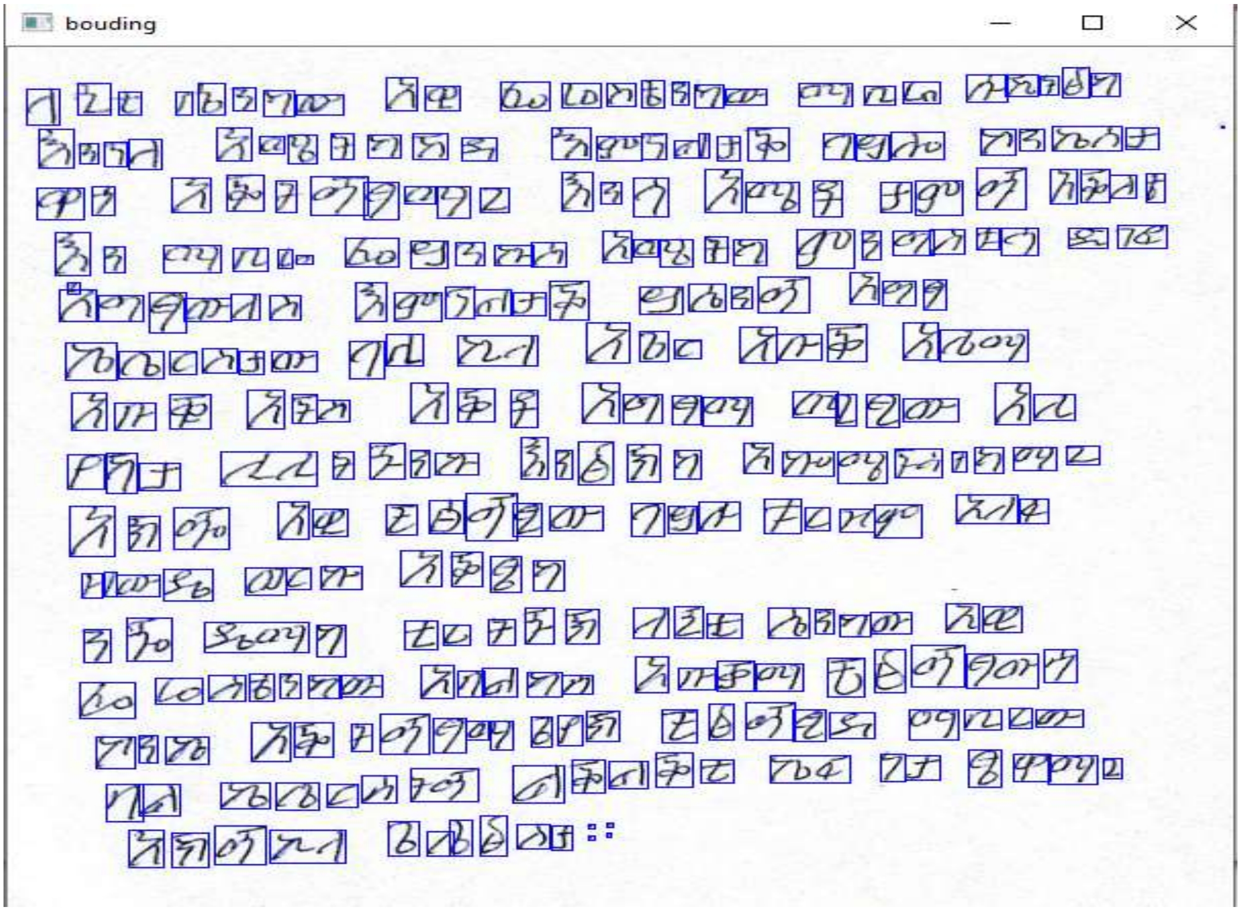
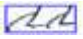


Figure 4. 4 bounding box result

In proposed offline handwritten Awigni character segmentation to segment characters from a given document accurately we have applied stepwise segmentation sentence, word, character After dilation and erosion to join gap of character find Contours and for each counter finds the bounding Rect The size of character differs from one person to another and also from time to time even when the person is the same .therefore we have normalized with the size character 28 x 28. The algorithm was tested on more than 28,210 characters written by different writers with varied writing styles, out of which almost 27,450 characters were segmented accurately, thereby, giving an accuracy of approximately 97.3%. Segmentation error = (760) or 3%.

Segmentation error analysis Two types exist the first type of segmentation error is considering one character as one or more characters. This type of error exists in degraded documents where primitive structures of a character are unconnected and in this study no such error was seen. The second type of segmentation error is considering two connected characters as one character. Example  connected character segmented or bounded as one character.

4.3.6. Segmentation of isolated character result

As we have depicted from figure 4.5 preprocessing of isolated Awngi handwritten characters using Gaussian blur for noise reduction method accepts the handwritten image and reduces the noise from the image. OpenCV method cv2.GaussianBlur is used because we have to deal with degraded character images

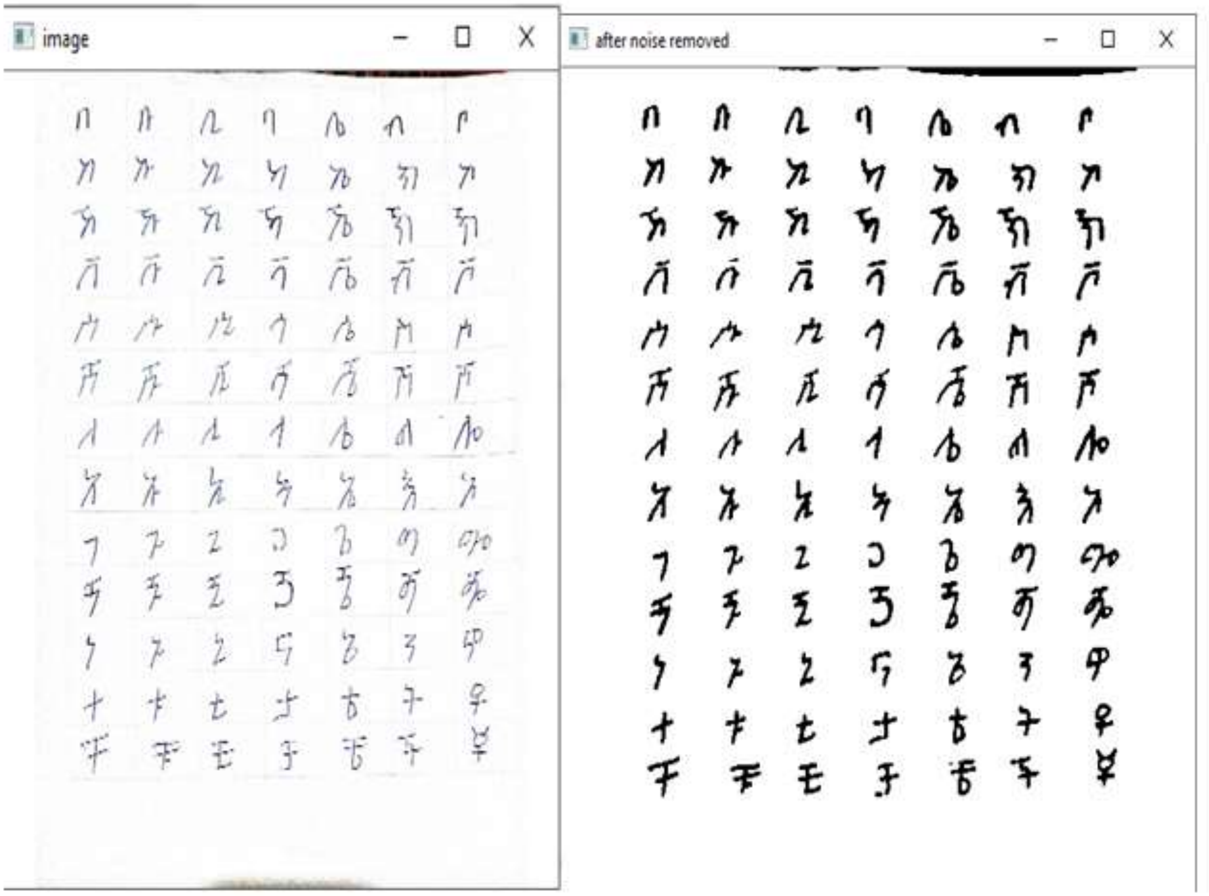


Figure 4. 5 Awngi handwritten isolated character before and after noise removal

As we depicted after noise is removed binarization is performed on character binarization is separating the background from the foreground. To binarized image characters we have used the Otsu binarization technique. We have used the OpenCV method named cv2.threshold() with its specifying as stated in many scholars recommend Otsu binarization segmentation for character recognition systems.

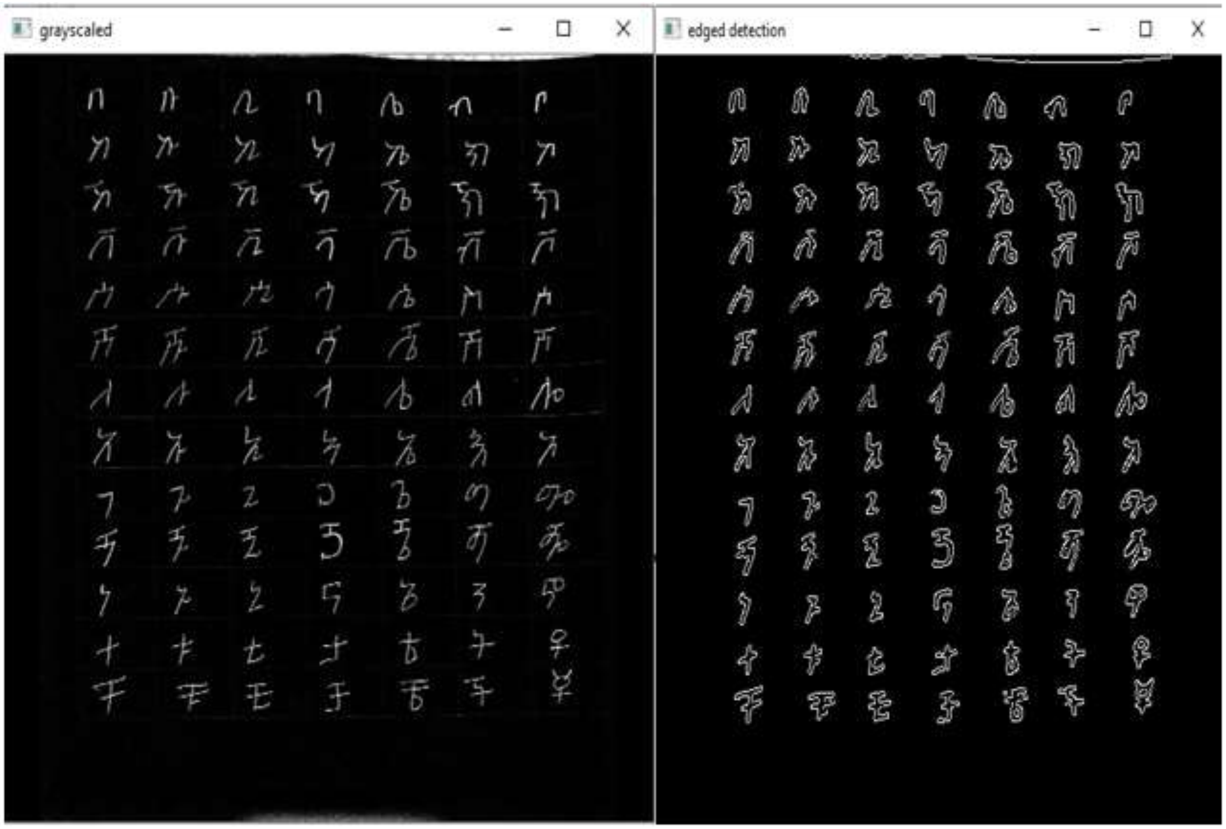


Figure 4. 6 (a)isolated character grayscale image (b)Edge detection

In our handwritten Awngi character, recognition data was morphologically transformed because of that the effects of the morphological transformation methods are clear and distinguishable.

We have applied efficient preprocessing methods for each phase and we have got good classification accuracy. But, in some conditions when there is a higher degree of noise on the documents reduces the recognition process. Some of the errors that have happened during the preprocessing phase of the proposed system.

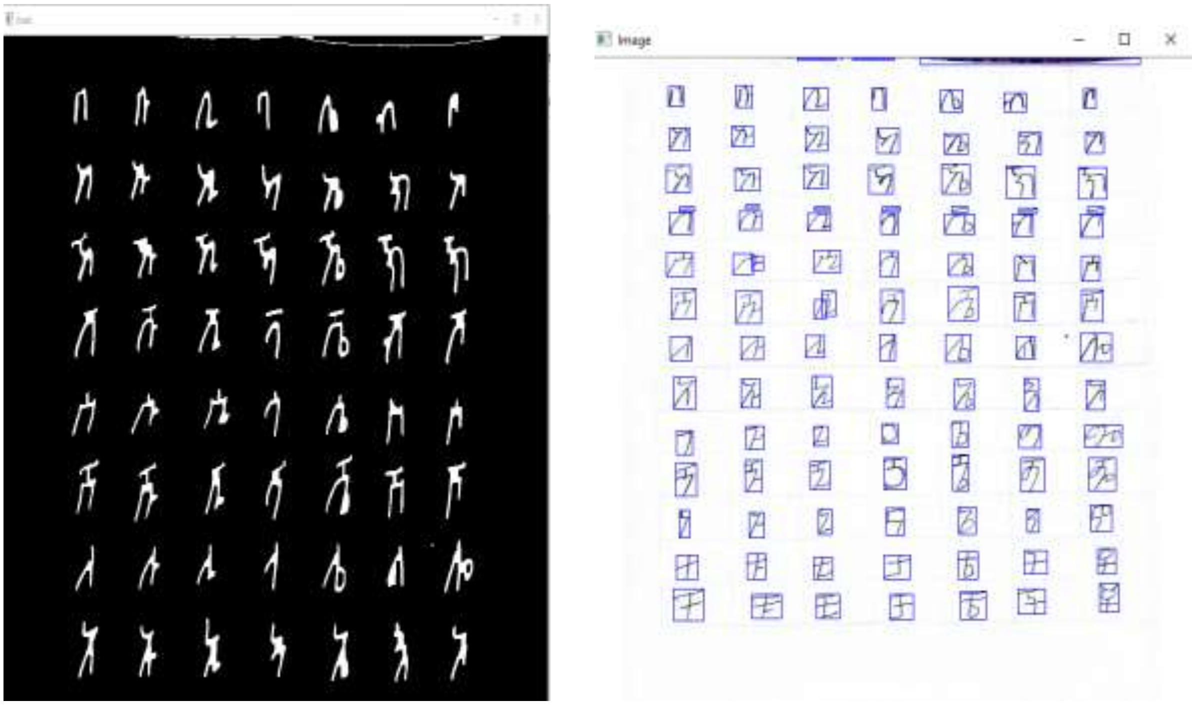


Figure 4. 7 closed characters   character level bounding box

As we depicted below figure 4.7 as a result of bounding box characters are normalized into equal same size 28 x28 pixel image because to reduce computational cost to resize character of the image we have used Opencv method resize(28,28,1)

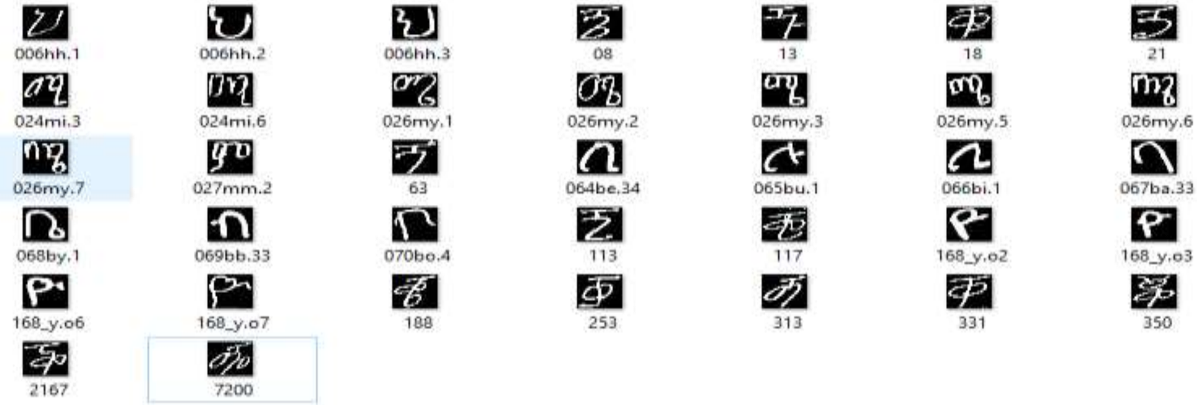


Figure 4. 8 Sample of segmented character used for training

4.4. Results of Training and Classification

A prepared dataset containing 30,115 binary images which are centered and normalized to a size of 28*28 is used in each class we have to train and test the prepared convolutional neural network. Out of the total dataset images,80% is used for training, 20% used for testing, model the designed neural network architecture. The result of the training and testing of the proposed convolutional neural network architecture is presented in the following sections. The proposed neural network system is trained with the prepared dataset according to the specifications Presented. In the proposed model we have achieved a training accuracy of 96.6% and 92.6% of testing accuracy of the proposed model.

The Convolutional neural network and Convolutional neural network architecture of the proposed model are trained. The network is trained with different epoch values and the batch size is 128 with the optimizer function of Adam SGD and RMSprop. From the all comparison of optimizer for our proposed model the Adam is the best gradient descent optimization technique compared with stochastic gradient descent and RMSprop optimizer. The other hand parameter is optimizer selection which takes an effect on the performance of our model. the main function of Optimizers is used to update the weight of the network and they have their behaviors. For our proposed model the adam optimizer gives better outcomes or results when we compared it with RMSprop, stochastic gradient descent, and degrade optimizer. Table 4.2 demonstrates the effect.

Table 4. 2 Comparative optimizer selection effect on accuracy

Optimizer	Training acc	Validation acc	Training loss	Validation loss
Adam	0.96	0.916	0.08	1.31
RMSprop	0.95	0.891	0.091	1.5
SGD	0.7	0.63	1.31	1.42
ADAGRAD	0.64	0.58	1.50	1.7

We trained the model and evaluate the optimization technique in terms of accuracy and loss function using our dataset Adam optimizer gives optimal results for our proposed model.

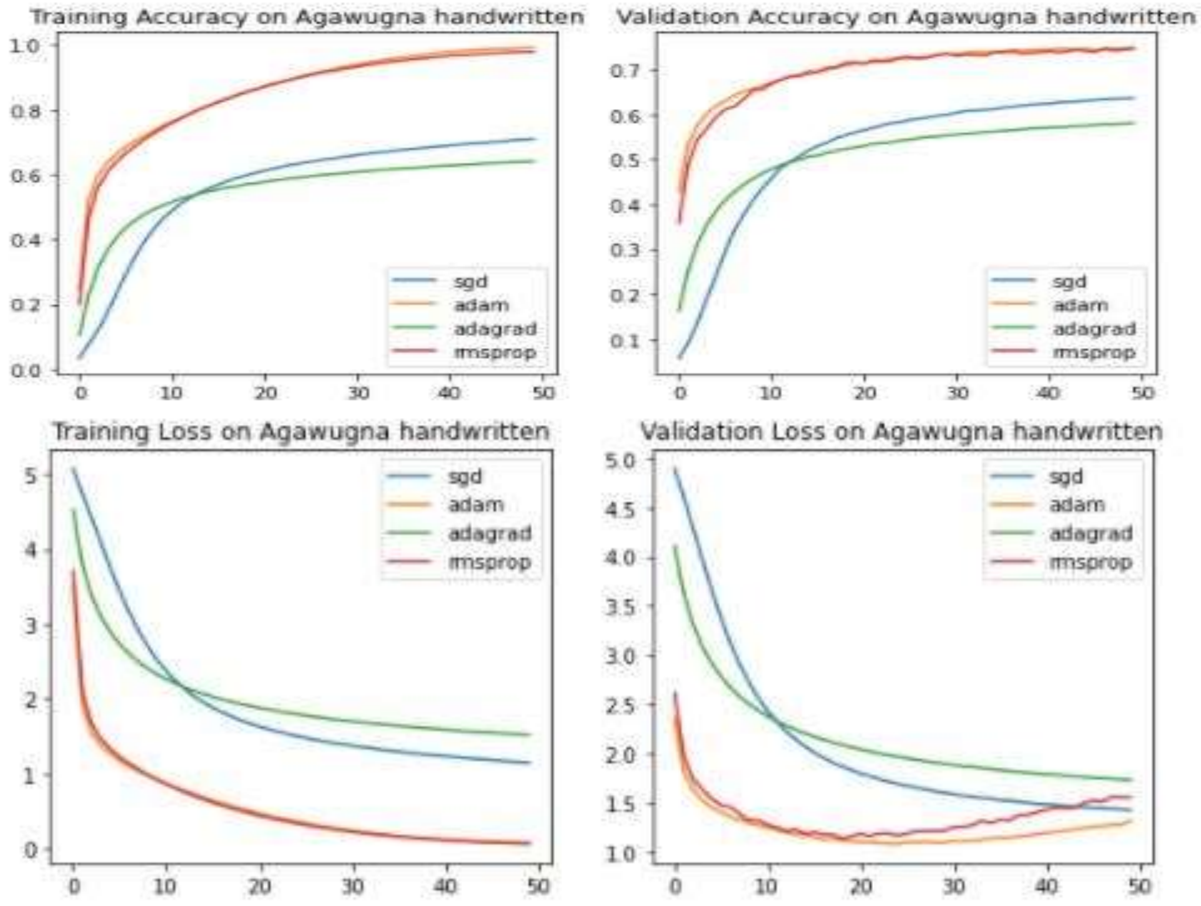
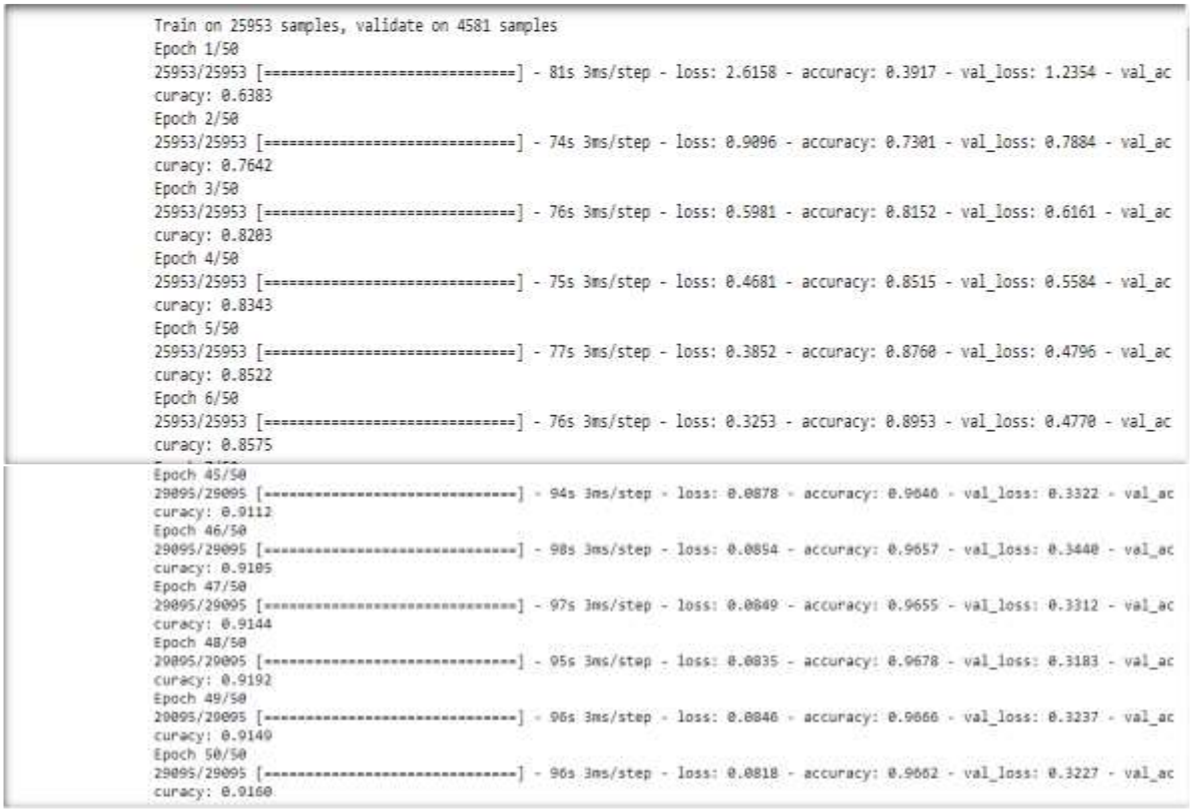


Figure 4. 9 comparative analysis of training and validation loss accuracy optimizers

Based on our experiment we performed or computed comparative evolution of the Four most commonly used optimization techniques in our convolutional neural network (ConvNet) architecture. The investigated techniques are SGD, adam, adagrad, RMSprop we trained the model and evaluate the optimization technique in terms of accuracy and loss function using our dataset Awngi handwritten characters dataset overall experimental results show Adam optimizer achieved better performance across other optimization techniques.



```
Train on 25953 samples, validate on 4581 samples
Epoch 1/50
25953/25953 [==============================] - 81s 3ms/step - loss: 2.6158 - accuracy: 0.3917 - val_loss: 1.2354 - val_ac
curacy: 0.6383
Epoch 2/50
25953/25953 [==============================] - 74s 3ms/step - loss: 0.9096 - accuracy: 0.7301 - val_loss: 0.7884 - val_ac
curacy: 0.7642
Epoch 3/50
25953/25953 [==============================] - 76s 3ms/step - loss: 0.5981 - accuracy: 0.8152 - val_loss: 0.6161 - val_ac
curacy: 0.8203
Epoch 4/50
25953/25953 [==============================] - 75s 3ms/step - loss: 0.4681 - accuracy: 0.8515 - val_loss: 0.5584 - val_ac
curacy: 0.8343
Epoch 5/50
25953/25953 [==============================] - 77s 3ms/step - loss: 0.3852 - accuracy: 0.8760 - val_loss: 0.4796 - val_ac
curacy: 0.8522
Epoch 6/50
25953/25953 [==============================] - 76s 3ms/step - loss: 0.3253 - accuracy: 0.8953 - val_loss: 0.4770 - val_ac
curacy: 0.8575
Epoch 45/50
29095/29095 [==============================] - 94s 3ms/step - loss: 0.0878 - accuracy: 0.9646 - val_loss: 0.3322 - val_ac
curacy: 0.9112
Epoch 46/50
29095/29095 [==============================] - 98s 3ms/step - loss: 0.0854 - accuracy: 0.9657 - val_loss: 0.3440 - val_ac
curacy: 0.9105
Epoch 47/50
29095/29095 [==============================] - 97s 3ms/step - loss: 0.0849 - accuracy: 0.9655 - val_loss: 0.3312 - val_ac
curacy: 0.9144
Epoch 48/50
29095/29095 [==============================] - 95s 3ms/step - loss: 0.0835 - accuracy: 0.9678 - val_loss: 0.3183 - val_ac
curacy: 0.9192
Epoch 49/50
29095/29095 [==============================] - 96s 3ms/step - loss: 0.0846 - accuracy: 0.9666 - val_loss: 0.3237 - val_ac
curacy: 0.9149
Epoch 50/50
29095/29095 [==============================] - 96s 3ms/step - loss: 0.0818 - accuracy: 0.9662 - val_loss: 0.3227 - val_ac
curacy: 0.9160
```

Figure 4. 10 Awngi handwritten model training process

As we have depicted from the above figure 4.10 Awngi handwritten model training on 25953 samples and validating the training process by 4581 samples as we have seen from the above figure 4.10 the training starts from epoch 1/50 with the training loss of 2.61 and training accuracy 0.391 and validation loss of 1.23 validation training 0.63 so at this time there is over fitting means validation training is greater than validation training then next model train well in the epoch of 6/50 validation training is less than validation training from the epoch starting from the epoch 15/50 model goes with constant direction as shown in the graph

4.5. Visualizing Accuracy and Losses of model

As clearly depicted in figure 4.11 and figure 12 shows the training accuracy and training loss with 50 epochs. The training accuracy is 96.6% and the validation accuracy of 91.6%. a training loss decreases from 2.61 to 0.08 and validation loss decreases from 1.23 to 0.32

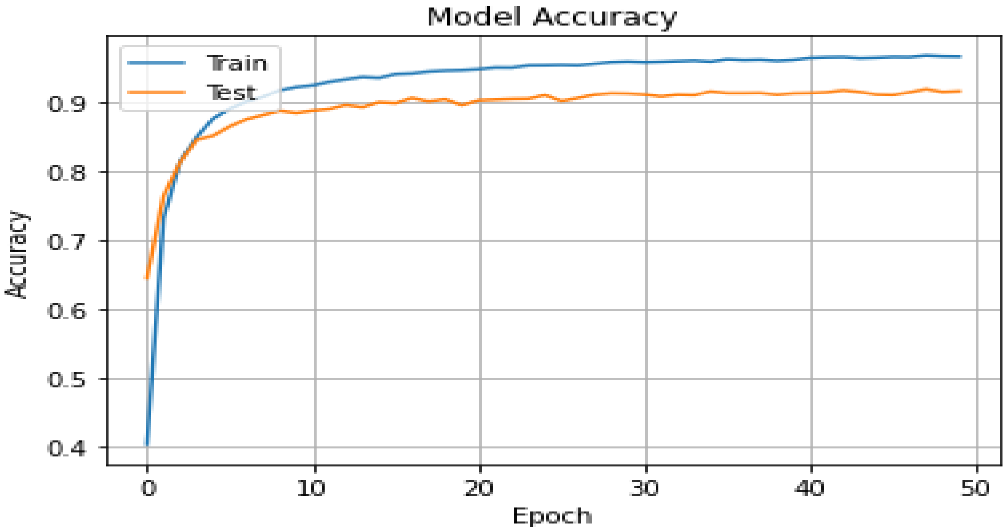


Figure 4. 11 Training and testing accuracy of the proposed model with the epoch of 50

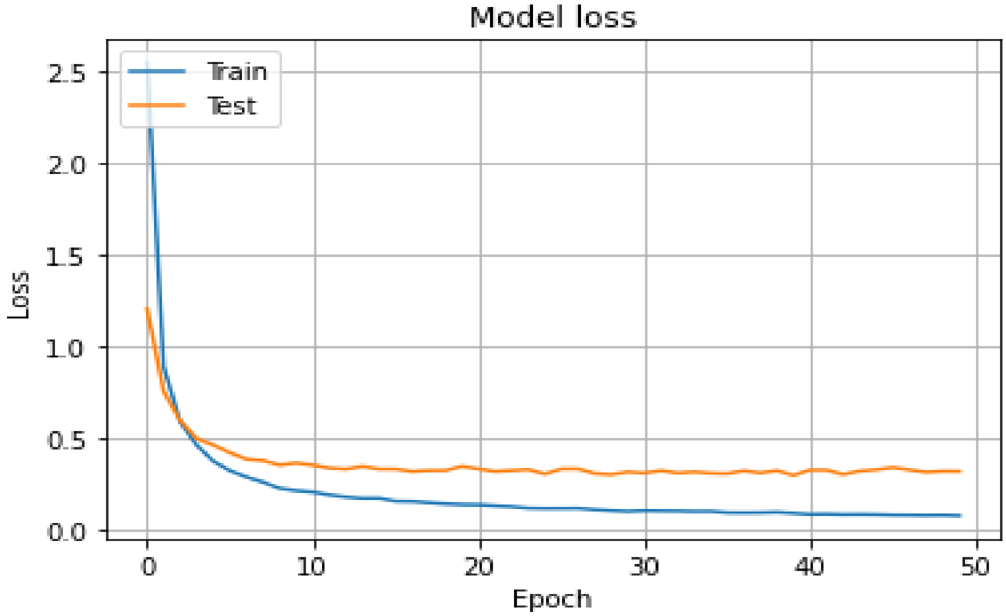
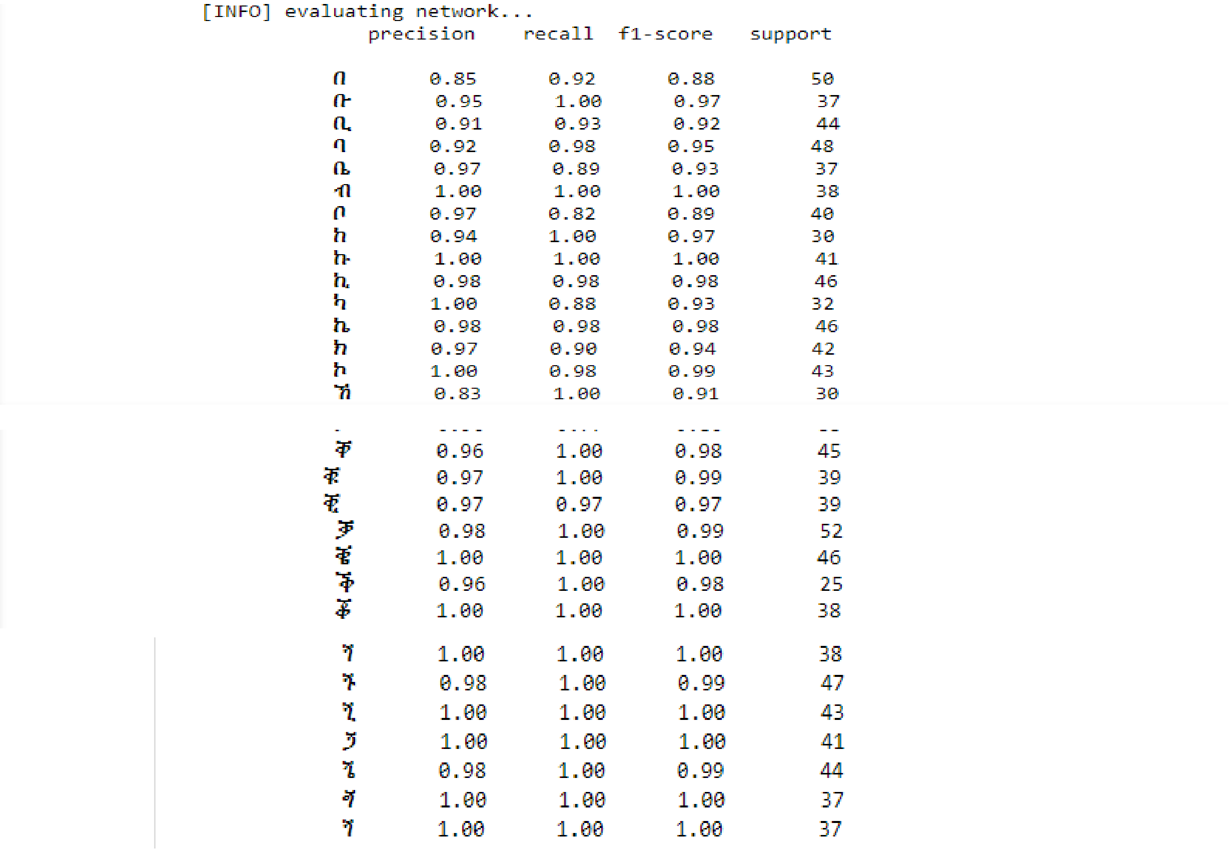


Figure 4. 12 Training and testing loss of proposed model with the epoch of 50

To evaluate the model, testing datasets were fed into the developed model., the model was evaluated by comparing its output against the observed data using precision, recall, and f1-score values for evaluating recognition accuracy .based on the experiment we have computed classification report the precision, recall, and f1-score for all characters as we have shown in the figure 4.13 for more details see Appendix D.



```
[INFO] evaluating network...
              precision    recall  f1-score   support

           በ       0.85      0.92      0.88        50
           ቡ        0.95      1.00      0.97        37
           ቢ        0.91      0.93      0.92        44
           ባ       0.92      0.98      0.95        48
           ቤ        0.97      0.89      0.93        37
           ብ        1.00      1.00      1.00        38
           ቦ       0.97      0.82      0.89        40
           ከ       0.94      1.00      0.97        30
           ኩ        1.00      1.00      1.00        41
           ኪ        0.98      0.98      0.98        46
           ካ       1.00      0.88      0.93        32
           ኬ        0.98      0.98      0.98        46
           ክ       0.97      0.90      0.94        42
           ኮ       1.00      0.98      0.99        43
           ኸ        0.83      1.00      0.91        30

           .        ----      ----      ----        --
           ቸ        0.96      1.00      0.98        45
          ቹ         0.97      1.00      0.99        39
          ቺ         0.97      0.97      0.97        39
           ቻ        0.98      1.00      0.99        52
           ቼ        1.00      1.00      1.00        46
           ች        0.96      1.00      0.98        25
           ቾ        1.00      1.00      1.00        38

           ኘ        1.00      1.00      1.00        38
           ኙ        0.98      1.00      0.99        47
           ኚ        1.00      1.00      1.00        43
           ኛ        1.00      1.00      1.00        41
           ኜ        0.98      1.00      0.99        44
           ኝ        1.00      1.00      1.00        37
           ኘ        1.00      1.00      1.00        37
```

Figure 4. 13 Sample (precision, recall, and f1-score) of the proposed model

4.6. Dataset Analysis and Results

In this part, a detailed description and analysis of the datasets are presented, and then the character recognition results obtained during the experiment are presented. As depicted in Figure 4.14, we saw that some of the characters are wrongly predicted with the ground truth because of their nature they have similar shapes and structures and some of the characters have similar shapes and a structure due to people's writing habits. Awngi handwritten characters, depicted in Figure 4.14 and 4.15, are the factors causing prediction or recognition errors

As we have mentioned above in addition to test data we are evaluated our model by external or unseen data here is an Awngi handwritten document is scanned and feed to our model the result is showed below

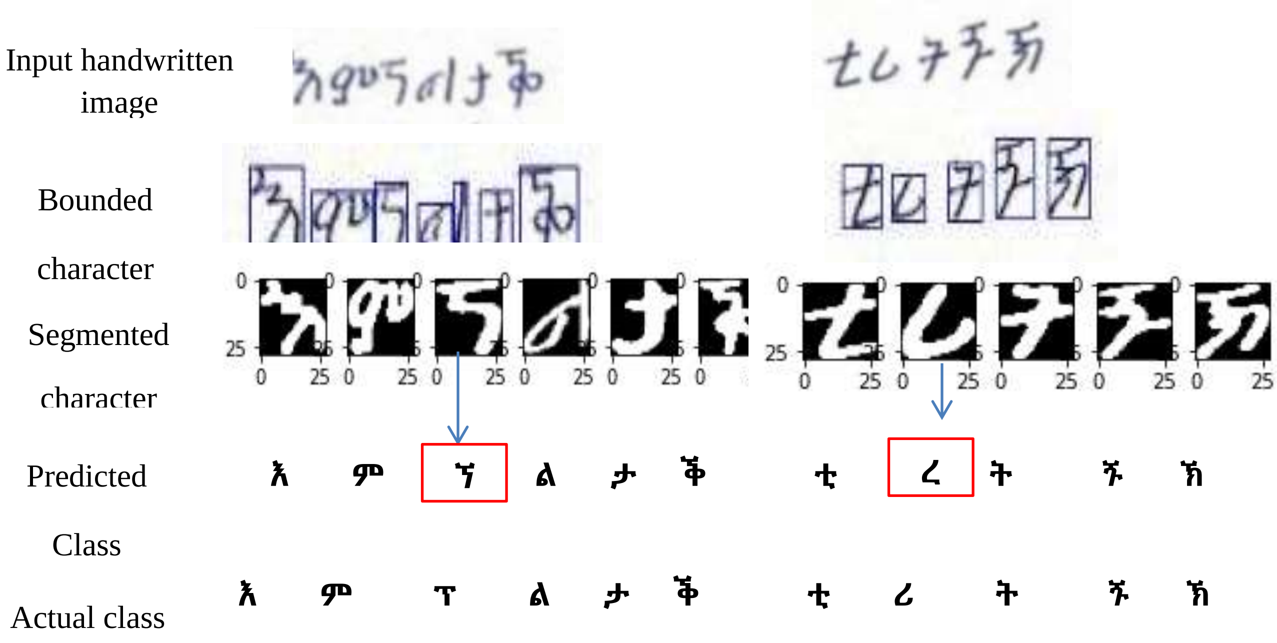


Figure 4. 14 Awngi sample_1 handwritten character recognition

As we have seen from sample 1 Figure 4.14 Awngi handwritten text segmented into characters bound with red color is misrecognized characters ፐ and ሪ miss recognition or classified as ኘ, ረ respectively. two characters are misrecognized from a total of 11 chars

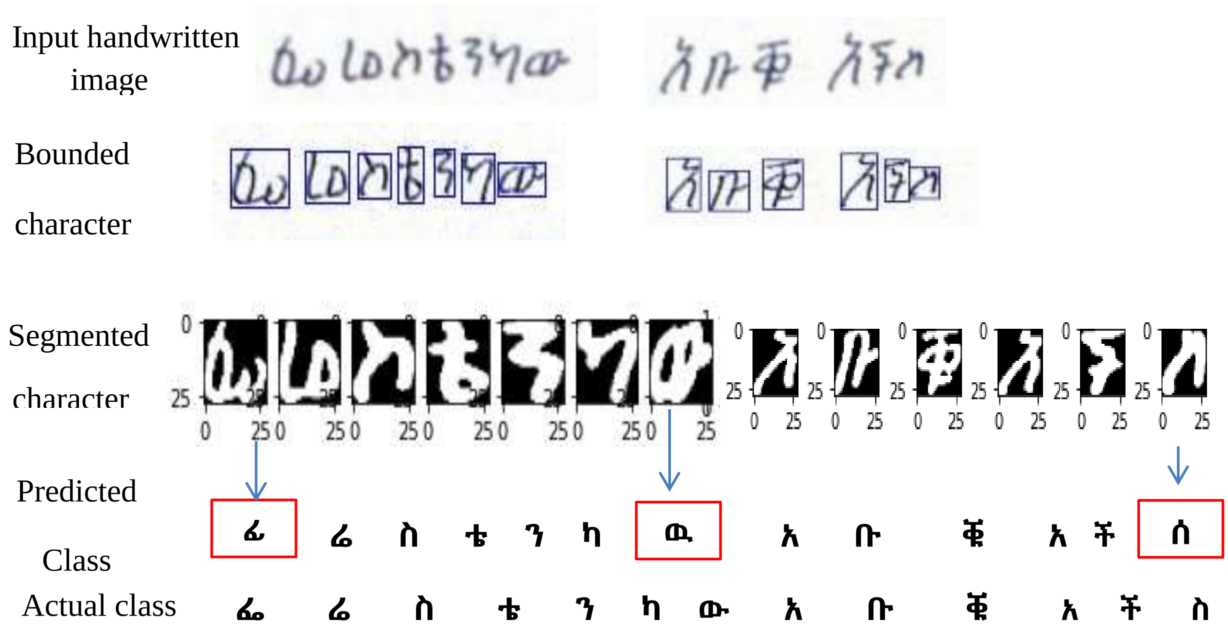


Figure 4. 15 Awngi sample_2 handwritten character recognition

As we have seen from sample 2 Figure 4.15 Awngi handwritten text segmented into characters bound with red color is misrecognized characters ፌ And ጢ miss recognition or classified as ፊ , ጡ(second level character of ጠ) respectively. From 4.15 characters totally three of characters are miss recognized as similar shapes This happens due to the following reasons:

1) They have similar shapes and structures in their nature. For example, the left side strock is hard to recognize.

2) Some of the characters (scripts) have similar shapes and a structure due to people's writing habits. Even humans cannot distinguish them without their labels. Even though every row represents different handwriting samples, the samples in every pair of the rows look very close to each other.

3) The degraded quality of the images, such as missing strokes, additional noises, rotated characters also have an impact on the classification. The cases could be happening by people's poor handwriting, or happen by the scanning procedure, when size normalization, and improper segmentation. For the third error class, it is extremely difficult even for a machine to make a correct prediction with such ambiguous character and may quality low inputs. Especially in this research paper, as many of the scripts have similarities, when a rotation is used, some of the scripts have a high probability to look like another script from their orders

Awngi handwritten characters which have a visual similarity of shape have given low result. Characters, such as ፈ, ፉ, ፊ, ፋ, ፌ, ፍ, ፎ, ጢ, ጡ, ፐ, ፒ, ፓ, ሸ, ሸ, ም, ሞ, ረ, ሩ, ሪ, ራ, ሬ, ር, ሮ, ዩ, ዪ, ዬ, ይ and ዯ are produce low accuracy than other characters. These characters are difficult to recognize even for human beings. Awngi characters are large in number to represent using graphically and using confusion matrix as we have mentioned above we have 182 class it is difficult to display 182 class in the python window

As we have explained above to evaluate the performance of the new model, accuracy is not enough. Performance parameters like confusion matrix and classification report are also visualized. Shows the detailed output of the classification report. For all output, parameters see Appendix D.

CHAPTER FIVE

5. CONCLUSION AND FUTURE WORK

5.1. Introduction

In this chapter, we present our conclusion drawn from the findings of our experiment and the comparisons we made. A recommendation is included for further improvements of handwritten character recognition. Our future work following this is discussed and how we plan to achieve it.

5.2. Conclusion

Handwritten character recognition is the most challenging task in pattern recognition. Since the handwriting of one person is different from another person. In addition to the variability of Different writing styles, in Awngi characters, there is a visual similarity of shape with minor Change (i.e. with the presences of the special appendage of line, loop, and a dot below, above, left, And the right of the character), which makes the recognition task difficult.

We have reviewed Papers related to our study, and we identified the problem of the system and its challenge. We discussed the different techniques like image preprocessing, feature extraction, and Classification, Even though, for the proposed model we have prepared a dataset that is used for training and testing the proposed model. Hence, there isn’t a dataset that can be used for Awngi recognition systems, it was a must to organize the datasets we collect dataset from scratch. The designed model was trained and tested with the handwritten character images of the 26 base and its labels of Awngi handwritten characters.

Experimental results point to that the proposed Awngi handwritten character recognition model is a promising optimal classification method in the Handwriting recognition field due to the. The features of the handwritten character images can be automatically extracted by the model, whereas the achievement of traditional classifiers trusts largely on the retrieval of the good hand-crafted based features extractor that is a tedious and time-consuming task. The model combines the qualities of CNN and classifier; algorithms are the successful classifiers in the Awngi handwritten character recognition.

5.3. Contribution of this thesis

As a contribution to the scientific world or knowledge, the proposed model offered a systematic approach in handwritten Awngi character recognition. Considering the input to the model, we provide a handwritten image character then preprocessing technique that has high capability in segmentation and recognition. We also propose a segmentation algorithm that is capable of discriminating the Considering the proposed architecture, we propose a convolution and pooling module that weighs less, train faster and classify more accurately we compute the comparison the most four used optimizers in the training process.

As Awngi handwritten character recognition has not been studied yet, the main contributions of this study are the following: (1) we were collected and ordered 182 Awngi characters in their order. (2) As there was no existing offline (handwritten) dataset for Awngi scripts, a we prepared new dataset For the first time from the scratch with a size of 28 x 28 pixels. we have made it publicly available for another researcher (3) To extract automatic features from handwritten images, we propose a CNN-based feature extractor module. (4) To reduce the computational cost, we adjust the images to a smaller size. we made a comparative analysis of the most four outstanding optimizers based on the performance of the optimizer in our dataset we have selected experimentally adam as a training optimization method.

5.4. Recommendation

As future work, The model performance can be additionally improved through fine-tuning of its model structure and the parameters. Improvements can be made based on the input layers, size of the number of the feature maps through layers, fine-tuning kernel functions used in the model. it also can be further studied on recognition of all non-Latin handwritten characters, in different languages. As most valuable Ethiopian manuscripts have been written by non-Latin scripts, the manuscripts are being exposed to, fire, theft, and. To defend or protect the preservation of the documents and the history of Ethiopia from injury, document digitization should be done. This thesis plans to develop an OCR application

In character recognition models are followed by some kinds of post-processing approaches for enhancing the results of classification. These techniques can be dictionary-based or natural language processing (NLP) based. However, the proposed system doesn’t include .we strongly recommend future researchers who have a research interest in handwritten recognition include post preprocess technique

Reference

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Devin, M. J. a. p. a. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Acharya, S., Pant, A. K., & Gyawali, P. K. (2015). *Deep learning based large scale handwritten Devanagari character recognition.* Paper presented at the 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA).

Adrian, R. D. (2017). Deep Learning for Computer Vision with Python. In: PyImageSearch.

Afroge, S., Ahmed, B., & Hossain, A. (2017). *Bangla optical character recognition through segmentation using curvature distance and multilayer perceptron algorithm.* Paper presented at the 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE).

Aggarwal, A., & Singh, K. (2015). *Handwritten gurmukhi character recognition.* Paper presented at the 2015 International Conference on Computer, Communication and Control (IC4).

Akal, O., Mukherjee, T., Barbu, A., Paquet, J., George, K., & Pasiliao, E. (2018). *A distributed sensing approach for single platform image-based localization.* Paper presented at the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA).

Arcelli, C., Di Baja, G. S. J. I. T. o. P. A., & Intelligence, M. (1985). A width-independent fast thinning algorithm. (4), 463-474.

Assabie, Y., & Bigun, J. (2009). *HMM-based handwritten amharic word recognition with feature concatenation.* Paper presented at the 2009 10th International Conference on Document Analysis and Recognition.

Bautista, R. M. J. S., Navata, V. J. L., Ng, A. H., Santos, M. T. S., Albao, J. D., & Roxas, E. A. (2015). *Recognition of handwritten alphanumeric characters using Projection Histogram and Support Vector Machine.* Paper presented at the 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).

Belay, B., Habtegebrial, T., Liwicki, M., Belay, G., & Stricker, D. (2019). *Factored convolutional neural network for amharic character image recognition.* Paper presented at the 2019 IEEE International Conference on Image Processing (ICIP).

Belay, B. H., Habtegebrial, T. A., & Stricker, D. (2018). *Amharic character image recognition.* Paper presented at the 2018 IEEE 18th International Conference on Communication Technology (ICCT).

Birhanu, A. T., Sethuraman, R. J. I. J. o. S., Engineering, & Research, T. (2015). Artificial neural network approach to the development of OCR for real life Amharic documents. *4*(1), 141-147.

Buades, A., Coll, B., & Morel, J.-M. J. I. P. O. L. (2011). Non-local means denoising. *1*, 208-212.

Cheriet, M., Kharma, N., Liu, C.-L., & Suen, C. (2007). *Character recognition systems: a guide for students and practitioners*: John Wiley & Sons.

Choudhary, A. J. I. J. I. C. T. (2014). A review of various character segmentation techniques for cursive handwritten words recognition. *4*(6), 559-564.

Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011). *Convolutional neural network committees for handwritten character classification.* Paper presented at the 2011 International Conference on Document Analysis and Recognition.

Demilew, F. A., & Sekeroglu, B. J. S. A. S. (2019). Ancient Geez script recognition using deep learning. *1*(11), 1315.

Desai, S., & Singh, A. G. (2016). *Optical character recognition using template matching and back propagation algorithm.*

Elleuch, M., Tagougui, N., & Kherallah, M. (2015). *Arabic handwritten characters recognition using deep belief neural networks.* Paper presented at the 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15).

Fitsum, K. T., & Patel, Y. (2018). *Optical Character Recogniation for Tigrigna Printed Documents Using HOG and SVM.* Paper presented at the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).

Gulli, A., & Pal, S. (2017). *Deep learning with Keras*: Packt Publishing Ltd.

Hope, T., Resheff, Y. S., & Lieder, I. (2017). *Learning tensorflow: A guide to building deep learning systems*: " O'Reilly Media, Inc.".

Hussien, R. S., Elkhidir, A. A., & Elnourani, M. G. (2015). *Optical character recognition of arabic handwritten characters using neural network.* Paper presented at the 2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE).

Josh, P., & Adam, G. (2017). Deep Learning: A Practitioner's Approach. In: Sebastopol: O'Reilly.

Kaleka, N. K., & Verma, K. G. (2016). *Segmentation of lā-consonant and duláwā-consonant combination strokes in online handwritten Gurmukhi script recognition.*

Katiyar, G., & Mehfuz, S. (2015). *SVM based off-line handwritten digit recognition.* Paper presented at the 2015 Annual IEEE India Conference (INDICON).

Khan, A. R., & Mohammad, Z. J. I. J. o. I. P. (2008). A simple segmentation approach for unconstrained cursive handwritten words in conjunction with the neural network. *2*(3), 29-35.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. J. C. o. t. A. (2017). Imagenet classification with deep convolutional neural networks. *60*(6), 84-90.

Malanker, A., & Patel, P. J. I. J. E. E. E. (2014). Handwritten Devanagari script recognition: a survey. *9*(2), 80-87.

Meshesha, M., & Jawahar, C. (2005). *Recognition of printed Amharic documents.* Paper presented at the Eighth International Conference on Document Analysis and Recognition (ICDAR'05).

Nair, P. P., James, A., & Saravanan, C. (2017). *Malayalam handwritten character recognition using convolutional neural network.* Paper presented at the 2017 International conference on inventive communication and computational technologies (ICICCT).

Nakkach, H., Hichri, S., Haboubi, S., & Amiri, H. (2016). *Hybrid approach to features extraction for online arabic character recognition.* Paper presented at the 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV).

Nasien, D., Haron, H., & Yuhaniz, S. S. (2010). *Support Vector Machine (SVM) for English handwritten character recognition.* Paper presented at the 2010 Second International Conference on Computer Engineering and Applications.

Pal, U., Wakabayashi, T., & Kimura, F. (2007). *Handwritten Bangla compound character recognition using gradient feature.* Paper presented at the 10th international conference on information technology (ICIT 2007).

Poornima, M., Kavitha, M. S., & Sornam, M. (2016). *An efficient morlet function based adaptive method for faster backpropagation for handwritten character recognition.* Paper presented at the 2016 IEEE International Conference on Advances in Computer Applications (ICACA).

Ramsundar, B., & Zadeh, R. B. (2018). *TensorFlow for deep learning: from linear regression to reinforcement learning*: " O'Reilly Media, Inc.".

Reta, B. Y., Rana, D., & Bhalerao, G. V. (2018). *Amharic handwritten character recognition using combined features and support vector machine.* Paper presented at the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI).

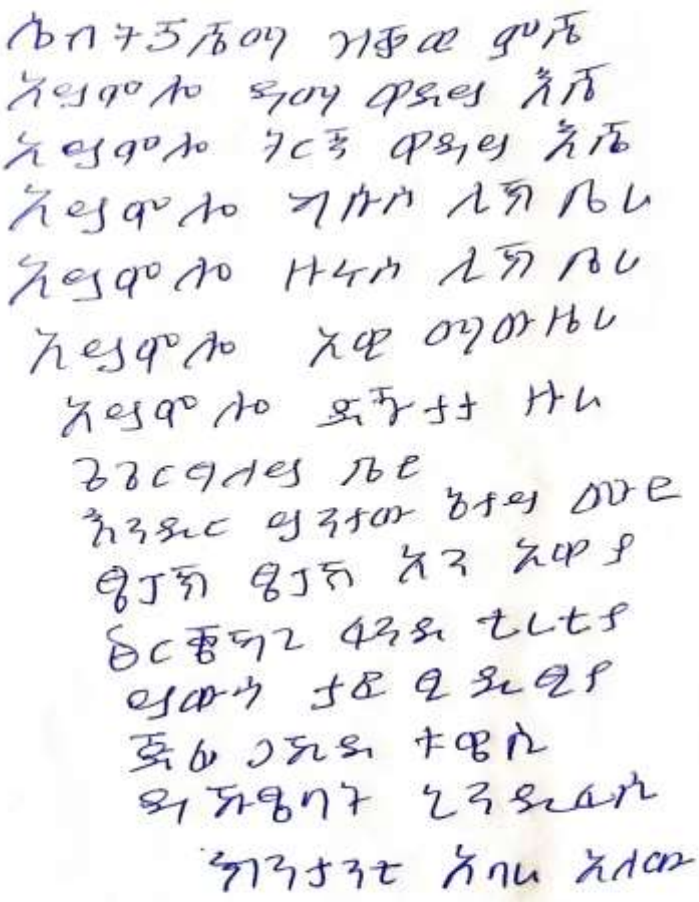
Retsinas, G., Gatos, B., Stamatopoulos, N., & Louloudis, G. (2015). *Isolated character recognition using projections of oriented gradients.* Paper presented at the 2015 13th International Conference on Document Analysis and Recognition (ICDAR).

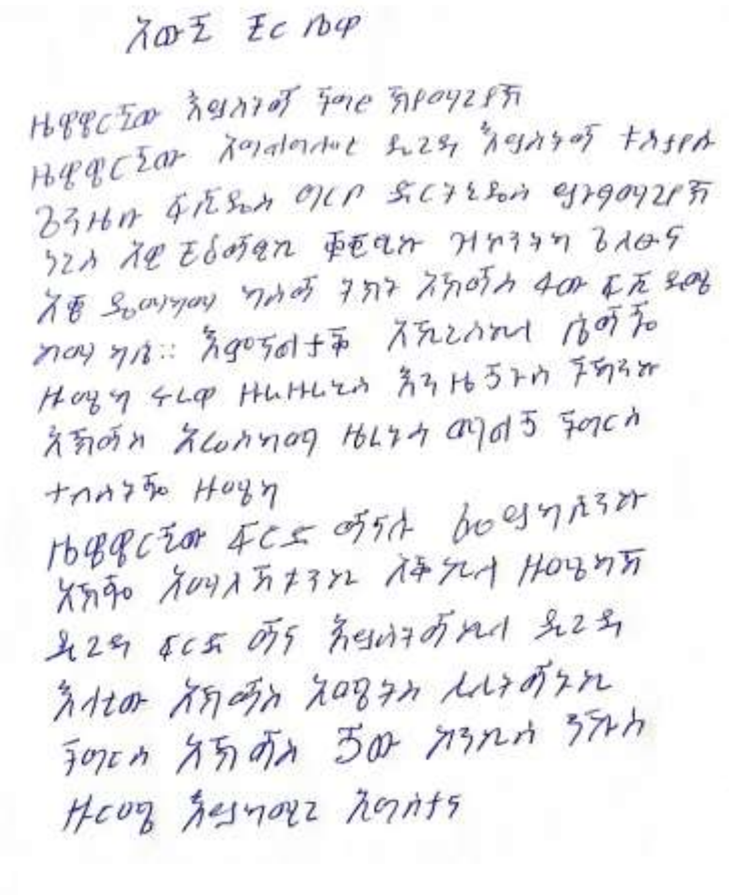
Suzuki, S., Abe, K. J. I. t. o. p. a., & intelligence, m. (1985). New fusion operations for digitized binary images and their applications. (6), 638-651.

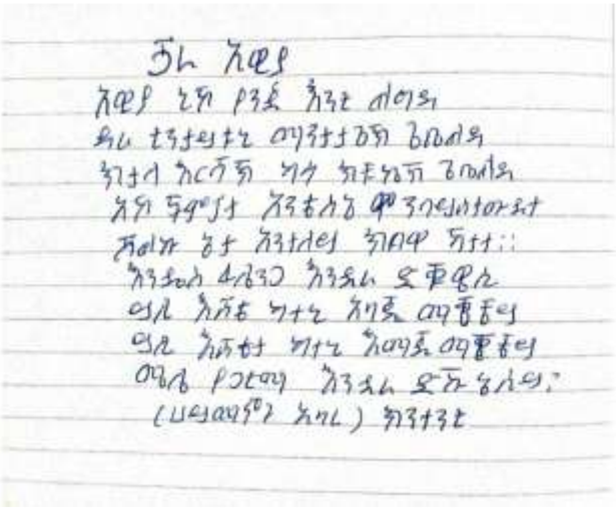
Wu, M., & Goodman, N. (2018). *Multimodal generative models for scalable weakly-supervised learning.* Paper presented at the Advances in Neural Information Processing Systems.

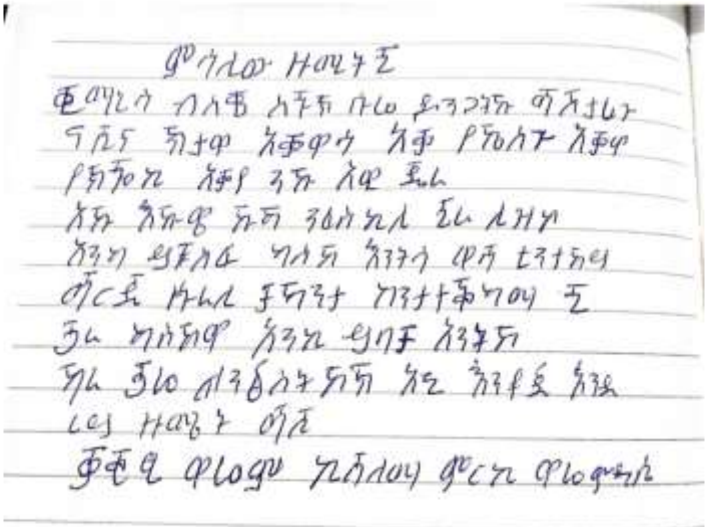
Appendix

Appendix A: sample Awngi handwritten document

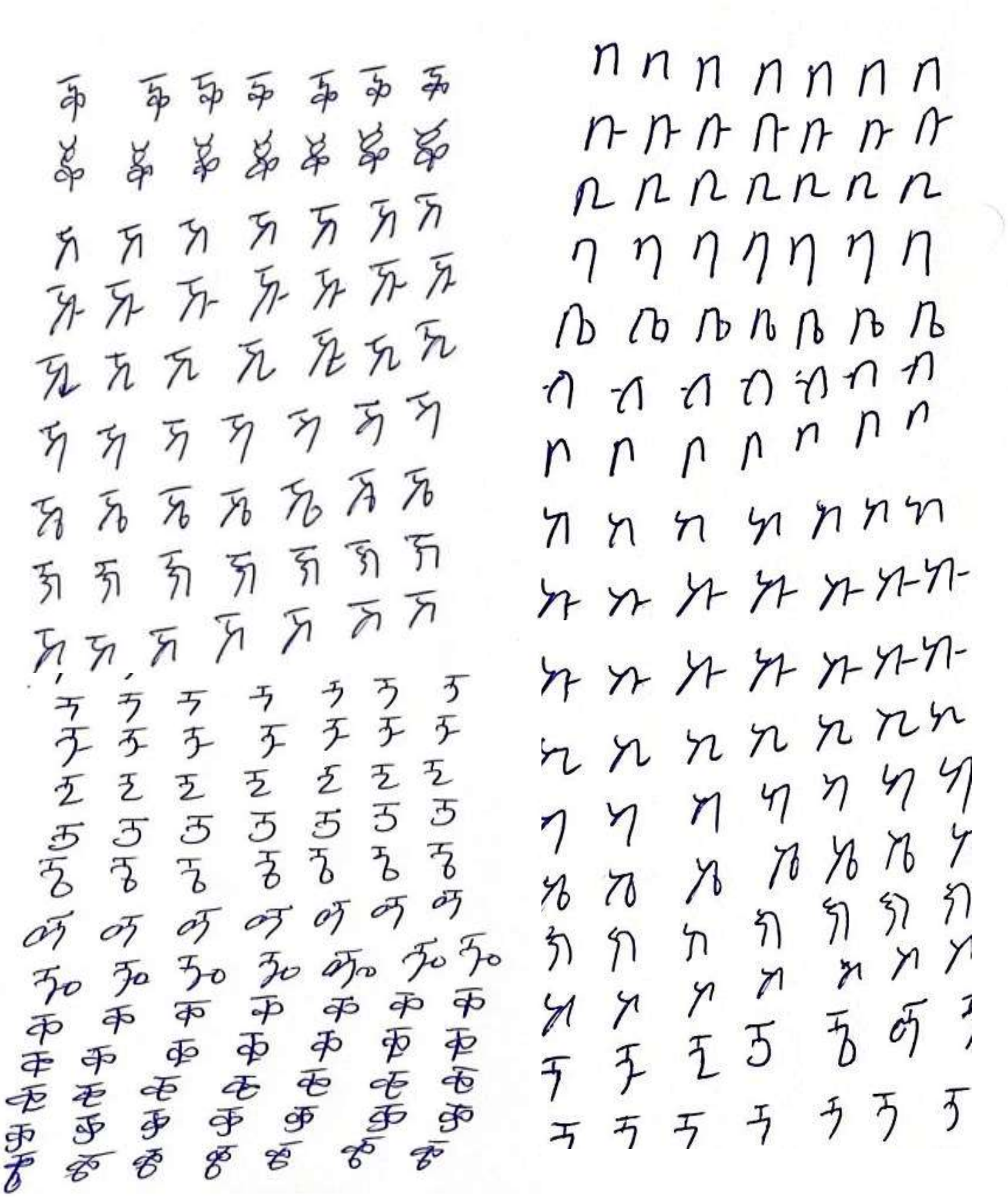




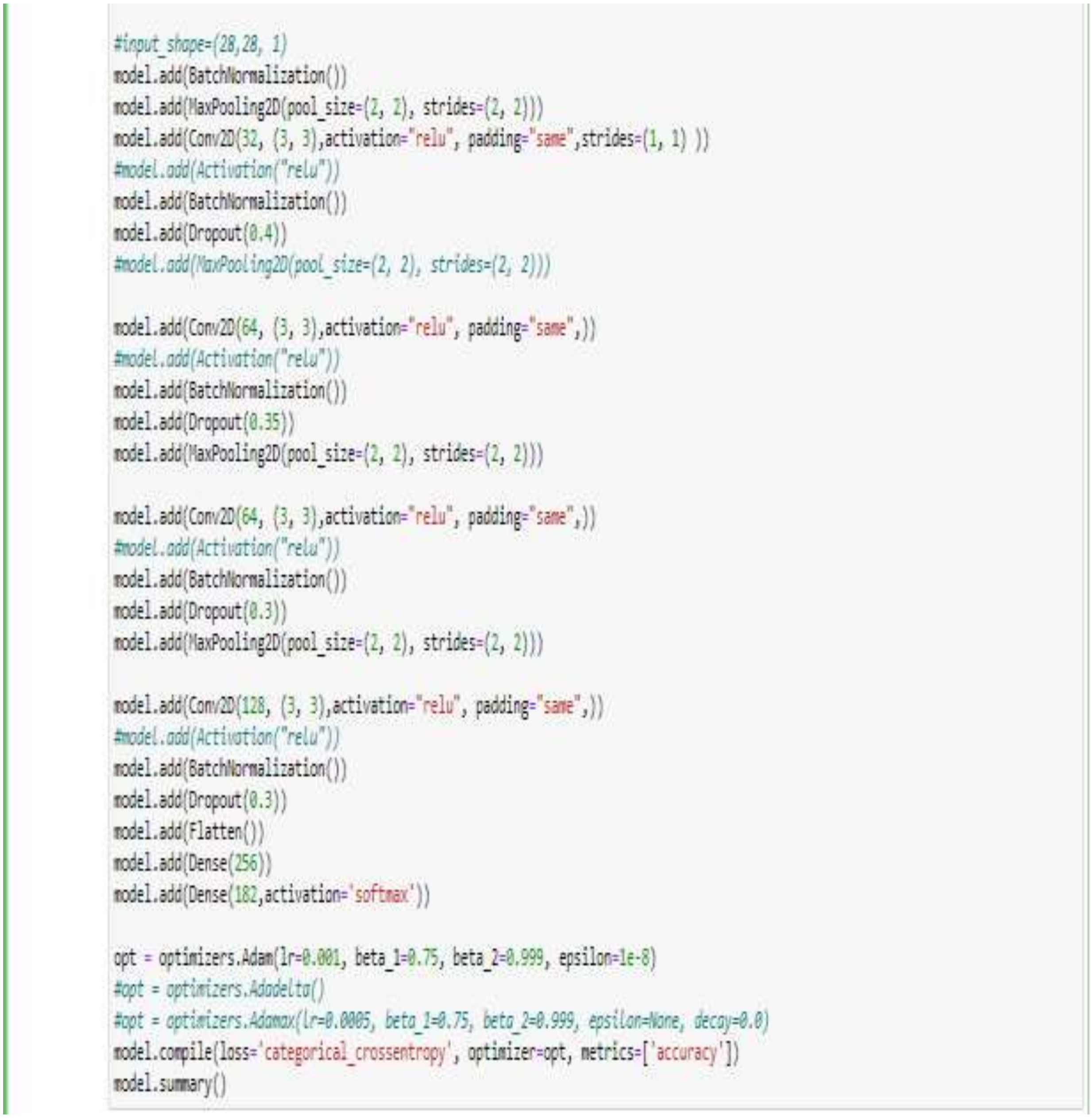




Appendix B: sample Handwritten isolated character

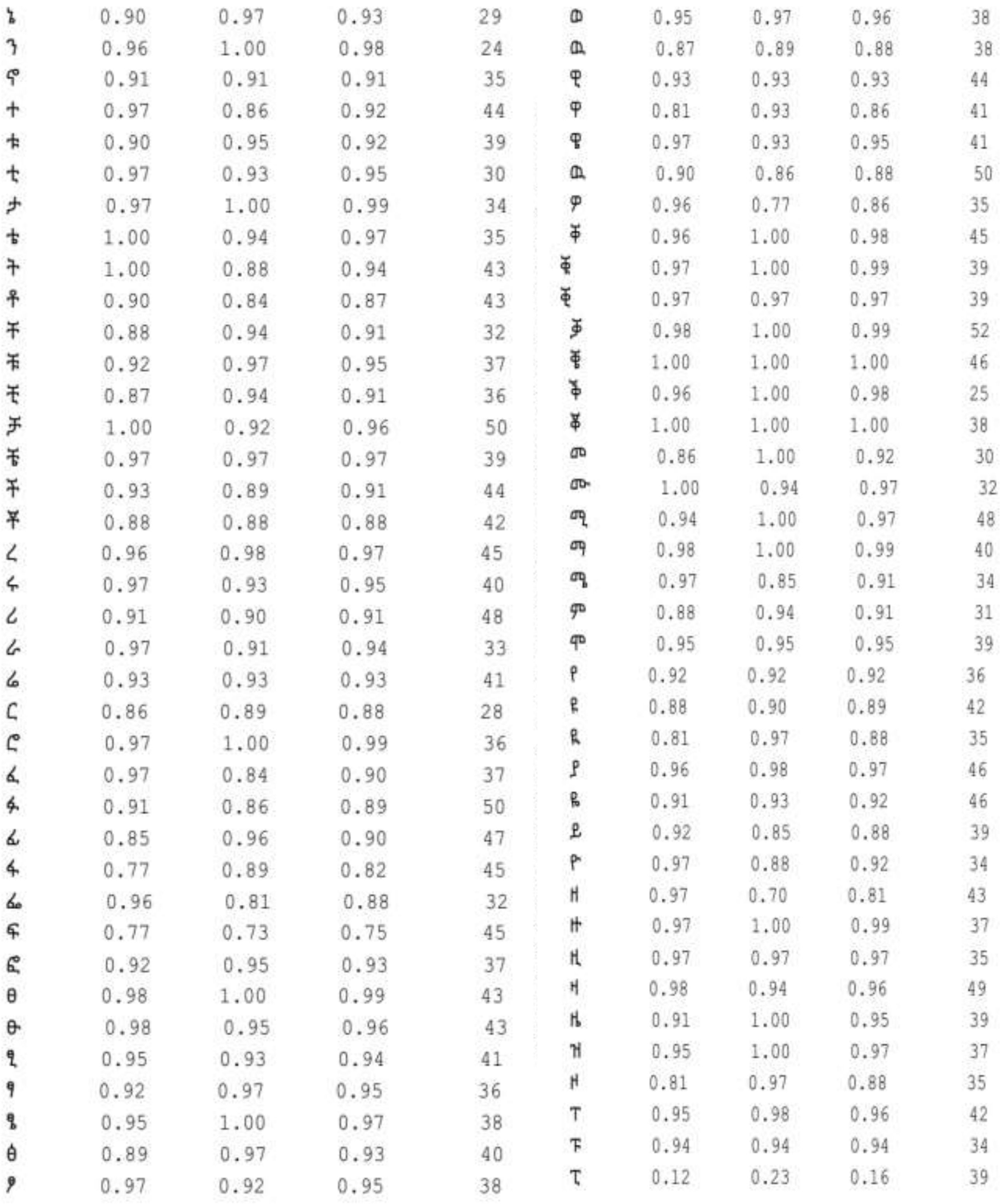


Appendix C: CNN-model design sample code



```
#input_shape=(28,28, 1)
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(32, (3, 3),activation="relu", padding="same",strides=(1, 1) ))
#model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.4))
#model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(64, (3, 3),activation="relu", padding="same",))
#model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.35))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(64, (3, 3),activation="relu", padding="same",))
#model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.3))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(128, (3, 3),activation="relu", padding="same",))
#model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(256))
model.add(Dense(182,activation='softmax'))

opt = optimizers.Adam(lr=0.001, beta_1=0.75, beta_2=0.999, epsilon=1e-8)
#opt = optimizers.Adadelta()
#opt = optimizers.Adamax(lr=0.0005, beta_1=0.75, beta_2=0.999, epsilon=None, decay=0.0)
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
model.summary()
```

Appendix D Model performance evaluation matrix



ኔ	0.90	0.97	0.93	29
ን	0.96	1.00	0.98	24
ኖ	0.91	0.91	0.91	35
ተ	0.97	0.86	0.92	44
ቱ	0.90	0.95	0.92	39
ቲ	0.97	0.93	0.95	30
ታ	0.97	1.00	0.99	34
ቴ	1.00	0.94	0.97	35
ት	1.00	0.88	0.94	43
ቶ	0.90	0.84	0.87	43
ቸ	0.88	0.94	0.91	32
ቹ	0.92	0.97	0.95	37
ቺ	0.87	0.94	0.91	36
ቻ	1.00	0.92	0.96	50
ቼ	0.97	0.97	0.97	39
ች	0.93	0.89	0.91	44
ቾ	0.88	0.88	0.88	42
ረ	0.96	0.98	0.97	45
ሩ	0.97	0.93	0.95	40
ሪ	0.91	0.90	0.91	48
ራ	0.97	0.91	0.94	33
ሬ	0.93	0.93	0.93	41
ር	0.86	0.89	0.88	28
ሮ	0.97	1.00	0.99	36
ፈ	0.97	0.84	0.90	37
ፉ	0.91	0.86	0.89	50
ፊ	0.85	0.96	0.90	47
ፋ	0.77	0.89	0.82	45
ፌ	0.96	0.81	0.88	32
ፍ	0.77	0.73	0.75	45
ፎ	0.92	0.95	0.93	37
ፀ	0.98	1.00	0.99	43
ፁ	0.98	0.95	0.96	43
ፂ	0.95	0.93	0.94	41
ፃ	0.92	0.97	0.95	36
ፄ	0.95	1.00	0.97	38
ፅ	0.89	0.97	0.93	40
ፆ	0.97	0.92	0.95	38
ጠ	0.95	0.97	0.96	38
ጡ	0.87	0.89	0.88	38
ቂ	0.93	0.93	0.93	44
ቀ	0.81	0.93	0.86	41
ቄ	0.97	0.93	0.95	41
ጢ	0.90	0.86	0.88	50
ቃ	0.96	0.77	0.86	35
ቐ	0.96	1.00	0.98	45
ቑ	0.97	1.00	0.99	39
ቒ	0.97	0.97	0.97	39
ቓ	0.98	1.00	0.99	52
ቔ	1.00	1.00	1.00	46
ቕ	0.96	1.00	0.98	25
ቖ	1.00	1.00	1.00	38
ጨ	0.86	1.00	0.92	30
ጩ	1.00	0.94	0.97	32
ጪ	0.94	1.00	0.97	48
ጫ	0.98	1.00	0.99	40
ጬ	0.97	0.85	0.91	34
ጭ	0.88	0.94	0.91	31
ጮ	0.95	0.95	0.95	39
የ	0.92	0.92	0.92	36
ዩ	0.88	0.90	0.89	42
ዪ	0.81	0.97	0.88	35
ያ	0.96	0.98	0.97	46
ዬ	0.91	0.93	0.92	46
ይ	0.92	0.85	0.88	39
ዮ	0.97	0.88	0.92	34
ዘ	0.97	0.70	0.81	43
ዙ	0.97	1.00	0.99	37
ዚ	0.97	0.97	0.97	35
ዛ	0.98	0.94	0.96	49
ዜ	0.91	1.00	0.95	39
ዝ	0.95	1.00	0.97	37
ዞ	0.81	0.97	0.88	35
ፐ	0.95	0.98	0.96	42
ፑ	0.94	0.94	0.94	34
ፒ	0.12	0.23	0.16	39

```
            ፓ       0.09      0.12      0.10        34
            ፔ       0.10      0.07      0.08        44
            ፐ       0.00      0.00      0.00        42
            ፕ       0.95      1.00      0.98        41
            ኘ       0.98      0.98      0.98        45
            ኙ       0.98      0.96      0.97        46
            ኚ       1.00      0.90      0.95        39
            ኛ       0.92      0.97      0.95        36
            ኜ       0.91      1.00      0.96        43
            ኝ       0.93      1.00      0.97        42
            ኞ       0.97      0.85      0.90        39
            ሀ       0.98      0.98      0.98        48
            ሁ       1.00      0.97      0.99        40
            ሂ       0.94      0.96      0.95        46
            ሃ       0.87      0.96      0.91        27
            ሄ       1.00      0.85      0.92        47
            ህ       0.98      0.98      0.98        43
            ሆ       0.98      1.00      0.99        40
            ደ       0.95      0.93      0.94        40
            ዱ       1.00      0.96      0.98        49
            ዲ       0.98      0.94      0.96        47
            ዳ       0.95      0.95      0.95        38
            ዴ       0.86      1.00      0.93        31
            ዶ       0.95      0.91      0.93        46
            ድ       0.90      0.92      0.91        39
            ጀ       0.76      0.79      0.77        43
            ጁ       0.93      0.85      0.89        33
            ጂ       0.83      0.93      0.88        27
            ጃ       0.88      0.84      0.86        55
            ጄ       0.97      0.90      0.94        40
            ጆ       0.97      0.94      0.95        33
            ጅ       0.86      0.90      0.88        42

     accuracy                           0.92      7274
    macro avg       0.92      0.92      0.92      7274
 weighted avg       0.92      0.92      0.92      7274
```