DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION SYSTEMS

Ancient Ge’ez and Amharic Manuscript Recognition Using Deep Learning

By
Desiyalew Haregu Bizuayehu
Advisor: Michael Melese (PhD)

DEBRE BERHAN UNIVERSITY, ETHIOPIA
July 2022

DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION SYSTEMS
Ancient Ge’ez and Amharic Manuscript Recognition using Deep Learning

By:
Desiyalew Haregu Bizuayehu

A Thesis Submitted to Department of Information Systems of Debre Berhan University in Partial Fulfillment for the Masters of Science in Information Systems (MSc in IS)

Advisor: Michael Melese (PhD)

DEBRE BERHAN UNIVERSITY, ETHIOPIA

July 2022

DEBRE BERHAN UNIVERSITY

COLLEGE OF COMPUTING

DEPARTMENT OF INFORMATION SYSTEMS

Desiyalew Haregu Bizuayehu

Advisor: Michael Melese (PhD)

This is to certify that the thesis prepared by Desiyalew Haregu, titled: Ancient Ge’ez and Amharic Manuscript Recognition by Deep Learning and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Systems complies with the regulations of the University and meets the accepted standards concerning originality and quality.

Approved by Board of Examiners

	Name	Signature	Date
Advisor:	Michael Melese (PhD)	______________	_____________
Internal Examiner:	Kindie Nahato (PhD)	______________	_____________
External Examiner:	Million Meshesha (PhD)	______________	_____________

Declaration

I, the undersigned, declare that this thesis is my original work, that it has not been submitted for a degree at any other university, and that all sources of materials used for the thesis have been properly acknowledged.

Name	Signature	Date of submission
Desiyalew Haregu Bizuayehu	________________	___________________

Place: Debre Berhan University, Debre Berhan, Ethiopia

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

Advisor Name	Signature	Date
Michael Melese (PhD)	__________________	____________________

ACKNOWLEDGEMENT

Words cannot express my gratitude to the ultimate **GOD** and his mother, **St. Virgin Mary**, for providing me with strength during the challenges of this thesis and in my life. I also could not have undertaken this journey without my advisor, Michael Melese (PhD), and each individual who gave me direction to pass difficulties through the study. Additionally, I would like to express my gratitude to Mr. Dawit Hassen and Mr. Badimaw Terefe, who gave me important ideas regarding the subject matter of the study.

I am also grateful to my classmates, especially my office mates, for sharing their ideas and moral support. Lots of thanks should be required for my instructors for showing and explaining a new learning path.

Lastly, I would be remiss in not mentioning my family, especially my parents and spouse. Their belief in me has kept my spirits and motivation to complete this study to fulfill the program.

Desiyalew Haregu Bizuayehu

Signiture ________________

Date: ___________________

ABSTRACT

Nowadays, character recognition is one of the hot from the varity research areas in computer vision with its application. It is the process of extracting, detecting, and recognizing characters and converting them to a machine-readable format from document images. Document images may be handwritten or machine printed. Focusing on ancient Ethiopian Ethiopic manuscripts. Among the two forms, handwritten formats in which are written the ancient periods of Ethiopia. Those documents contained the most relevant cultural, and religious knowledge of ancient Ethiopians, but knowledge is limited in place and time to overcome this problem, and if those documents were destroyed by a human or natural disaster, we might lose the knowledge they contained. To address those problems, different scholars have conducted various studies; image digitazation and character recognition are two of them. But still, they have problems with the coresiveness of writing, inconsistancy of writing, nonuniformity of spaces between lines, words, and characters, and morphological similiarity of characters.

In the study, different image processing stages were implemented using Python 3.10.4 through design science research methdology. The researcher primarly collects manuscript document images and binarizes them using OTSU's global thresholding algorithm and bi_level noise filter algorithm, which are implemented for noise filter algorithms and image segmentation, including both line, word and character level image segmentation. After image segmentation is conducted, researcher selects a total of 39,084 character images for dataset preparation from 705 image documents and from 11 different manuscript documents. This is followed by two different experiments using convolutional neural networks(CNN) and a hybrid of convolutional neural networks and bidirectional LSTM (BiLSTM) algorithms with two conditions, one with a dataset split ratio of 70:30% and the other with 80:20% with different parameters and hyperparameters.

Finally, the hybrid of CNN and BiLSTM algorithms outperforms with the second condition of an 80:20 training and testing set split at an epoch of 15 and with a learning rate of 0.0001, and its result is 97.46% tranning accuracy, 90.86% of validation accuracy, and 30.1% of testing accuracy. The performance of manuscript recognition is highly influanced by morphology of characters and oversegmentation.

Keywords: *Convolutional Neural network, Bidirectional Long-Short Term Memory, Manuscript Recognition, Ge'ez Manuscript, Amharic Manuscript, Computer vision*

Table of Contents

List of figures

List of tables

List of Abbreviation

AI	Artificial Intelligence
AMF	Adaptive Median Filter
ANN	Artificial Neural Network
ASCII	American Standard Code for International Interchange
RBM	Restricted Boltzmann Machine
BiLSTM	Bidirectional Long-Short Term Memory
BRNN	Bidirectional Recurrent Neural Network
CTC	Connectionist temporal classification
CNN	Convolutional Neural Network
CNN_BiLSTM	Convolutional Neural Network with Bidirectional Long-Short Term Memory
CPR	Car Plate Recognitions
DL	Deep Learning
DSR	Design Science Research
EOTC	Ethiopian Orthodox Tewahedo Church
RNN	Recurrent Neural Network
ML	Machine Learning
ROI	Region of Interest
ReLU	Rectified Linear Unit
OCR	Optical Character Recognition
PDA	Personal Digital Assistance
PSNR	Peak signal-to-noise ratio
MSE	Mean Square Error
LSTM	Long Short-Term Memory
NN	Neural Network
SVM	Support vector machines
IS	Information System
Thr	Threshold
MI	Manuscript Image

CHAPTER ONE
INTRODUCTION

1.1. Background of the Study

The word manuscript comes from the medieval Latin words 'manu', which means 'by hand,' and 'scribere,' which means 'to write' [1][2]. As a result, any book or document written entirely by hand is referred to as a manuscript. A manuscript is a handwritten document or the text of a musical or literary piece in handwritten or typed script form that has not been copied in multiple copies in that form. Manuscripts are primary sources of human history that can be found on a variety of media, including stones, clay tablets, palm leaves, metal leaves, barks, animal skin, fabric, paper, and other materials [1].

Christian, Islamic and beta Esrayel cultures have developed and been transmitted in Ethiopian and Eritrea mainly through the production and preservation in manuscripts form of numerous texts in different language including Ge'ez, Arabic, Amharic, Harari, Tigrigna and others [2]. Mainly, these language resources are available in various fields (such as theological, liturgical, literary, and history documents.

Ethiopia is the only country in Africa to have its own indigenous alphabets and writing systems called Ge'ez or Amharic alphabets [3][4][5]. Most of the African countries use English and Arabic scripts or alphabets for their languages. The word Ge'ez can also spell as Ge'ez, it's the liturgical language of the Ethiopian Orthodox Tewahedo Church (EOTCs). Ge'ez is a type of Semitic language and mostly used in Ethiopian Orthodox Tewahedo churches and Eritrean Orthodox Tewahedo churches. Ge'ez belongs in the South Arabic dialects and Amharic which is one of the most spoken languages of Ethiopia [6]. There are more than 83 registered languages and up to 200 dialects spoken in Ethiopian, some of those languages use Ge'ez as there writing script. Among the languages, Ge'ez and Amharic, Tigrinya are the most spoken languages and they are written and read from left-to-right, unlike the other Semitic languages which are written and read from right-to-left [6].

There are several ancient manuscripts written in Ge'ez and Amharic that are now in Ethiopian, particularly in the EOTCs [6]. However, due to a lack of a system that can convert them, we are unable to locate a digital format of those manuscripts. The manuscripts, without a doubt, contain a wealth of old knowledge, civilizations, and political and religious attitudes of those peoples [6]. Similarly, documents and texts written in Ge'ez language can be found in Eritrean Orthodox Churches. For long years, Ethiopians and Eretria have shared the same religious beliefs. For many years, Ge'ez has been the primary language in Ethiopian and Eretria's Orthodox churches. The ancient Ethiopian manuscripts are essentially unique and distinct from modern texts [6].

There are a lot of ancient manuscripts which are written in Ge’ez currently in Ethiopia especially in the EOTCs, Ethiopian Catholic Church, Eritrean Orthodox Churches, and the Beta Israel Jewish community [2]. But ancient Ethiopian manuscript is not limited on Ge’ez but also manuscripts are written in Amharic. However, we can’t find a digital format for those manuscripts due to a lack of a system that can convert them. There is no doubt that the manuscripts contain an intense amount of ancient knowledge, civilizations, and the political and religious attitude of those peoples [7][8]. The ancient Ethiopian manuscripts are basically unique and different from the modern handwritten documents. The main differences between modern and ancient handwritten documents are styles of writing, characters size, background colors of the paper or parchment, the writing materials nature, and morphological structure of the characters.

Even though ancient manuscripts have lots of knowledge and wisdom documented with them, they are faced with different challenges that leads to deterioration by physical factors (light, moisture and humidity), biological factors (fungus, insect and worms etc), chemical factors (dust, acidy and atmospheric pollutions) and human factors as discussed by [1].

Sharing and extracting of ancient Ethiopian wisdom has been limited, but this has also advantaged to protect them from being stolen by external parties. To protect manuscripts from the aforementioned and other similar threats, computer technology allows humans to process, store, retrieve, and disseminate information, knowledge, and wisdom with greater flexibility and ease. One of the computer technologies to conduct this task via Artificial Intelligence (AI).

It is "the science and engineering of producing intelligent machines, especially intelligent computer programs," according to John McCarthy [9], the father of Artificial Intelligence. Artificial intelligence is a technique for programming a computer, a computer-controlled robot, or software to think intelligently in the same way as intelligent humans do. AI is achieved by first understanding how the human brain works, as well as how humans learn, decide, and work when attempting to solve a problem, and then applying the findings to the development of intelligent software and systems [9].

AI is a research method that focuses on building systems, or "agents," that work intelligently in a given area and decide to take specific steps based on pre-planned prices and "desirable" areas [10].

Optical character recognition (OCR) techniques are being transformed by artificial intelligence. OCR is a type of computer vision that scans text images and translates them into machine-readable formats. To put it another way, it translates handwritten or typed text from physical documents into digital representations

[11]. It is the way of converting handwriting and printable document into machine-understandable as well as machine editable format [12][13].

Character recognition can be online or offline [14]. Online optical character recognition: based on the real-time operation of writing that means it recognized characters, punctuations, signatures, and the like during the time of writing based on two-dimensional coordinates of successive points are represented as a function of time and the order of strokes made by the writer. While offline optical character recognition this type of character recognition is performed by capturing optically using the scanner or camera then convert into a digital or machine-understandable and editable format [12], [13], [15][14].

The main procedure of optical character recognition is the following stages those are image acquisition or scanning document, segmentation that used to distinguish the text and image, feature extraction, recognition, error correction, and finally output formatting those stages. But there are lots of process in each stage to come up to the desired objective [16].

OCR converts the image of a handwritten or printed document into machine-encoded text or a machine-understandable format that can be used for machine translation, information retrieval, and text mining [17]. In addition to this, OCR makes the file searchable and editable beside saving time, money and space requirement for data storage [17]. The ultimate goal of OCR research is to create systems that can recognize printed or handwritten letters at electronic speed simply by capturing a document. A camera is used to capture documents. The collected documents are then fed into an OCR system, which recognizes the characters and converts them to picture format. Thus, there is a need an automatic way Ancient Ge'ez and Amharic Manuscript Recognition using the current state of the art of artificial intelligence.

1.2. Motivation of the study

In recent years, there has been a lot of interest in Ge'ez and Amharic manuscript recognition due to amount of knowledge they have different domain. The presence of Ethiopic manuscript recognition contributes for the development in different area. In addition, nowadays, applications of computers in document and data processing tasks are crucial parts of automation. There is strong demand for methods to input Amharic and Ge'ez characters. Many of those methods are applied to fulfill those tasks, such as keyboard direct input, encoding direct input, phonetic encoding input, etc. This is incontinence and inefficient, even if for a well-trained typist, and would be a time-consuming task. Another benefit of converting manuscripts into computerized text documents is that we do not need to worry about the maintenance of the manuscript. In addition, new information can be added and modified, something that cannot be done on a simple scanned document.

1.3. Statement of the Problem

Handwriting recognition has become one of the most promising and demanding research areas in artificial intelligence, pattern recognition, and image processing [6]. Handwritten character recognition has proven to be a demanding and difficult area of pattern recognition and image processing thus far. It is critical for digital libraries because it allows image textual information to be entered into computers via digitization, image restoration, and recognition methods [18]. It used for machine translation, information retrieval, and text mining [17]. In addition to this, it makes the file searchable and editable beside saving time, money and space requirement for data storage [17].

As we know, Ethiopia has a long history, majority of the Ethiopian history are written in either in Ge'ez, or Amharic language [21]. The ancient Ethiopian history, culture, customs, expert knowledge, and various religious types of literature has been preserved in these ancient languages. Therefore, by using these handwritten recognitions, a huge set of knowledge can be documented in a digital format for a number of uses in the respective area.

The research attempt made so far shows the segmentation approach and statistical texture analysis for historical written documents. They proposed a horizontal and vertical segmentation approach that is used in different problem solving related to overlapping character documents from historical ancient manuscripts for a single-layer writing style [12]. The study was used for various local languages, including Amharic, majority focusing on Amharic handwriting and machine printed documents, and some focusing on Ge’ez number recognition on handwriting and machine printed documents [3], [6], [12], [13], [18], [20]. And there are many studies and OCR-based applications available that can extract text from an image, but they lack the accuracy to provide the same function for handwritten documents Thus, combining the Amharic and Ge’ez Manuscript recognition could benefit the other Semantic language.

The existing Amharic recognition technology, on the other hand, is incapable of processing Ge’ez characters and Ge’ez numbers. Due to the difficulty in recognizing handwritten factors such as, non-uniformity in writer’s, writing style, cursive-ness of handwriting, morphological similarity and the numbers of characters variation in the languages [3], [4] , [6],[13], [18], [20], [21]. Amharic characters originated from Ge’ez characters, including additional characters and labialized characters.

A large number of manuscripts in Amharic and Ge'ez have been discovered, but they are of poor quality and may be lost. As a result, developing an OCR system for Amharic and Ge'ez manuscripts is necessary in order to digitally preserve these manuscripts. As a result, the current study's primary goal is to create an OCR model for Amharic and Ge'ez manuscript scripts. At the end this study attempts to solve the above stated problem, and the researchers investigated possible answers to the following research questions.

1. Which deep learning technique best work for the recognition of Ancient Ge’ez and Amharic manuscript?
2. To what extent does the developed model correctly recognize manuscripts in Amharic and Ge’ez scripts?

1.4. Objective of the study

In the study, general and specific objectives are set to achieve the goal of the research.

1.4.1. General objective

The general objective of the study is to design and develop Ge’ez and Amharic character recognizer model that enable Ancient Ge’ez and Amharic manuscript character recognition by using Deep learning technique.

1.4.2. Specific objective

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

- ✓ Investigate literature to understand the state-of-the-art in manuscript recognition and deep learning technique.
- ✓ Gather required manuscript image data.
- ✓ Do basic noise removal on the collected manuscript image.
- ✓ Apply image binarization technique.
- ✓ Apply image segmentation.
- ✓ Apply size normalization.
- ✓ Prepare representative datasets for training and testing purpose.
- ✓ To Design deep learning-based recognition models.
- ✓ To Conduct experiments with deep learning models.
- ✓ To Evaluate the performance of the proposed model.
- ✓ To Select the best performer model.
- ✓ To develop recognition prototype for best performer model.
- ✓ To Report the finding of the study and,
- ✓ To Recommend for upcoming research area in the field.

1.5. Scope and Limitation of the Study

This research is conducted based on the image data collected from different sources and the data is collected from Ethiopian orthodox church (from Dima St. George's Ancient Monastery and Yetmen Abune geberemenfeskidus church, East gojjam), from 11 different manuscript document from both Amharic and Ge’ez scripts and in type from paper and Parchment(ብራና) documents. However, using those data directly can’t achieve research objective because by nature optical character recognition needs to pass different stages to achieve its objective. Therefore, to address our goal we are planned to conduct image preprocessing technique to binarize, denoise, segmentation, normalize simple to make our dataset smoothed, clean and normalized. The main goal of this study is to develop recognition model, for recognizing Amharic and Ge’ez manuscripts. To develop the model deep learning-based algorithms are used by setting neural network parameter and hyperparameter.

This research was limited to recognized Ethiopic scripts (abugida), which contains 26 main characters, the Amharic script (Fidel), which borrows all of the Ethiopic scripts except labialized characters and Ge’ez numbers. In the Ethiopic script, all are consonants; there is no vowel with seven different forms. They are derived from the main character by adding a stroke. The Amharic script accepts all of the Ethiopic script and adds its own, so it becomes 34 in number with an additional 27 labialized characters (character representing two sounds). Due to a lack of occurrences in the extracted characters, this work is limited to all main characters ((34*7) ==> 238) and does not include labialized characters. In addition, the study is limited to developing character-level Amharic and Ge’ez manuscript recognition.

1.6. Significant of the study

Manuscript documents, written in a variety of languages, can be found in abundance all over the world, particularly old texts, scriptures, and manuscripts that offer a wealth of knowledge and are written by hand. However, due to the difficulty and impossibility of finding a digital or editable format of the information, a large portion of the documents containing valuable information cannot be easily accessible on the internet. For any language, the process of transforming information into digital forms is heavily dependent on the language's characteristics as well as the morphological structures of the characters [6].

Like other countries, our country, Ethiopia, has lots of relevant manuscript documents that encode a vast amount of information, knowledge, culture, norms, and beliefs etc. However, extracting, disseminating, searching, and inserting information is difficult. Due to its nature and being limited in place and time, in addition to this, those documents require large storage spaces and they are faced with different disasters in case we may lose them. While we are losing them, we may lose a vast amount of information.

There are many potential significances of the proposed OCR system in this thesis work. To mention some:

- To allow for the most efficient storage of manuscript real-life documents in all automated organizations.
- To make searching, inserting, and changing the manuscript document portion easier.
- To digitize and preserve irreplaceable materials in museums, archives, and other institutions.
- To help government and corporate entities automate their offices.
- To facilitate extraction and dissemination of knowledge, wisdom and culture documented of the manuscript from ancient and current generation to the next one.
- It will be useful information for other researchers who are working on manuscript recognition.

1.7. Methodology

In conducting this research, different methodology used. This include, research design, literature review, data source, tools and technique beside the evaluation technique.

1.7.1. Research Design

The Design Science research methodology is used as a general research methodology to achieve the study's general and specific objectives, as well as to answer the research questions. The following subtopics will be used as an additional research methodology to achieve the study's general and specific objectives and to answer the research questions.

1.7.2. Literature Review

Conducting a literature review provides insight into current study areas' state-of-the-art models. It will assist us in understanding current approaches, strategies, and instruments. It provides a detailed interpretation of the problem domain in order to develop an ancient Amharic and Ge’ez Manuscript recognition model that is effective. A comprehensive survey of the literature in the field of OCR will be conducted, with a special emphasis on Deep Learning techniques and algorithms. Many peer-reviewed publications will be reviewed, including books, essays, journals, and other scholarly publications.

1.7.3. Data Collection

To conduct this research, the required data was collected and preprocessed by the researcher. The necessary data for construction was obtained from an Ethiopian orthodox Tewahedo church using a digital camera and 11 different Ge’ez and Amharic manuscript documents. A researcher collected 705 ancient manuscript document images for the Amharic and Ge’ez languages. Out of 705 manuscript images, 515 were collected from Ge’ez manuscripts and the rest 190 were from Amharic manuscript documents. Then, after the acquired images go through the development of the manuscript recognition model.

1.7.4. Tools and Technique

Python has a large number of unique libraries that can be used to implement deep-learning algorithms. It's an image preparation library that's open-source. With just a few routines, it can execute complicated actions on images using machine learning and built-in algorithms. We'll use the keas, OpenCV, TensorFlow, Numpy, Mateplot packages, and Jupiter Notebook to create models with Python 3.10.4. OpenCV is an excellent tool for image processing and computer vision. Because images are made up of arrays, we can use NumPy to accomplish various image processing tasks as well as start from scratch. Matplotlib is a Python data visualization and graphical plotting package that works across platforms. imshow () is a matplotlib function that makes a picture from a two-dimensional numpy array and its numerical extension NumPy. Kera’s will used for creating of deep learning models.

In this study, the above listed python packages were used while we were preparing or preprocessing image data with an appropriate algorithm for each step of image preprocessing activity, and development of proposed model.

1.7.5. Evaluation Technique

At this stage, the results of the study will be evaluated. As we have discussed above, for developing recognition models, two deep learning algorithms with different scenarios will be implemented. Then, the developed model computes against each other with respect to recognition accuracy of training and validation also training complexity of the model. Then the outperformer of the model which generates promising training and validation accuracy will be selected for the development of the proposed model. Therefore, researchers will be implementing accuracy and training complexity as evaluation metrics and also researcher will implement human (manual) evaluation system.

1.8. Organization of the Study

This document is organized into five chapters. The first chapter presented the statement of the problem, objective, scope, methodology, Significant, scope, and limitations of the study. Chapter two, deal with literature review about ancient Amharic and Ge’ez Manuscripts recognition and character recognition technology, which has been done by different researchers. Different researcher’s present different approach to recognize the OCR system in different language with different technique to achieve better accuracy of the system. Chapter three is about addresses the proposed character recognition model involves training of a convolutional neural network and CNN-BiLSTM using binary images of individual characters segmented and discusses manuscript document image preprocessing stages and general architecture of the proposed system with them explanation and futures. Chapter four addresses the Experimentations and results. In this chapter how each experimental process is clearly discussed and how researcher implements the proposed system framework and how researcher address his research questions will be discussed clearly. Finally, chapter five concludes the study and give future research direction.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview

In this chapter, we are going to discuss different issues regarding optical character recognition, Ethiopian manuscript, image processing in machine learning, and deep learning algorithms.

2.2. Ethiopian Languages

Ethiopia is home to 86 different languages, with up to 200 dialects [19]. Ge'ez is an ancient language that was established as an official written language during the first Aksumite kingdom, when the Sabeans sought refuge in Aksum. The Ge'ez is a unique script derived from the Sabean alphabet, which was developed by the Aksumites and is still used by the Ethiopian Orthodox Tewahedo Church today. The modern languages derived from Ge'ez are Tigrigna and Amharic. Ethiopia's official national language is Amharic.

Ethiopian languages are classified into four major groups. Semitic, Cushitic, Omotic, and Nilo-Saharan are among them [19].

Semitic languages are spoken primarily in Tigray, Amhara, Harar, and the northern part of the Southern Peoples' State regions of Ethiopia. They use the country's unique Ge'ez script, which consists of 33 letters, each of which denotes 7 characters, for a total of 231 characters. Cushitic languages are mostly spoken in Ethiopia's central, southern, and eastern regions (mainly in Afar, Oromia and Somali regions). The Roman alphabet is used in the Cushitic languages. The Omotic languages are primarily spoken between the southern Rift Valley lakes and the Omo River. And the Nilo-Saharan languages are mostly spoken in the country's west, near the border with Sudan (mainly in Gambelia and Benishangul regions) [19].

2.3. Manuscript in Ethiopian

The manuscript is the document that is written in hand and the opposite of machine-printed on [4], [20]. It bears diverse contents on various subjects, about traces of civilization and literary traditions in the long history of humanity throughout the world. In various monasteries, churches, mosques, libraries, museums, and also in private collections in every part of the world, these documentary heritages have been gathered.

Manuscripts are written artifacts, that is, material objects provided by the man with visual signs [21]. Visual signs are their contents that express, record, store, and transmit human culture in its broadest meaning. Contents are formatted according to patterns, and patterns are in turn determined by settings and practices. No formatting is therefore neutral, but every single feature of each pattern adopted is determined by specific factors that are culturally and materially determined. There are cases, however, when a transmission or use of the written artifact and specific practices determine more complex formatting types. Ethiopia is the only African country with its indigenous alphabets and writing system which is the Ge'ez or Amharic alphabets. Most other African countries use English and Arabic scripts or other alphabets [22], [6]. Many languages in Ethiopia use a unique alphabet Ethiopic for the writhing system.

Ethiopia, a country of nation, nationalities, and peoples with history and culture of their own, have experienced and advanced in literary traditions with magnificent intellectual involvements in the ancient and medieval periods during which a large number of Manuscripts were hence produced. In terms of quantity, Ethiopia possesses a tremendous number of Manuscripts. This is because of the late introduction of the printing press to the country at the beginning of the 20th C. Despite, many Manuscripts were exposed to damage due to civil war, lack of public awareness, poor preservation method, and bad weather conditions of the country, the surviving manuscripts bearing the antique wisdom of the country [23].

Manuscript in Ethiopia developed in course of Christian, Islamic and Beta Israel cultures and tradition, the language that used to write the Manuscript includes Ethiopic (Ge'ez, old Ethiopic), Arabic, and Amharic and in the lesser degree: Harari, Ajami, and Tigrigna scripts [23].

With reference to Manuscripts (MSS) proclamation No. 209/2000 issued by the Ethiopian House of Representatives in 2000 on the research and preservation of cultural heritage, manuscript (MS) is an integral part of Tangible Heritage [23] [24]. Cultural Heritage articulated in the proclamation by saying that 'Cultural Heritage not attached to the foundation which are often moved from place to put easily and which are handed down from the past generation and shall include: Parchment Manuscripts, etc…" In the shape of a manuscript preserved over generations, Ethiopia has inherited a huge treasure of data as a present from our ancestors. Most of the Ethiopic Manuscripts are concentrated within the northwest part of the country, especially the Amhara and Tigray regions [24]. Historically, Ethiopian orthodox churches (EOC) were the sole archive center of the empire most official documents and included Manuscripts were saved and preserved. Thus, most Ethiopic Manuscripts collections are lying in monastic 'libraries' and churches, though they need not yet been systematically registered. Thousands of Ethiopian manuscripts reside outside of Ethiopia. Many of those are completely unknown, uncatalogued, and unavailable for scholarly purposes, privately hands. Currently, only the three largest collections of Ethiopian manuscripts in Europe alone are; Rome, Biblioteca Apostolica Vaticana; Paris, Bibliothèque Nationale de France; London, British Library together include ca. 2,700 manuscripts. Similarly, Richard Pankhurst stated around 5,000 Ethiopian manuscripts are scattered outside the country. Many of those works are of fundamental importance for Ethiopian studies [23][25].

In Ethiopia, there are only two collection centers for different manuscripts even if many European countries have a larger number of Ethiopian manuscripts in different collection centers. These collection centers are found in Ethiopia Institute of Ethiopian Studies and the National Archives and Library of Ethiopia (1500 and 835 manuscripts respectively). And these collections are still uncategorized [23][25].

While Ethiopia holds much more knowledge and wisdom from the earliest up to the day to day in different area which is written on historical manuscripts for a different purpose just like traditional medicine, governance, literature, social, economic, political, cultural and the like but not transfer in connivance and appropriately for the current generation due to lack of proper preservation and transfer of those documents from the oldest to the current generation.

2.4. Computer Vision

Computer vision means a computer understands and processes images and videos [26]. Seeing is about checking out from images what is present within the scene and where it's located. In Computer Vision, a camera (or cameras) is connected to a computer. The computer interprets images of a true scene to supply information useful for tasks like navigation, manipulation, and recognition. Computer vision is a neighborhood of computer science that permits a computer to spot and process objects in videos and pictures even as we humans do. Although computer vision doesn't appear to be a really old concept, it dates back to the late 1960s when the primary digital scanner was invented that converted images into grids of numbers [26][27].

Computer vision is a neighborhood during which a machine has got to "see". This technology uses a camera and computer rather than the human eye to spot, track, and measure targets for further image processing [28]. Vision enables humans to perceive and understand the planet around them, while computer vision aims to duplicate the consequences of human vision through electronic perception and understanding of a picture [29].

In order to automate the visual perception process, computer vision addresses theories and algorithms and includes tasks such as noise removal, smoothing and edge sharpening (low-level vision); image segmentation to isolate object regions and segmented region description (intermediate level vision); and finally, scene interpretation (high-level vision) [30] [31]. The main purpose of Computer Vision is to generate image models and data extracts and data, while Image Processing is about implementing image computational transformations, such as sharpening, contrast, among others.

In order to help in decision-making, it includes ways to capture, process, and analyze images and video from the real world. Computer vision implies mimicking vision that is biological (that is, human and non-human). The final objective of most computer vision systems is to extract useful information for decision-making from still images and videos (including pre-recorded videos and live feeds). Systems of biological vision work similarly. In addition, computer vision can also acquire and work with images from the visual spectrum that are not visible to biological entities, such as infrared and profound images, unlike biological vision [32].

2.5. Digital Image Processing and Recognition

Now a day world needs much more data consumption from different sectors and in a different form. So, we use a computer for this purpose to prepare those different formats of data for further processing. Computers are being increasingly used to process and analyze the image.

Image is a kind of language that expresses visual information to human beings [33]. The image is an optical representation of an object illuminated by a source of radiation. [34], There are two types of images which are analog and digital. The analog image is continuous and indiscreet, and the natural image of the environment and the digital image is a two-dimensional image with a finite set of digital values called image elements or pixels [33][35]. A digital image consists of discrete pixels or a set of pixels, each of which has an associated integer brightness value called a gray-level. In other words, a digital image is a data matrix or array of elements known as image pixels. Each pixel has an integer location and the integer value is proportional to the brightness at the pixel's spatial point. Pixel brightness accuracy is measured by the number of bits or gray levels [33].

In a 2D discrete space, a digital image a [m, n] described is derived from an analog image a(x, y) in a 2D continuous space through a sampling process often referred to as digitization. The (x, y) continuous 2D image is split into N rows and M columns. A pixel is called the intersection of a row and a column. a[m, n] is the value assigned to the integer co-ordinates [m, n] with {m=0,1,2,...,M-1} and {n=0,1,2,...,N-1} [36]. A digital image is formed as a 2-D function, f(x, y), where it is possible to describe x and y as spatial coordinates, and the amplitude of 'f' at any pair of coordinates (x, y) at that point is called the image's intensity or gray level. All finite, separate quantities are the coordinates x, y, and the amplitude values of 'f', we call the image a digital image that is obtained by digitizing analog image [33].

There are three types of digital images that are [37]:

A. Binary images

Are the simplest types of images and can take two values, usually black and white, or 1 and 0. It is also referred to as the image of a 1-bit map. These types of images are applicable where the only information required is general shape or outline just as optical character recognition (OCR). This is a type of image that is created from a grayscale image through a threshold operation, where each pixel above the threshold value is white ('1') and the one below is black ('0').

B. Gray-scale images

Grayscale images are referred to as monochrome (single color) images they contain gray level information, no color information. For each pixel, the number of bits used determines the number of different available gray levels. In typical grayscale images, there is 8 bit/pixel information, which allows us to have 256 different gray levels. And this applies to 12 or 16 bit/pixel images, just like medical imaging and astronomy.

C. Color images

It is possible to model color images as three-band monochrome image data, where each data band corresponds to another color. In each spectral band, the actual information stored in the digital image data is the data on the grayscale image. Typical images of color are represented in red, green, and blue (RGB image).

Digital Image Processing (DIP) is a process that uses different computer algorithms to process digital images. This digital image processing has been employed in several areas such as pattern recognition, remote sensing, image sharpness, color and video processing, and medical [38]. As it provides advanced visual data for human simplification and image data processing for transmission and explanation for machine perception, image processing is always interesting Digital images are processed to give a better solution using image processing [39]. Image processing can be defined as manipulation of images such as refining of images, digital image processing is the most emerging field with many growing applications in field science and engineering, the total goal of this manipulation can be divided into three categories as [40]:

- Image processing: image in→ image out
- Image analysis: image in → measurements out
- Image understanding: image in→ high-level description

Image processing is a technique for improving raw images obtained from a camera placed on satellites, aircraft, or images taken for different applications in normal life. The use of computer algorithms to carry out image processing on a digital image is digital image processing. Digital image processing focused on the following two major tasks:

- Optimizing pictorial information for human interpretation
- Processing image data for storage, transmission, and representation from machine self-perception.

Digital image processing is classified in to low, middle or high levels in three different levels [1].:

Low-level process: - involves primitive operation such as image preprocessing to reduce noise, contrast enhancement, image sharpening, etc. in the low-level process both inputs and outputs are images. Like low level, the mid-level process: - image processing involves tasks such as image segmentation, description of images and object recognition. Inputs are typically images in the mid-level process, but outputs are usually image attributes. Whereas high-level image processing includes "make sense" from a group of recognized objects, usually associated with computer vision in this process.

The digital image processing (DIP) has been employed in a number of areas, particularly for feature extraction and to obtain patterns of digital images. And it is the critical task under character recognitions. Because character recognition is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. These digital images don't pass directly as input for character recognitions instead the digital image primarily preprocess using digital image processing techniques and it generates extracted features and patterns of digital image these features and patterns are used as input for character recognition.

2.6. Character Recognition

Character recognition is the mechanical or electronic conversion of typed, handwritten or printed text into machine encoded text, whether from a scanned document, a photo of document [41]. We have to process, analyze, classify, and recognize a digital image for different purposes. From those we have focused on character recognition, it is one task of image recognition. But before recognition or detection of character from the digital image many stages pass to achieve the main objective of character recognition. So, before we have to focus on the technical part of character recognition, we have to discuss character recognition and its overview as well as what scholars say about it. "Character recognition is a process which allows computers to recognize written or printed characters such as numbers or letters and to change them into a form that the computer can use" as Collin dictionary defines.

Character recognition is the process by which the input image detects and recognizes characters and converts them into ASCII or another editable form of an equivalent machine. An image of a printed or handwritten document can be the input image. Recognition of handwritten / printed characters involves converting handwritten/printed documents into a machine-editable document in the original order containing the extracted characters. Technically, various kinds of step-by-step stages and sub-processes are included in the handwritten/printed text recognition process. Image acquisition, which includes collecting and preparing images for further processes, is the first procedure. Pre-processing, which includes operations such as removing noise and clearing the scanned image background so that the relevant character pixels are visible. The process of segmentation involves extracting lines, words, and characters from the scanned image after completing the preprocessing stage. Finally, the segmentation step is followed by extracting features and using the selected classification approach to classify the characters.

With a large number of applications, it is fundamental but most difficult in the field of pattern recognition [42]. Character recognition is the process to classify the input character according to the predefined character class [43]. It is the process that allows a system to identify the script or alphabets written in the verbal communication of the user without manual assistance, and it achieves segmentation, feature extraction, and classification [44]. Character recognition is a scheme that gives full alphanumeric recognition of printed or handwritten characters by simply scanning the text image. It interprets and converts the image of a printed or handwritten character into an appropriate editable text document. By isolating each line, the text image is split into regions, then individual characters with space [45].

Character recognition is the process of detecting and recognizing characters from an input image and converting them into the American Standard Code for Information Interchange (ASCII) or other equivalent machines in editable form. The input to this system can be handwritten text or printed text. A handwritten text recognition system aims at transforming a large number of documents, handwritten into machine-encoded text [12].

Character recognition is the recognition of characters from documents it may be printed or handwriting documents but not only the recognition of character from documents instead of also characters detected while we have written. This leads us to character recognition is classified into two categories; offline and online character recognition. Let’s discuss the classification of character recognition which is adapted from [42] as follow:

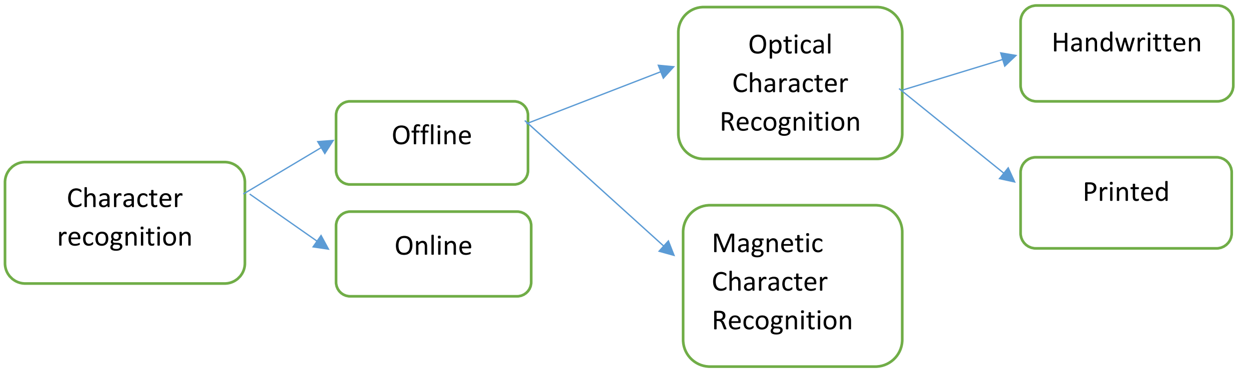


Figure 2. 1. Classification of Character Recognition

An increasing number of people in emerging markets are obtaining access to computer devices, many exclusively using mobile devices with touchscreens. So at the time of the user writing, data is prepared from a pressure sensing of the transducer because the successive movements of the pen are modified into an electronic signal with memory and can be easily analyzed by the computer [46]. The source of online handwriting data is systems that simulate a pen-paper as an arrangement for a writer. These systems record the information of the pen-tip as a sequence of coordinate data points sampled over a period of time (x, y) [47]. For the calculation of the similarity measurement, this method uses all points per stroke. It involves automatic text conversion as it is written on a special transducer or PDA, where the movements of the pen tip, as well as the up/down pen shift, are detected by a sensor [43]. While in offline character recognition the data is captured optically by a scanner in the form of an image [47]. These involve the automatic conversion of the text in an image into letter codes which are usable within computer and text-processing applications [43][48].

Optical Character Recognition vs Magnetic Character Recognition

Optical character recognition is the fast and automatic recognition and conversion of images of printed or typed text into real electronic data that users can organize, search, and edit. Typically, an OCR engine analyzes the pixel data of scanned images and searches for patterns resembling letters, numbers, and other symbols to create a digitized record. Magnetic character recognition is a system that uses ink and special characters to recognize characters. It passes through a system that magnetizes the ink when a paper containing this ink is to be read and then converts the magnetic information into characters [4]. Magnetic character recognition operates only on a predefined font family, while OCR can work with various font shapes and varieties.

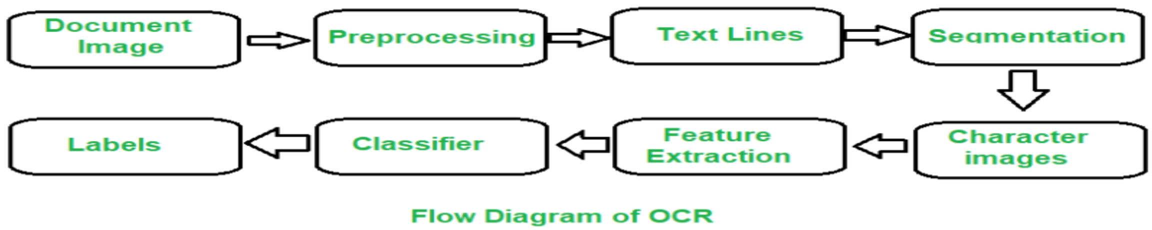


Figure 2. 2. workflow diagram for optical character recognition [49]

OCR extracts text from scanned document images. The shape of each possible text character in the image data is compared with a sample (template) for each character stored in the computer by OCR software. When a character is recognized, it is added to the output data sequence [49].

Figure 2. 3. workflow diagram for magnetic character recognition [49]

MICR is capable of reading text printed with magnetized ink. The banking industry is the sole user of magnetic ink character recognition for check processing. Customers' check numbers, bank numbers, and account numbers are represented by the characters. An MICR can recognize characters printed with a special ink that contains magnetic material particles. This device is particularly useful in the banking industry. Because the MICR system can only recognize certain character styles, the characters must be correctly formed [49].

2.7. Artificial Intelligence

AI is a broad discipline aimed at understanding and designing systems that display the characteristics of intelligence emblematic of which is the ability to learn: to draw knowledge from data. This is a broad definition that is arguably somewhat interlinked with current statistical techniques [35]. With its roots in philosophy, mathematics, and computer science, this broad scientific discipline aims to understand and develop systems that display the properties of intelligence.

Artificial intelligence, both within and outside the scientific community, has become a popular subject. In 1956, a group of computer scientists suggested that computers could be programmed to think and reason, "In principle, every aspect of learning or any other feature of intelligence should be described so precisely that a machine should be made to simulate it," and they describe this concept as artificial intelligence [36].

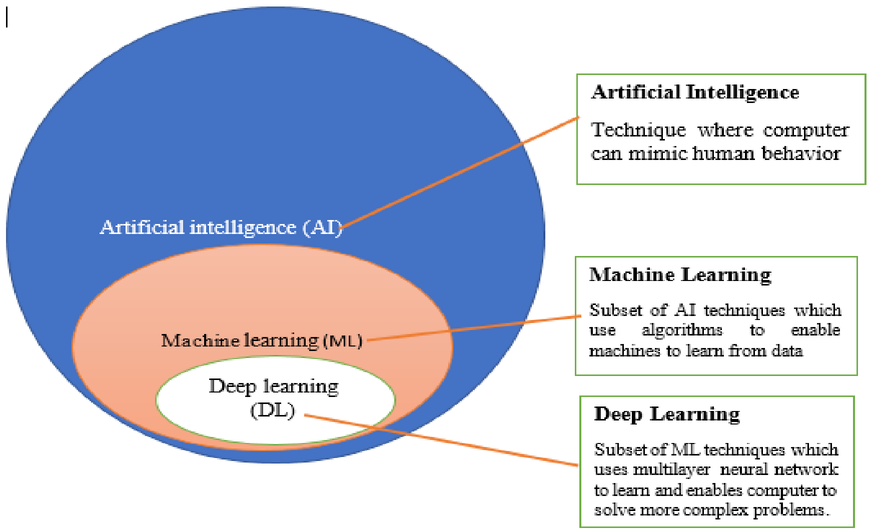


Figure 2. 4. relationships between AI, ML and DL adapted from [50] [51]

Simply AI is a field focused on automating intellectual tasks normally performed by human beings. And to achieve this task machine and deep learning methods are used. That is, they are within the field of AI. Deep learning is part of the learning of machines. Machine learning is also part of AI, and AI is an umbrella for any smart computer program. So, we can redefine this statement as, all machine learning is AI, but not all AI is machine learning, and so forth**.**

2.8. Machine learning

A sub-discipline of AI, where computer programs (algorithms) learn associations of predictive power from examples in data [52]. The most basic application of statistical models to computer-based data is machine learning. Machine learning employs a broader range of statistical techniques than are typically employed in different sectors. Deep Learning, for example, is based on models that make fewer assumptions about the underlying data and can thus handle more complex data. It is the ability of a machine to improve its performance based on previous results, as well as machine learning methods that allow computers to learn without being explicitly performed and have numerous applications, such as data mining improvement.

Machine learning is the science in which computers learn and act like people and, over time, improve their learning autonomously, by feeding data and information through observation and real interaction with the world [53].

Machine learning uses a range of algorithms to boost, explain data, and forecast results iteratively learned from data. As training data is consumed by the algorithms, it is then possible to generate more reliable models based on that data. The performance generated when you train your machine learning algorithm with data is a machine learning model. After preparation, you will be presented with an output when you supply a sample with an input. A predictive algorithm, for example, can construct a predictive model. Then you will obtain a forecast based on the data that trained the algorithm when you supply the predictive model with data. Machine learning is now important for the creation of computational models [54].

Machine learning incorporates several hundred algorithms based on statistics and it is a relentless task for those working in this area to pick the best algorithm or combination of algorithms for the job. But it's important to understand the three overarching categories of machine learning before we analyze individual algorithms. These three types are regulated, unmonitored, and enhanced [55][56][57].

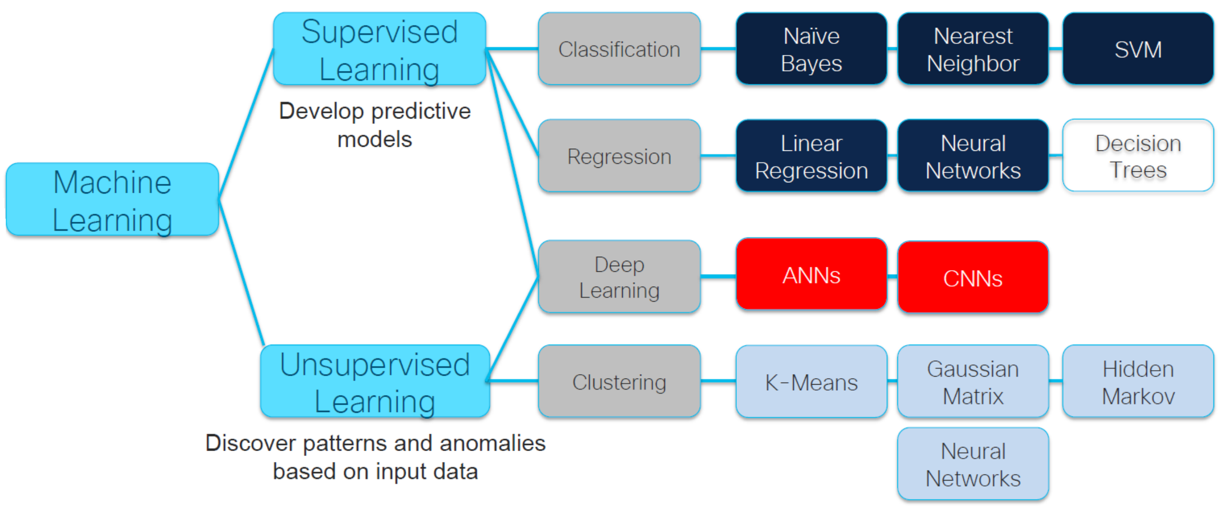


Figure 2. 5. Classification of machine learning [57]

In figure 2.5, shows that machine learning classifications. Machine learning (ML) tasks are classified into two types based on the learning signal of the learning system: supervised learning and unsupervised learning, and discussed as follows.

2.8.1. Supervised Learning

As the first branch of machine learning, supervised learning focuses on learning patterns by linking the association between variables and known outcomes and interacting with labeled datasets (learning with labeled training set) [53] .

Supervised learning operates by feeding computer sample data with different features (represented as "X") and the right output value of the data (represented as "y"). The assumption that the output and function values are known describes the dataset as "marked. The algorithm then deciphers patterns that appear in the data and generates a model that can replicate the same underlying rules with new data [53]. Examples of supervised learning are classification (the outputs are discrete label) and regression (where its outputs are real-valued).

2.8.2. Unsupervised Learning

Not all factors and data patterns are categorized in the case of unsupervised instruction (discovering patterns in unlabeled data) [53] [58]. Instead, the computer must discover secret patterns and create labels using unsupervised learning algorithms. Examples of unsupervised learning involve clustering and dimensional reduction. A common example of unsupervised learning is the k-means clustering algorithm.

There are also other categories of machine learning in addition to the above which is discussed in [58] as follows:

Reinforcement Learning: Reinforcement learning is a machine learning subfield in which algorithms are equipped by collecting simulated "rewards" or "punishments," basically by trial and error learning [53] [58]. It's a semi-supervised learning model and no training data or correct/incorrect instruction needs to be provided.

Semi-supervised learning: where only a portion of the training knowledge is labeled this is categorized under.

Time-series forecasts: Supervised learning can be applied to time series data. We can restructure a time series dataset given a sequence of numbers to look like a supervised learning problem. This can be accomplished by using previous time steps as input variables and the next time step as output variables [59]. For example, in capital markets.

Detection of anomalies: used in factories and in monitoring for fault detection. The unsupervised approach is used for anomaly detection. Using an unlabeled dataset, we train a machine-learning model to fit to normal behavior.

Active learning: a types of semi-supervised learning, means that models are trained using both labeled and unlabeled data [60]. in which it is costly to collect data, and so an algorithm must decide which and many others to acquire training data.

In [61],[62],[63],[64], explains that Classification, regression, clustering, and association problems can all be solved with machine learning. To apply these machine learning algorithms to specific tasks, different parameters must be considered. As a result, the first parameters are the types of problems we will solve. On this basis, supervised machine learning can solve classification and regression problems, while unsupervised machine learning can solve clustering and association problems. As a result, depending on the type and nature of the data, as well as the type of problem, we must choose appropriate supervised, unsupervised, and semi-supervised machine learning techniques to apply machine learning algorithms. As a result, a supervised machine learning algorithm is selected for discussion because based on the areas where supervised learning is used like types of the problem that we want to solve is the classification problem. This algorithm is appropriate for our purposes. As previously stated, supervised machine learning can be used to solve image classification, risk assessment, fraud detection, and visual recognition problems [65]. As a result, the researcher chose for discussion the most well-known supervised machine learning algorithms used to solve recognition problems.

2.9. Deep learning

Deep learning is a virtual computer that mimics the brain's neuron network. It is a branch of machine learning and, since it makes use of deep neural networks, it is considered deep learning. To learn from the results, the computer uses numerous layers. The model depth is defined by the number of layers within the model. In terms of AI, deep learning is the latest state of the art. The learning process is achieved by a neural network in deep learning [66].

Deep learning methods allow vast amounts of raw data to be fed to a computer and the representations required for identification or classification to be discovered. Methods of deep learning focus on several layers of data representation with successive transformations that amplify aspects of the information that are necessary for discrimination and remove unnecessary variations. It is possible to supervise or unsupervised deep learning. Many of the latest fundamental developments in machine learning have been responsible for deep learning approaches [52]. Deep learning uses a neural network between input and output, with many layers of nodes. In a sequence of stages, the series of layers between input and output feature recognition and processing, just as human brains execute their task.

Deep learning is a function of artificial learning that functions the same way the human brain does. It processes the data and generates patterns that allow decisions to be taken and data collected based on the patterns generated. Deep learning is considered a component of machine learning, consisting of networks that enable unsupervised learning from unstructured or unlabeled knowledge. Deep learning is popularly referred to as deep neural learning. It has implementations in fields such as fraud prevention, money laundering, and more.

Deep learning or hierarchical learning is based on the layers used in every artificial neural network and is part of general machine learning. It is possible to supervise, semi-supervised, or unsupervised deep learning. It depends on how one understands to get off with the reasoning and fundamentals. Deep learning is focused on functions and algorithms that allow complex problems to be solved effectively.

Deep learning helps multi-layer cognitive models to learn and represent data with several layers of abstraction that simulate how multimodal information is interpreted and understood by the brain, while indirectly capturing complex large-scale data structures. Deep learning is a diverse family of tools, including neural networks, hierarchical probabilistic structures, and a number of unmonitored and supervised algorithms for functional [67].

2.9.1. Deep Learning Approaches

We need to explore how deep learning is used to solve real-life problems in this chapter and what solution should follow would mean that we can discuss deep learning approaches.

Deep learning, as discussed in the above section of this article, is an aspect of machine learning and artificial intelligence that makes a machine intelligent in learning, thinking, and making decisions by itself without human intervention, by imitating human behaviors by computer algorithms and by accepting input data to learn and train itself. Each information reaches a neuron and is multiplied by the weight. The multiplication result flows to the next layer and becomes the input. For each layer of the network, this step is replicated. The output layer is called the final layer: it contains a real value for the regression task and a likelihood for the classification task for each level. A statistical algorithm is used by the neural network to change the weights of all the neurons. If the value of the weights gives an output similar to the fact, the neural network is fully qualified.

2.9.1.1. Convolutional Neural Network (CNN)

A convolutional neural network, also known as a convnet or CNN, is an artificial neural network variant that specializes in emulating our visual cortex's functionality and behavior. CNN is a multi-layered neural network with a special architecture designed to derive more and more complex data characteristics from each layer for performance Convolutional neural networks use large volumes of data to acquire a complex interpretation of visual data. They are inspired by the human visual system and learn many layers of transformations that are added on top of each other to eventually extract a more advanced representation of the information [68].

A CNN is composed of one or more convolutional layers (matching those of traditional artificial neural networks) with completely linked layers on top. It also utilizes tied weights and layers for pooling. This architecture enables CNNs to take advantage of the input data 2D structure. Convolutional neural networks are beginning to produce superior performance in both imaging and speech applications compared to other deep architectures [69]. CNN’s are well suited for perceptual tasks. CNN's typically consist of the following three components [70][71].

The convolution function is defined as the integral of the product of two functions after one is reversed and shifted.

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\iota).\, g(x - \iota)\, d\iota$$

Where ‘*’ denotes convolution and ‘.’ Denotes multiplication

The Convolution layer is the first layer of several convolutional layers and extracts the features from an input picture. To define attributes, it uses an array known as a Filter or Function Extractor. An activation function is introduced in the convolutional layer that allows to achieve non-linearity and to get an input-based output. ReLU, Sigmoid, and tanh, etc., may be the activation functions. ReLU is the most widely used variant of CNN. This consists of several filters that are transformed by essentially computing a dot product to give a two-dimensional activation map over the height and width of the input data (e.g., image raw pixels) [71][72]. On stacking all the maps across all the filters, we end up getting the final output from a convolutional layer.

Sigmoid function: $s(x) = 1/1 + e^{-x} = e^{x/(e^{x}+1)}$

Tanh function: $\tan\mathrm{h}(x) = \left(e^{x} - e^{(-x)}\right)/\left(e^{x} + e^{(-x)}\right)$

ReLU function: $RELU(x) = \begin{cases} 0, if\ x < 0 \\ x, if x \geq 0 \end{cases}$

Pooling layers, which are essentially layers doing non-linear down-sampling to decrease the input size and amount of convolutional layer output parameters to further generalize the model, avoid overfitting and minimize computation time. Filters go into the input heights and width and decrease by taking an aggregate such as total, average, or max. Average or peak pooling is a common pooling element.

Fully connected *MLPs* to perform tasks such as image classification and object recognition.

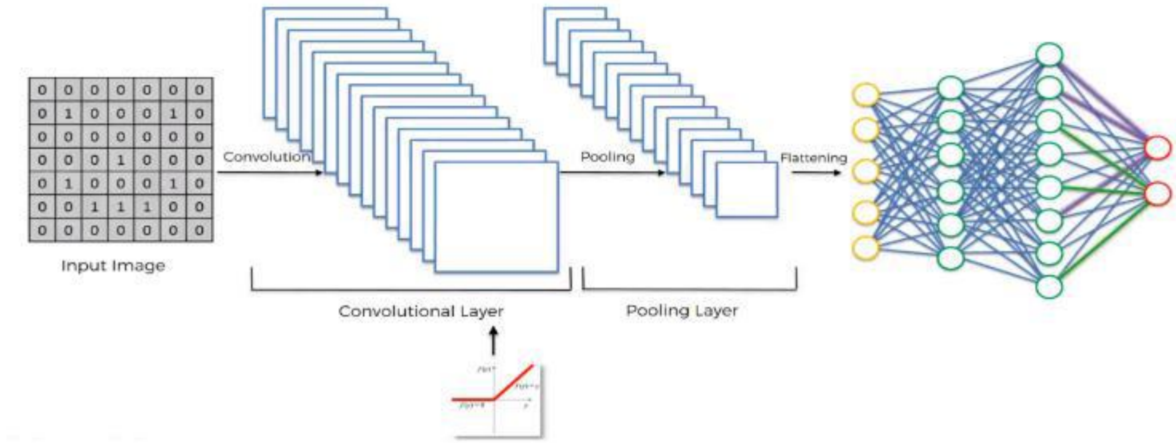


Figure 2. 6. Components of convolutional neural network [71]

1D array when fed into the completely linked network after flattening the extracted function diagram. This operates much like any other neural network and is faced with the most suitable classification after any backpropagation. Other than image interpretation, CNN is often used to turn handwritten records into digital text for optical character recognition, which is part of Natural Language.

2.9.1.2. Recurrent Neural Networks (RNN)

RNN is a multi-layered neural network that can store data in background nodes, enabling data sequences to be learned and a number or another sequence output. It is a special type of neural network that, using a special type of looped architecture, enables persistent information based on previous experience. In basic terms, it's an artificial neural network whose loops contain interactions between neurons. For processing sequences of inputs, RNNs are well adapted. Such looped networks are called recurrent since, in a sequence of input data, they perform the same operations and computation for every element. RNNs have a memory that helps to capture previous sequence knowledge [73].

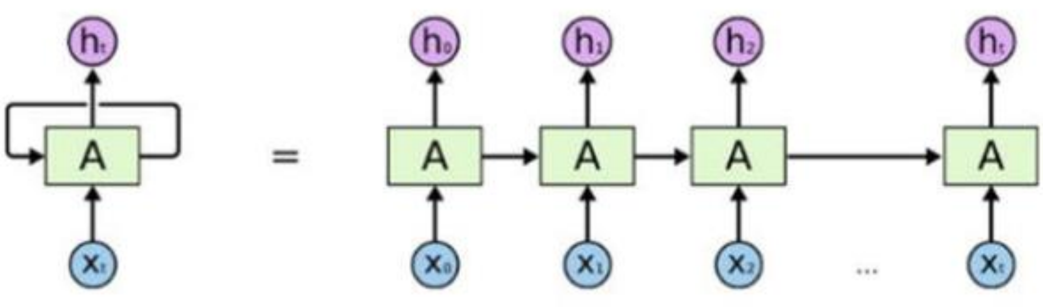


Figure 2. 7. Recurrent neural network diagram [73]

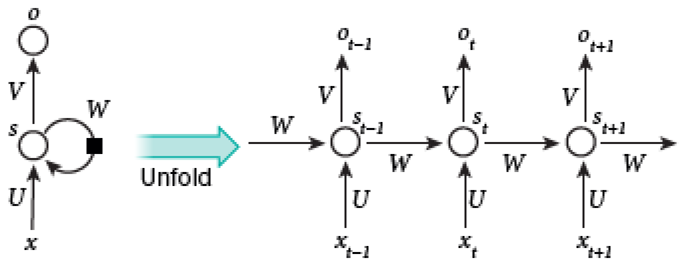


Figure 2. 8. Unfold RNN [73]

A recurrent neural network and the computational progress involved in its forward processing in time. Artificial neurons (such as secret units clustered under node s with S_t Values at time t) obtain signals from other neurons at previous stages in time (this is represented with the black square, representing a delay of one-time step, on the left). In this way, an input sequence of elements X_t can be mapped by a repeating neural network into an output sequence with elements O_t, with each O_t based on all the previous $x_{t'}$ (for $t' \leq t$). At - point in time, the same parameters (matrices U, V, W) are used. Many other configurations are feasible, including a version in which a series of outputs (e.g. words) can be created by the network, each of which is used as inputs for the next level [74].

2.9.1.2.1. Long Short-Term Memory Network (LSTM)

LSTM is a particular architecture of the recurrent neural network (RNN) that was developed to model temporal sequences. LSTM has a long-range dependency that makes LSTM more specific than regular RNNs. The RNN design backpropagation algorithm triggers the problem of error backflow. LSTM, unlike RNN, includes special units in the repetitive hidden layer called memory blocks. In addition to special multiplicative units called gates, memory blocks contain memory cells with self-connections that store the temporal state of the network to regulate the flow of information [75]. Each memory block in the original architecture contained three gate types which are namely:

Input gate: The input gate governs the flow into the memory cell with input activations.

Output gate: The output gate regulates the output flow through the rest of the network of cell activations.

Forget gate: scales the cell's internal state before introducing it to the cell as an input via the cell's self-recurring connection, thereby adaptively ignoring or resetting the memory of the cell.

Furthermore, to learn correct output timing, the current LSTM architecture requires peephole connections from its internal cells to the gates in the same cell. The LSTM architecture is always unfolded over t (time)-dimension to make interpretation simpler, which can be represented by the following diagram.

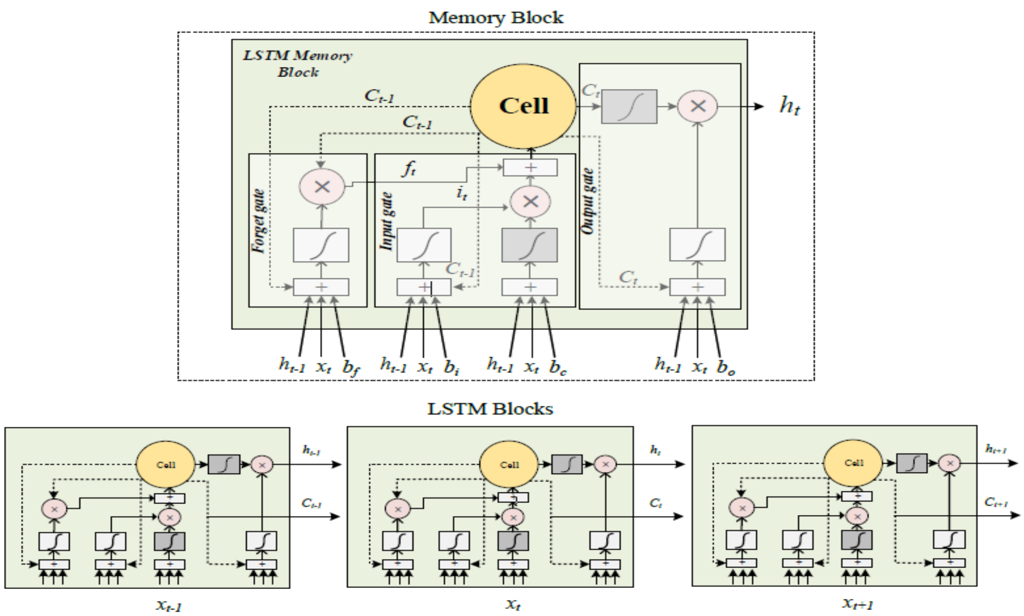


Figure 2. 9. Structure of LSTM [76]

A basic one-layer neural network manages the forgotten gate in the memory block structure. This gate's activation is determined as follows.

$$ft = \sigma(W[Xt, ht - 1, Ct - 1] + bf)$$

Where x_t is the list of inputs, the previous block output is h_{t-1}, the previous LSTM block memory is C_{t-1}, and the bias vector is bf. For each input, W represents different weight vectors and σ is the function of the logistic sigmoid. The sigmoid activation function, which is the output of the forgotten gate, is extended by element-wise multiplication to the previous memory block. The degree to which the previous memory block on the new LSTM would be successful is thus calculated. If there are values close to zero in the activation output vector, so the previous memory will be forgotten.

The other gate, the input gate, is a segment where a simple NN with the tanh activation function and the previous memory block effect produces the current memory. The following calculations measure these processes.

$$it = \sigma(W[xt\ , h(t - 1), C(t - 1\ \ ]\ \ + bi)$$

$$Ct = ft.Ct - 1 + it.\tan\mathrm{h}(W[xt, ht - 1, Ct - 1] + bc)$$

Finally, the output gate is the segment where the current LSTM block output is generated. As in the following equations, these outputs are measured as

$$Ot = \sigma(W[Xt, ht - 1, Ct] + bo)$$

$$h_t = \tanh(C_t).\ Ot$$

2.9.1.2.2. Bidirectional LSTMs

A bidirectional RNN (BRNN) is a paradigm suggested for eliminating traditional RNNs separate constraints. Standard RNN neuron states are broken into forward and backward by this model. There are two separate recurrent networks, forward and backward, in other words. To produce output information, these two networks link to the same output layer. In this system, sequential inputs in a time frame are measured without delay in both past and future scenarios [77].

The BRNN structure's LSTM variant is called Bidirectional LSTM (BiLSTM). In classification systems, this version will boost the efficiency of the LSTM model. Two separate LSTM networks are educated in the BiLSTM architecture for sequential inputs, unlike the traditional LSTM framework. A simple BiLSTM architecture operating on sequential inputs is seen in the following figure[77].

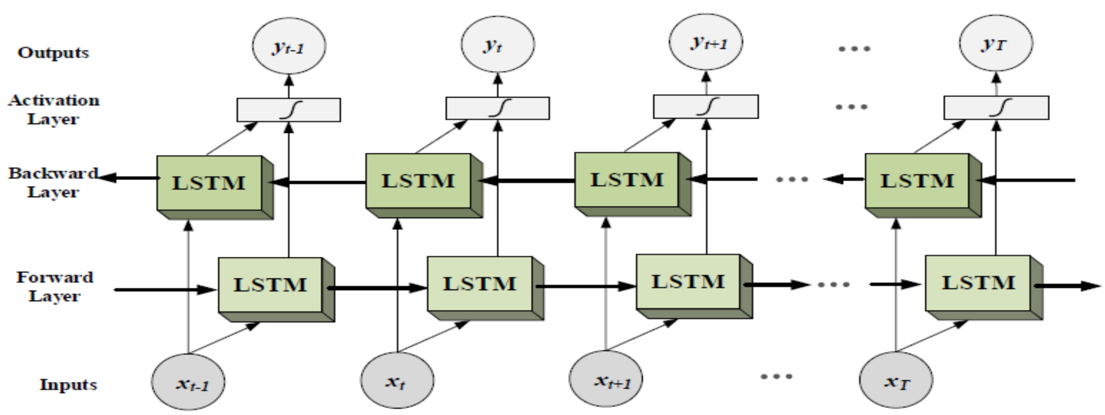


Figure 2. 10. Structure of BiLSTM [77]

Neurons function as unidirectional LSTM systems in the forward state of BiLSTM. Since the neurons are not bound to each other in both networks, network training can be given as a standard unidirectional LSTM. Generally, the preparation protocol for such networks is as follows. All input data for a time slice ($1 \leq t \leq T$) is run via BRNNs in the forward pass and all expected outputs are calculated. Forward transfers are completed for time only from t= 1 to t= T and backward from t= T to t= 1. For the output neurons, too, a forward pass is done. For the time slice ($1 \leq t \leq T$) that is used in the forward pass, the error function derivative is determined in the backward pass. For output neurons, a backward pass is done and for the forward states from t=T to t=1 and backward states from t=1 to t=T, a backward pass is done. Finally, it updates all weights. Depending on the problem area, BiLSTM structures can provide better outcomes than other network structures.

2.10. Related Works

Alula [78], the study was aimed to separate text area from non-text area from mobile captured image and propose method to separate text from document image. But cellphones have various limitations, such as blur, lightning condition, alignment, and geometric distortion. The study was attempted to apply pre-processing techniques such as skew detection and correction, perspective rectification, noise removal, image binarization, and text region extraction to a test data set which the researcher collects. The experiment is selected using different methods of Skew and Perspective correction and based on document boundary correction, this technique is supported by the user by putting the four vertexes of the document image. Based on the experimental three noise removal techniques, the Wiener filter with 1.92 MSE and 48.99 PSNR and the three binarization techniques, Sauvola with 0.13 MSE and 57.62 PSNR performed best with the highest PSNR and lower MSE. And for text area extraction, a modified connected component and dilation process is used. The method proposed identified the text area with 93.60% accuracy and 99.99% recall.

Abay [79], Tried to apply OCR approaches to the recognition of Amharic printed documents for the Amharic character recognition system in real life. He took from real-life documents his training and testing data: The Holy Bible, the popular Ethiopian fiction ' Fiker Eskemekabir,' the newspaper ' Addis Zemen ' and the Federal Negarit Gazette. Before the segmentation algorithms were applied to the scanned images, the researcher first tried to remove noises found in many real-life documents. He tested three algorithms for noise removal (Linear filtering, Median filtering, and Adaptive Wiener filtering algorithms) and found that Wiener filtering works well for most of the Amharic documents thus considered for his job. It was chosen because edges and other high-frequency parts of a character were retained. And Otsu and adaptive thresholding Two-Dimensional Variance Adaptive Thresholding of Wavelet Coefficients were tested for binarization global thresholding. Form the Otsu binarization method of the experiment was found to be more effective in isolating the text pixels from the background pixels. The researcher recorded, using this and other recognition technology, an average accuracy of 96.87% in the training set and 11.40% in the test set. The researcher recommended further exploration in noise removal algorithms to increase recognition rates as some of the Amharic characters are structurally similar and Amharic document images are highly degraded.

Biruk [80], the study was focused on how we restoration and retrieval of historical documents specially the researcher focused on Amharic historical document. For as much examine diverse pictures rebuilding systems need aid experimented, for example, Dilate, dissolve What's more blending about scientific morphological tenet systems and additionally Haar, Daubechies, Furthermore Symlet wavelet systems. These systems are tried different things previously, authentic documents, and besides genuine documents. Execution dissection reveals that the best result will be acquired by joining together scientific morphological tenet for Otsu thresholding. Finally, the execution of the framework will be assessed previously, then after the coordination of the chosen restoring systems on which a normal in general execution from claiming 87. 02 % F-measure will be enrolled over documents Hosting low, the medium also large amounts of corruption for a change about recovery adequacy Eventually Tom's perusing 4. 65 % F-measure. The execution enlisted in this consider indicates guaranteeing to bring about shortages to outlining appropriate Amharic document image recovery. That major test is unapproved unlucky deficiency about the institutionalized corpus and the dataset holds a set number from claiming verifiable archive pictures. Therefore, later on, an institutionalized corpus ought to further bolster to be arranged what's more utilized for experimentation over comparative investigations.

Fetulhak [3], this study aimed to develop and proposed recognition model for Amharic hand writhing using deep learning algorithm. the researcher was implemented one of deep learning algorithms which is convolutional neural networks. In the Handwritten Amharic character recognition method, the incoherence of the text, inconsistency in the various writer's styles, the relative number of characters in the script, high interclass similarity, structural complexity, and document degradation are all difficult tasks. A new method based on profound neural networks has recently demonstrated exceptional performance in different pattern recognition and machine learning applications to recognize handwritten Amharic characters. Their database containing 132.500 datasets of hand-written Amharic characters evaluates the Convolutional neuronal network model for performance. The combination of feature extractors and classifiers is used by common hand-written machine learning recognition systems. The use of deep learning techniques currently shows promising improvements for classification tasks based on machine learning. In training data and validation data, 91.83 percent and 90.47 percent respectively of the proposed CNN model is accurate.

Sertse [13], in this study the researcher aimed to develop and proposed recognitions model for mixed Amharic and English printed material in bilingual script identification. In this study, a system is presented which identifies Amharic and English scripts from an image of a document. At the word level, the system tackles the problem of language identification. Preprocessing tasks such as noise reduction, binarization, segmentation, measurements, and design normalization function are carried out when the essential feature meaning of word image in the scripts is extracted. Maximum horizontal projection profiles from the three selected regions, the size of the word image, and the ratio of the number of connected components to the width of the word image are important features to distinguish the two languages of the script. The Support Vector Machine algorithm is used to classify the word images for new instances. A large number of words with various font styles and dimensions will be tested for the suggested algorithm. The findings obtained and the main reasons for the word image misclassification. In the case of the non-noise document image the classifier generally achieved an extremely good accuracy of classification; for the Amharic document image at least 95.78% and a maximum of 97.77%. The accuracy rates for the classification of the English Document image are similar to the Amharic image of the document, hit and minimum 96.0% and max. 97.55%.

Siranesh [12], the aim of the study was to recognized and proposed recognition model for ancient Ethiopian Manuscript Recognition using a deep-learning approach with artificial neural network. The study tries to discussed on how we design and implementation deep neural network for the recognition of ancient Ethiopian manuscript character. Deep learning is employed and trained by the Restricted Boltzmann Machine (RBM), a greedy layer-wise, unsupervised training technique. In each step, efficient and efficient algorithms have been selected and implemented. The system consisting of 24 Ge’ez alphabet base characters, each with 100 frequencies, was also compiled with a dataset. Overall, 93.75 percent recognition accuracy with 3 hidden layers of 300 neurons was achieved. Results from each step of the recognition process are analyzed to show that the system can be extended and refined for practical use.

Seble [81], has proposed recognition model for Ethiopian car plate recognition. During the study, one of the key elements of ITS named Car Plate Recognition (CPR) is being developed for Ethiopian car plates. The proposed system has three main modules: Plate Detection, Character Segmentation, and Character Recognition. The template matching method based on correlation is used for character recognition. In addition to the value of the correlation, the recognition process is supported by color analysis techniques and character location information. MATLAB TM is used to produce the prototype of the system, and its performance is tested in 350 RGB car pictures taken at various angles, distances, motion, and illuminations. The system developed results in a 63.14 percent accuracy and can also detect a plate between 2-5 seconds depending on whether or not after processing is needed.

Nigus Kefyalew [15], has proposed a model for Amharic Sign Language Recognitions based on Alphabet Signs and focused on development of an automatic Amharic sign language translator, which translates Amharic alphabet signs into their corresponding text using a digital image processing and machine learning approach. The system input is the video frames of the Amharic alphabet and the output of the system is the Amharic alphabet. Used for Classification Model building Neural Network and Multi-Class Support Vector Machine. Finally, with 57.82 percent and 74.06 percent, the identification system can recognize these signals with NN and SVM respectively. The classification efficiency of the Multi-Category SVM classification was, therefore, higher than that of the NN classification.

Research by Eyob [82] was conducted using deep learning approach on Handwritten and machine printed document image for Ge'ez number and it was carried out with Artificial Neural Network algorithm. Ethiopian scripts were researched. Nevertheless, the Ge'ez numbers were omitted for several reasons from research. This article provides an offline recognition of the number of Ge'ez handwritten and machine-printed by feedforward artificial neural propagation network. In this analysis, different picture characters from the Ge'ez scan were collected and three individuals were directed by the pencil to write the numbers. They collected 560-character numbers in all. 460 characters have been used for training and 100 for testing. They have thus achieved a total of 89.88 percent of classification.

To improve the effectiveness of relevant document hospitalization from printed real-life images Amharic documentation, the users accepted several word requests, Biniam [83], attempted to develop an actual image retrieval Amharic document image through integrated effective image preprocessing techniques to reduce the noise level. He uses 4,974-word pictures from realistic books, magazines, newspapers, and regulations with different font types, sizes, and types. Biniam [83], explicitly evaluates three noise removal techniques: medium filters, Wiener filters, and adaptive median filtering (AMF) for cleaning of noise in real-life printed document images. And from his experiment, the Wiener filtering document image is better than the median and adaptive median filtering technologies in terms of reduction of noise and no blurring. To binaries, Otsu implements and assesses global thresholding and local thresholding methods of Niblack and Sauvola. The best image of the document is the one with the Otsu threshold based on experiments. The system's output has a recall of 96.67% and a precision of 77.24%.

All researchers conduct their own best work, but the only thing that they have not focused on during their studies is that some of them work only on scripts that are machine printed, and some of them work on machine and handwritten manuscripts, while others focused only on Ge'ez number recognition on handwritten and machine printed ones. This study aims to address the issue of ancient manuscript document images from handwritten scripts, such as parchment and paper books, which are extremely poor in quality and noise, and the study will cover the basic characters of the two languages, Geez and Amharic, as well as geez numbers.

1.10.1. Summary of Related Work

#	Author	Topic	Algorithm used	Result	Remark
1	Fitehalew Ashagrie (2019)	Ancient Ge'ez script recognition using deep learning	CNN	Accuracy of 99.39% with a model loss of 0.044	Not consider numbers
2	Fetulhak Abdurahman (2019)	Handwritten Amharic character recognition using CNN	CNN	Accuracy of 91.83% on training and 90.47% on validation data	Not consider numbers
3	Eyob Gebretinsae Beyene (2019)	Handwritten and Machine printed OCR for Ge'ez Numbers Using ANN	ANN	Recognition Accuracy 89.88%	Focused on handwriting and machine printed only on Ge'ez number
4	Abel Teshome Hurisa (2020)	Handwritten ancient saint yared hymn script recognition system using artificial neural network	CNN	Recognition Accuracy of 85.4%	Low percent of recognition accuracy
5	Shiferaw (2017)	Optical character recognition for Ge'ez scripts written on the vellum	SVM classifier	Accuracy 63.4% and 51.7% on noise free and noisy images, respectively	Low percent of accuracy
6	Abay Teshager (2010)	Amharic character recognition system for printed real-life documents	ANN	Recognition rate of 96.87% for the test sets from the training sets and 11.40% new test sets	Poor testing accuracy

Table 1. 1. Summary of Related Work

CHAPTER THREE
DESIGN OF ANCIENT AMHARIC AND GE'EZ MANUSCRIPT RECOGNITION

3.1 Overview

This section, provides a detailed explanation of the research design methodology and the research framework of the study. As well as we have also discussed techniques of image collection/acquisition of data for this study and the way of data preparation. And this section also provides explanations on different image data preprocessing techniques. Also, we have covered the parameters and what will look like the recognition model is and how its design and how we implement the proposed model also covered with is chapter.

3.2 Research Design

Research design is the crucial part of the research as it includes all four important considerations: the strategy, the conceptual framework, the identification of whom and what to study on and the tools and procedures to be used for collecting and analyzing data. So, in this research, we have selected design science research methodology for Ancient Ge'ez and Amharic manuscript recognition among different types of research design methodology to address the objective of the study. And the reason why? We select this type of methodology is explained in the following section.

3.2.1 Design Science Research Methodology

In this study, design science research (DSR) methodology has been selected as an approach for discovering and identifying opportunities and problems of Ethiopian ancient manuscript recognition to creating new artifacts to overcome those challenges which are happened in the subject area of the study. DSR was chosen as the philosophical approach due to the following reasons [84].

Much of the academic research related to organizational learning has used varied methodologies to study organizational learning such as, exploratory, quantitative studies that employ surveys, action research using case, action research using detailed literature review etc. The objective of most of these studies was to create a shared understanding of organizational learning.

The emphasis of DSR is practical problem-solving and encompasses narrow or solution-oriented knowledge where the outcomes from scientific reasoning (forecasting, recognizing or clarifying phenomena) can be employed in devising solutions to intricate and appropriate everyday issues. Thus, it is an approach driven by problems encountered in real-life and is also solution-oriented. In other words, DSR helps in drawing attention to the challenges in a field from a solution-based viewpoint. Hence, DSR can be used to develop knowledge that can be useful to professionals in the field under consideration to design solutions to their everyday problems by illustrating and examining other strategies to address such problems.

The design science as a paradigm has its root in engineering and science of the artifact, its fundamentally on solving a problem through creative innovations which define the ideas, practices, technical capabilities, and products in which analysis, design, implementation, and information system use which can be effectively and efficiently reached.

Research Methodology This is seen as a concern with a way of thinking on and studying a specific phenomenon of interest by a researcher [85]. Research methodology is also seen as an action plan, strategy, process, or design laying, behind the choice of and methods and linking the choice of methods use [85]. This just a little of the definition from the many views of different authors in Information Science, and Computer Science next, we will conduct an explanation of the concept.

The Design science process includes six steps: problem identification and motivation, the definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. So, for this study, we have adopted the six-design science research process to achieve the objective of the study. But before we have to discuss each design science process it is better to discuss the Design science research methodology framework in information systems and the DSRM process model which is adopted from [86].

The design science research methodology (DSRM) proposed by Peffers et al. [87][88], is consistent with earlier literature and offers a procedural model for presenting the DSR. According to the authors, this process model supports researchers, and studying the design science paradigm in this way is beneficial. In this section, we defend [87][88]'s DSRM process model.

According to Peffers, et al. [87], given these starting grounds, the problem-centered initiation, the objective-centered solution, and the development-centered solution constitute the research's entry points. Six actions make up the DSRM process model, which covers the entire study from the beginning (motivation) through the conclusion (communication). According to Figure 3.1 below, these steps are problem identification, objective definition, design and development, demonstration, assessment, and ultimately communication.

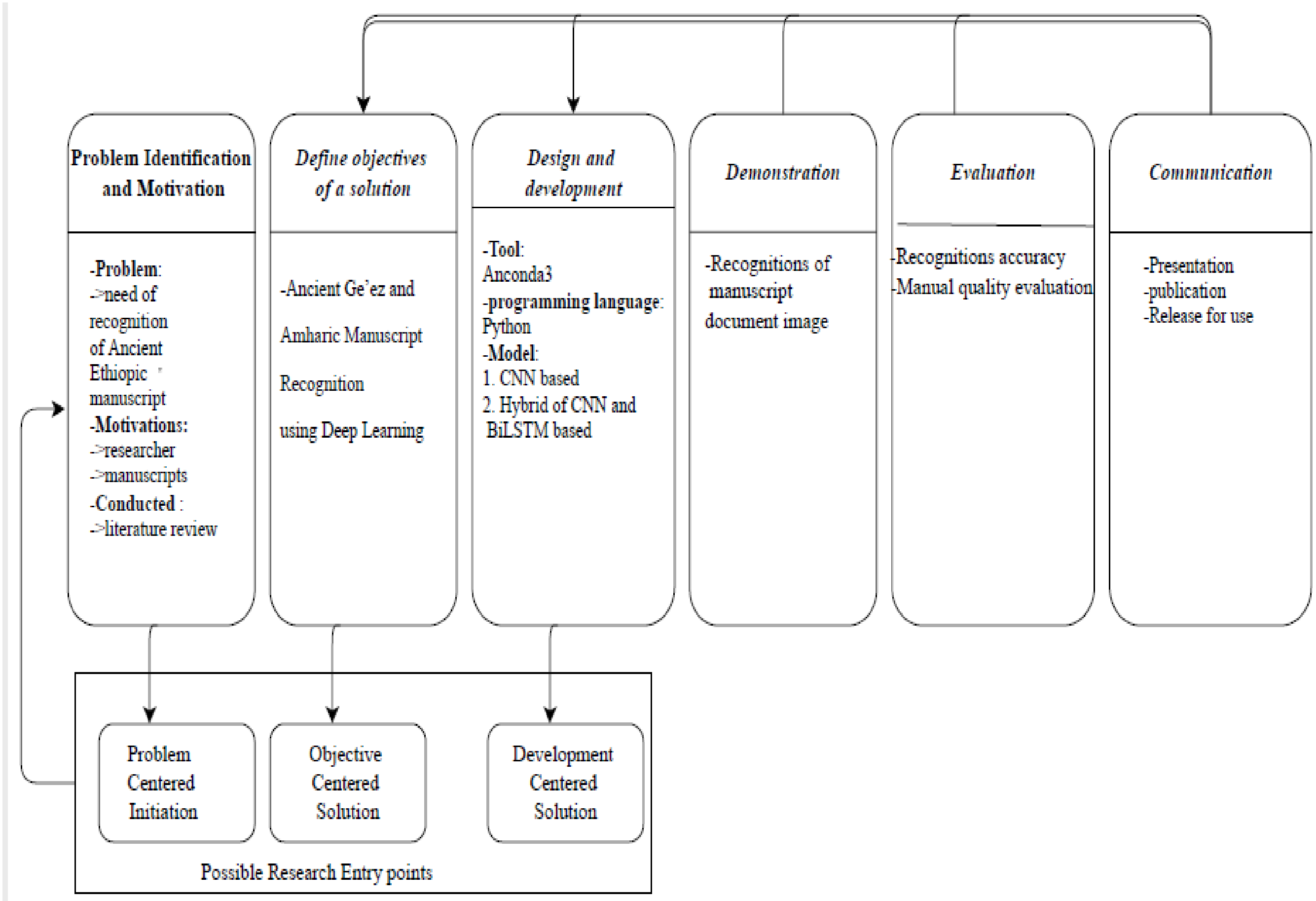


Figure 3.1. Design science research methodology process model [86] [87][89][88]

As we have seen from the figure the researcher initially understands and study the real world and identify the problem of the study area and define the identified. And the researcher tries to determine the solution for identified problems based on existing knowledge. Then in the development stage the artifact or design is developed to enhance the solution for the identified problem and here the actual deliverables are developed. Then the developed deliverables will be evaluated based on different performance measure criteria and finally researchers they can concluded the result of the study and add new knowledge on the existing knowledge on the concept matter of the study.

Identify problems and motivations: in this stage the researcher tries to read different related works as a literature review on Optical character recognition as well as Ethiopian ancient manuscripts and its recognition. And the researcher tries to compare and contrast related works and researchers have identified problems, gaps and then the researcher motivated to fix problems, gaps and motivated to design deep learning based Ancient Ethiopian manuscript recognition for Ge'ez and Amharic Ethiopian manuscripts.

Define objectives of a solution: The next stage after defining the problem is determining the goals of a solution and choosing a workable one. For instance, asking how the proposed manuscript recognition system benefits society can lead to the goal of a solution.

Design and development: here researcher deign different new artifacts for the recognition of ancient Ethiopian manuscript. And among different model best alternative design leads for development.

Demonstration: In this stage, it is demonstrated that systems or artifacts were used to address the reported issue. This could involve simulations, experiments, evidence, or another suitable stage in many academic disciplines.

Evaluation: here evaluation of the developed model is conducted to cross check the objective of the study and the solution is achieved or not.

Communication: is the process of discussing the problem. Design and its effectiveness with audience and researchers and processionals. So, the researcher tries to present the developed artifacts for different stakeholder or the domain area experts in order to grape different feedback.

Image acquisition in computer vison and image processing is a crucial and the first stage to move forward to further processing. This is the first and prior task that we should have to conducted to convert the original manuscript into digital and the process is called image digitization.

3.3. Acquisition of Amharic and Ge'ez Manuscript

It is the process of retrieving image from an external source for further processing. It's always the foundation step in the workflow since no process is available before obtaining an image. It is the process of capturing an unprocessed image from an object or scene by an optical device into a manageable form for processing and analysis purposes. To develop the proposed Amharic and Ge'ez manuscript recognition model, we were collected lots of different ancient manuscripts of Ge'ez and Amharic language from Ethiopian Orthodox Church via direct image capturing via digital camera. The number of images were collected is noted in the following table.

Manuscript type	*Number of images collected*
Ge'ez manuscript	*515*
Amharic manuscript	*190*
Total number of document image	*705*

Table 3. 1. Number of Collected data Information

As we have discussed in table 3.1 the total number of collected number of manuscript document images, we have gathered in total number of document image is that 705 before we have performing any types of image processing techniques. From those number of collected document images are 515 number of document images are captured or take from Ge’ez document manuscript and the rest 190 of images are capture from Amharic language ancient Ethiopian manuscript documents.

Therefore, those collected images were used for further analysis or on these images we apply different image processing techniques until they provide the required information for the development of the proposed model.

3.4. Proposed System Framework

This research work was concerned with designing Ancient Ethiopian manuscript recognition model for Ge’ez and Amharic language that has trained on deep learning algorithms via convolutional neural network and BiLSTM. The designed model is trained with manuscript document image. Then the trained model and an outperform model while we are trained has used for the recognition of ancient Ethiopian manuscript document image.

The proposed optical character system consists of a series of tasks such as digitization, binarization, noise removal, line and character segmentation, size normalization, feature extraction, training deep learning algorithms that has used for model development and character recognition. Figure 3.3 illustrates the proposed system's overall architecture. Each task and subtask are discussed in detail in the following section. Techniques for binarization, noise removal, size normalization, and location detection are covered in the preprocessing stage. The segmentation stage deals with line and character segmentation. Finally, tasks involving feature extraction and recognition are presented.

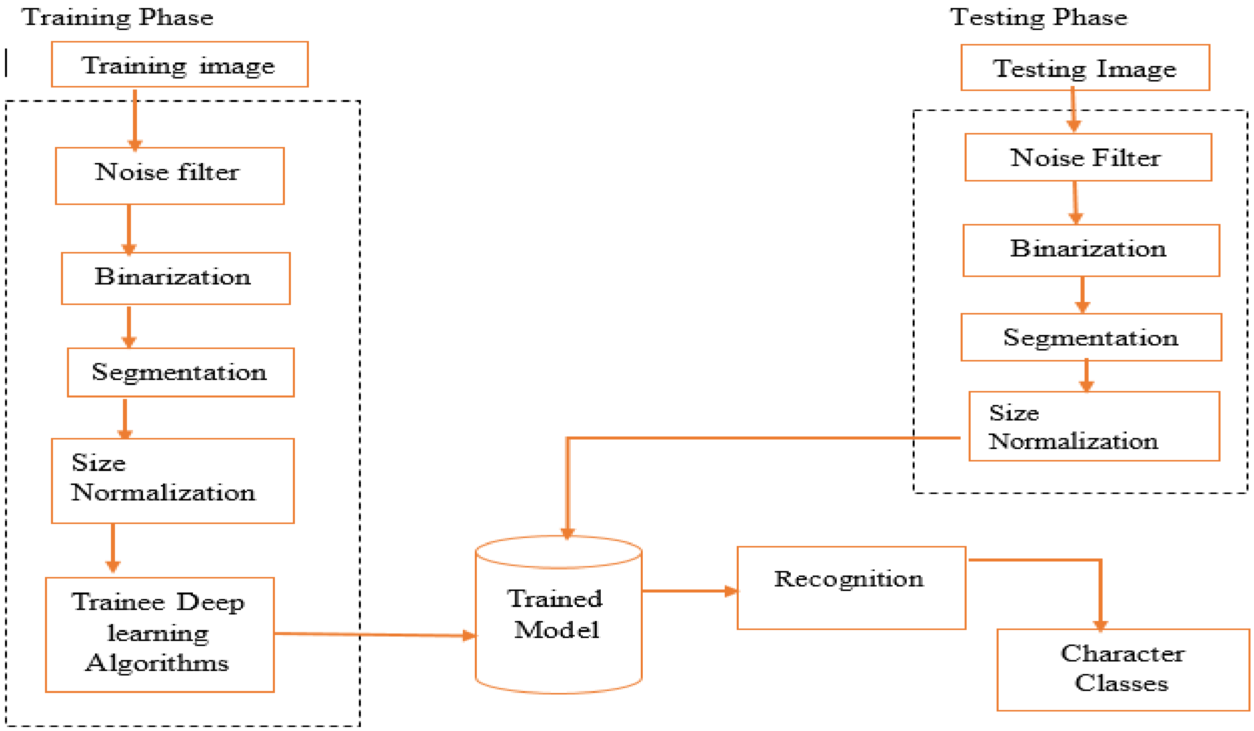


Figure 3.2. Proposed System Framework adapted from [4] [20]

3.4.1. Digitization

Digitization is the process of converting a paper document into a machine-readable format or document image. An image is optically captured during digitization. The captured image is then fed into the preprocessing stage. In this study, the researchers used a Samsung A71 mobile phone. The phone has a super AMOLED plus FHD+ 6.7" display, a 64 MP wide, 12 MP ultra wide, 5 MP depth, and 5 MP macro camera and a jpg format to digitize the ancient manuscript document.

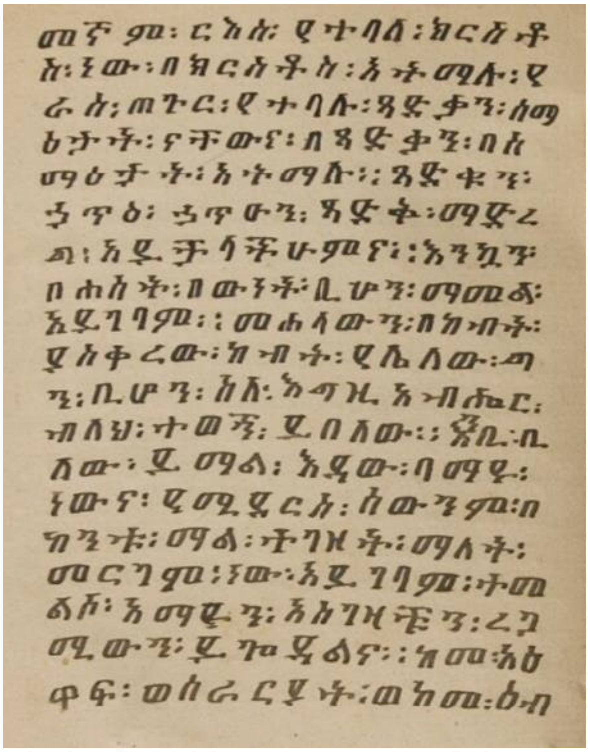


Figure 3.3. sample captured image of Amharic manuscript

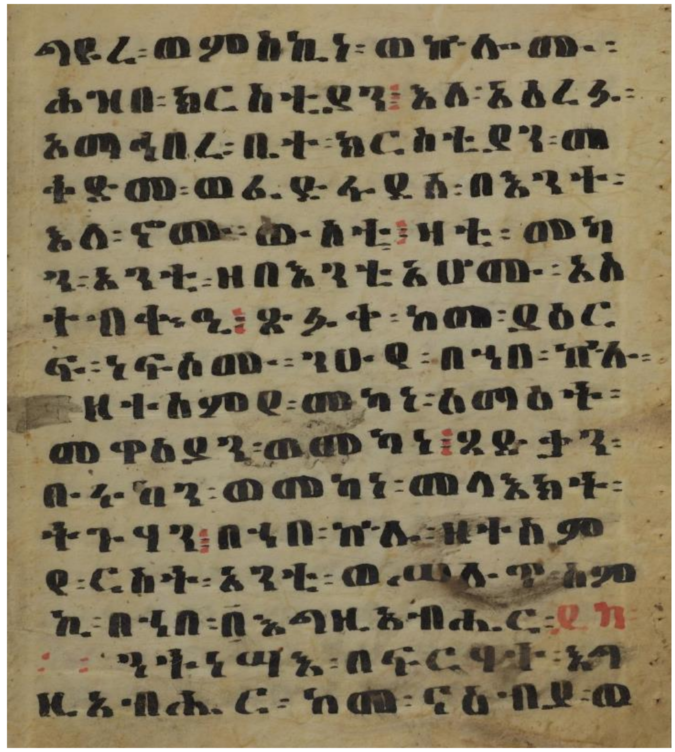


Figure 3.4. sample captured image of Ge'ez manuscript

3.4.2. Image pre-processing

As data pre-processing or data cleaning is a key stage and most of the ML engineers spend a considerable amount of time in data pre-processing before creating the model. Some examples for data pre-processing include outlier identification, missing value treatments and eliminate the undesirable or noisy data.

Similarly, Image pre-processing is the name for actions on pictures at the lowest level of abstraction. These procedures do not enhance picture information content but they diminish it if entropy is an information metric. The objective of pre-processing is an enhancement of the picture data that suppresses unwanted distortions or increases specific visual characteristics essential for subsequent processing and analysis work.

So, to create the recognition model as effective as feasible and to fulfill user needs, the researcher has attempted to inspect the manuscript image to make sure that whether gathered and augmented images have excellent quality and the naturalness as expected or not. Towards that at least low-level image pre-processing is important. Therefore, the proposed character recognition model for this study has followed six steps, First, the images must typically be binarized with different techniques that will discussed on the next time and secondly, to minimizing the effects of the degradations image quality during acquisition we should have perform noise removal task or image denoising to optimized the image quality and image de-skewing in order to correct the degree of image. Thirdly, denoised and de-skewed image passed through other image processing task which is image segmentation during this time it should pass different image segmentation stages. Then resize the segmented image to normalized to fit the input layer and the rest of recognition algorithm operations. Finally, the segmented image passes through our model and recognized what is it.

3.4.2.1. Noise Removal (image denoise)

In this case, noise is unwanted information that is associated with images. In this case, noise is present on manuscript document images.

Noise means that the pixels within the image have different intensity values and incorrect pixel values. Noise has its origin in the physical nature of recognition processes and has many specific forms and causes. It is defined as a process (n) that affects the captured image (f) and is not part of the scene (initial signals). Therefore, the noise model can be written as [90]:

$f(i, j) = s(i, j) + n(i, j)$.

Manuscript documents, by their nature, are associated with different forms of noise more than printed or scanned documents. This is partially due to their deterioration and partly due to the fact that many historical documents are scanned using cameras rather than scanners, which do not have regulated lighting conditions and almost no environmental factors. Because of poor lighting or because the page is not flat, some camera-generated pictures exhibit uneven illumination (e.g., curved edges, bleed through ink, thin strokes) [17].

Due to this case, the document quality has become very low, so to improve the quality of historical documents, image enhancement (noise removal) techniques are applied. Noise removal techniques or algorithms are applied based on the nature of noise analysis, as the types of data and the behaviors may be complete for quantitative analysis of noise. And in this research, researchers applied the following image noise removal algorithms:

A. *Adaptive noise filter*

The adaptive Wiener filter changes its behavior according to the statistical characteristics of the image in the filter window. The performance of adaptive filters is generally higher than that of non-adaptive filters. Mean and variance are two important statistics that can be used to design adaptive filters.

B. Mean noise filter

The mean filter is a simple sliding window in which each pixel value replaced by the mean (average value) of all pixel values in the window. The window or core is usually square but can be any shape.

C. Median noise filter

The median filter is a simple and powerful nonlinear filter. It is used to reduce the amount of intensity variation between one pixel and another. In this filter, we replace the pixel values with the average value. The median value is calculated by sorting all pixel values in ascending order and then replacing the calculated pixel with the average pixel value.

D. Bi-leave noise filter

The researcher applied the two-level noise filtering techniques in order to get a better-quality image. During noise filter stage the researcher applied two level noise filter to achieve objective of the study.

3.4.2.2. Binarization

Many groups have conserved ancient manuscripts in order to protect these texts and regain traditional knowledge. Digitized media is now frequently utilized to record these papers, thanks to advances in computer technology. One goal of this research is to create an effective image processing system that can be used to automatically recover knowledge and information from these old texts. Binarization is a manuscript preparation technique for extracting text and characters [91]. Binarization is the process of converting a colored image (RGB) into a black and white image. The output is then used for further processes such as character recognition and knowledge extraction.

Image binarization is the process of convert a grayscale image to black-and-white, thus reducing the image's information from 256 shades of gray to two: black and white, or a binary image. Image thresholding is a term used to describe this process, which can result in pictures with more than two shades of gray.

Binarization is the process of turning a gray-scale (multi-tone) image to a black-and-white (two-tone) image. To begin the binarization process, determine the gray scale threshold value and determine whether or not a pixel has a certain gray value. If the gray value of the pixels exceeds the threshold, they are changed to white. In the same way, if the gray value of the pixels is less than the threshold, those pixels are changed to black.

Histogram analysis is one of the most used approaches for image binarization. There are two types of binarization approaches which are global and local thresholding methods. The most classic global methods are; Otsu's method which is created by Nobuyuki Otsu [3] and found to be more effective in isolating the text pixels from background pixels from the image without affecting the basic features of the character

[9]. Local adaptive (Niblack's) binarization techniques, on the other hand, modify themselves based on local information and are more adaptable even when processing low-quality texts. Most of them are based on statistical information in some small area, as the method of Niblack and require a prior knowledge for the image content [5] [23].

3.4.2.2.1. Global thresholding methods

These techniques attempt to find a suitable single threshold value (Thr) from the overall image. The pixels are separated into two classes: foreground (text which is black color) and background (white color) [91]. This can be expressed as follow.

$$I_b(x, y) = \begin{cases} \text{Black} & \text{if } I_f(x, y) <= Thr \\ \text{White} & \text{if } I_f(x, y) > Thr \end{cases}$$

Where $I_f(x, y)$ is the pixel of the input image and $I_b(x, y)$ is the pixel of the binarized image.

Let’s discussed some of global thresholding methods.

3.4.2.2.2. Local Thresholding methods

These techniques calculate the threshold values which are determined locally based on pixel by pixel, or region by region. A threshold value (Thr (x, y)) can be derived for each pixel in the image, and the image can be separated into foreground and background as given in expression.

$$Ib(x, y) = \begin{cases} \text{Black} & \text{if } I_f(x, y) <= Thr(x, y) \\ \text{White} & \text{if } I_f(x, y) > Thr(x, y) \end{cases}$$

3.4.2.2.2.1. Otsu’s Method

Otsu's thresholding approach is used in image processing to make an automated binarization level selection based on the form of the histogram. Iterating over all possible threshold values and computing a measure of spread for the pixel levels on either side of the threshold, i.e. pixels that are either in the foreground or background, is Otsu's thresholding technique. The goal is to identify the threshold value at which the total of the foreground and background spreads is the smallest [92].

Steps for Otsu’s method

1. Separate the pixels into two clusters according to the threshold.

$$q1(t) = \sum_{p=1}^{t} p(i) \ and \ q2(t) = \sum_{p=t+1}^{t} p(i)$$

Where P represent the image histogram

2. Find the mean of each cluster

$$\mu1(t) = \sum_{j=1}^{t} ip(i)/q1(t) \ and \ \mu2(t) = \sum_{j=t+1}^{I} ip(i)/q2(t)$$

3. Calculate the individual class variance.

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu1(t)]^2 * \big(p(i)\big)/\big(q1(t)\big) \ and \ \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu2(t)]^2 * \big(p(i)\big)/\big(q2(t)\big)$$

4. Square the difference between the means

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = q1(t)[1 - q1(t)][\mu1(t) - \mu2(t)]^2$$

5. Finally, this expression can satisfy be maximized and the solution is ‘t’ that maximizing σ_b^2 (t).

3.4.2.2.2.2. Niblack’s Method

The Niblack method's main goal is to determine the threshold value using the local standard deviation and mean. Each pixel's threshold was calculated using the following expression [93].

$$T(x, y) = m(x, y) + k\delta(x, y)$$

Where, standard deviations $\delta(x, y)$ and local mean m (x, y) where determined by window size and standard k value in [0;1] range.

3.4.2.2.3. Iterative Global Thresholding

The approach of iterative global thresholding is simple and efficient. It uses an iterative technique to choose a global threshold for a document picture. The following steps are done in each ‘i’ Calculation of the average pixel value (Threshold T_i)

$$Ti = \left(\sum x \sum yIi(x, y)\right)/(m * n)$$

Where I (x, y) is the image after (i-1) th reputation

1. Subtraction of T_i from each pixel.
2. The gray scale histogram is stretched so that the remaining pixels to be distributed in all the grey scale tones.
3. Repetition of steps 1-3 till the termination condition is fulfilled.
4. Binarization of the final image.

In this study, the researchers used Otsu's thresholding methods to binarize the manuscript document because it provides the best result and other researcher recommendations for the requirement of producing a highly qualified image for further processing.

3.4.2.3. Image Segmentation

Image segmentation is one of the computer visions tasks which is used for partitioning digital images into different segments or parts according to their features and properties [41]. We can partition images into various parts that have similar attributes, and the parts that you have segmented are called "region of interest" or "image object". As a result, when the scripts are segmented, lines, words, and characters are isolated. Usually, recognition performance is proportional to segmentation techniques [20].

In this research, the researchers followed three steps in this study: line segmentation, word segmentation and character segmentation. The researchers used a stage-by-stage segmentation algorithm via projection profiles to isolate lines, word and characters in the scripts. The first step in the segmentation process is to divide the image of the script into lines. Horizontal projection can be used to segment the scripts' lines. The horizontal projection scheme computes the sum of all pixels in each row. If the sum is zero, begin a new line and terminate the current one.

The next step is word segmentation, in which each line of text is separated by a distance. It includes vertical scanning of the image, pixel-row by pixel-row from left to right and top to bottom. At The intensity of each pixel is evaluated.

The last step is to segment the characters within each line using the bounding box projection. Character segmentation using bounding box projection is more selective and has advantages over pixel projection. The bounding box projection method works on the premise that each character is treated as a connected object.

3.4.2.4. Size normalization

Size normalization or scaling is responsible for ensuring that all characters encountered in the scanned image are the same width and height. The purpose of size normalization is to reduce the size of the input image to a manageable size for the recognizer [20]. Because of handwriting variability, the characters in the manuscripts vary in size and shape. As a result, the researcher determined a baseline size for all images before feeding them into a convolutional neural network classifier. The image datasets were resized to 28 by 28 to be consistent and rearranged with a fixed resolution as part of the image preprocessing to make the model training computationally feasible and reduce the time of training or to reduce the computational burden in later processing.

3.4.3. Feature extraction

Feature extraction is a part of the dimensionality reduction process, in which an initial set of the raw data is divided and reduced to more manageable groups. So, when you want to process, it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So, feature extraction helps to get the best features from those big data sets by selecting and combining variables into features, thus effectively reducing the amount of data. These features are easy to process but still able to describe the actual data set with accuracy and originality [94].

In this study, feature extraction extracts relevant information that explains characters that an image can distinguish from others. And the accuracy of the developed model is highly dependent on this process. And in this study, this task is performed during recognition modeling or during CNN deep learning algorithm implementation. Then this algorithm has the ability to extract features from the given input data. The CNN model has two components: one for feature extraction and one for the classification part. The convolution and pulling layers perform feature extraction operations and the fully connected layer for classification. If your task is multiclass or multilabel, the output will be SoftMax. The main advantage of CNN is that it automatically detects the important features without any human supervision.

3.5. Proposed Deep Learning Models

The researcher uses two different deep learning algorithms for the recognition of manuscripts: three deep learning models were proposed, the first model was developed with CNN algorithm and the second two model were developed with hybrid of CNN algorithm and BiLSTM algorithm.

3.5.1. Proposed CNN model

The first proposed network has 7 layers, the first layer is convolved the first features via convolutional layers, after max pooling, and activation were applied. Then, the same operations were applied to the second convolutional layers. The third convolutional layer has 128 feature detectors (i.e., filters), with kernel size of 3*3 followed by a flatten layer and it converts the 2D matrix data to a vector called Flatten. Thus, it allows the output to be processed by standard fully connected NN. Then, the flatten layer has followed by two consecutive fully connected layer with 64 and 128 neurons respectively. The output of this layer is given to output layer and the output layer has 251 neurons for the 251 classes and Softmax function which produces a probability distribution of the 251 output classes.

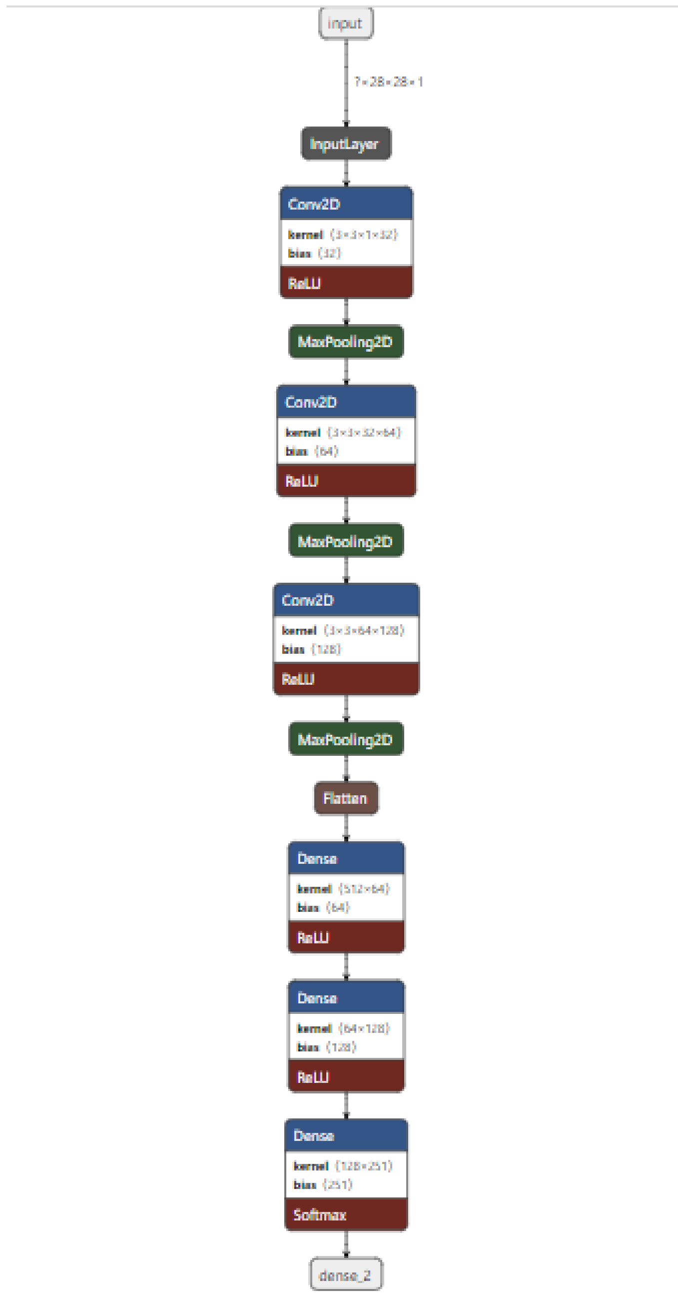


Figure 3.5. Proposed CNN layers for recognition model

3.5.2. Proposed CNN-BiLSTM (I)

The second proposed network has 3 convolutional layers with filters of 32,64,128 respectively, then followed by batch normalizer and next to batch normalizer layer two fully connected layers were followed with 64 and 128 filters then the output of filters was reshaped and its output is passed to the next dense layers. Next to this dense layer two BiLSTM layers were followed one for forward and the other is backward one. And these LSTM layers were followed by flatten layer and the output of this layer passes with dropout layer and connected to last output layer the output layer has 251 neurons for the 251 classes and Softmax function which produces a probability distribution of the 251 output classes.

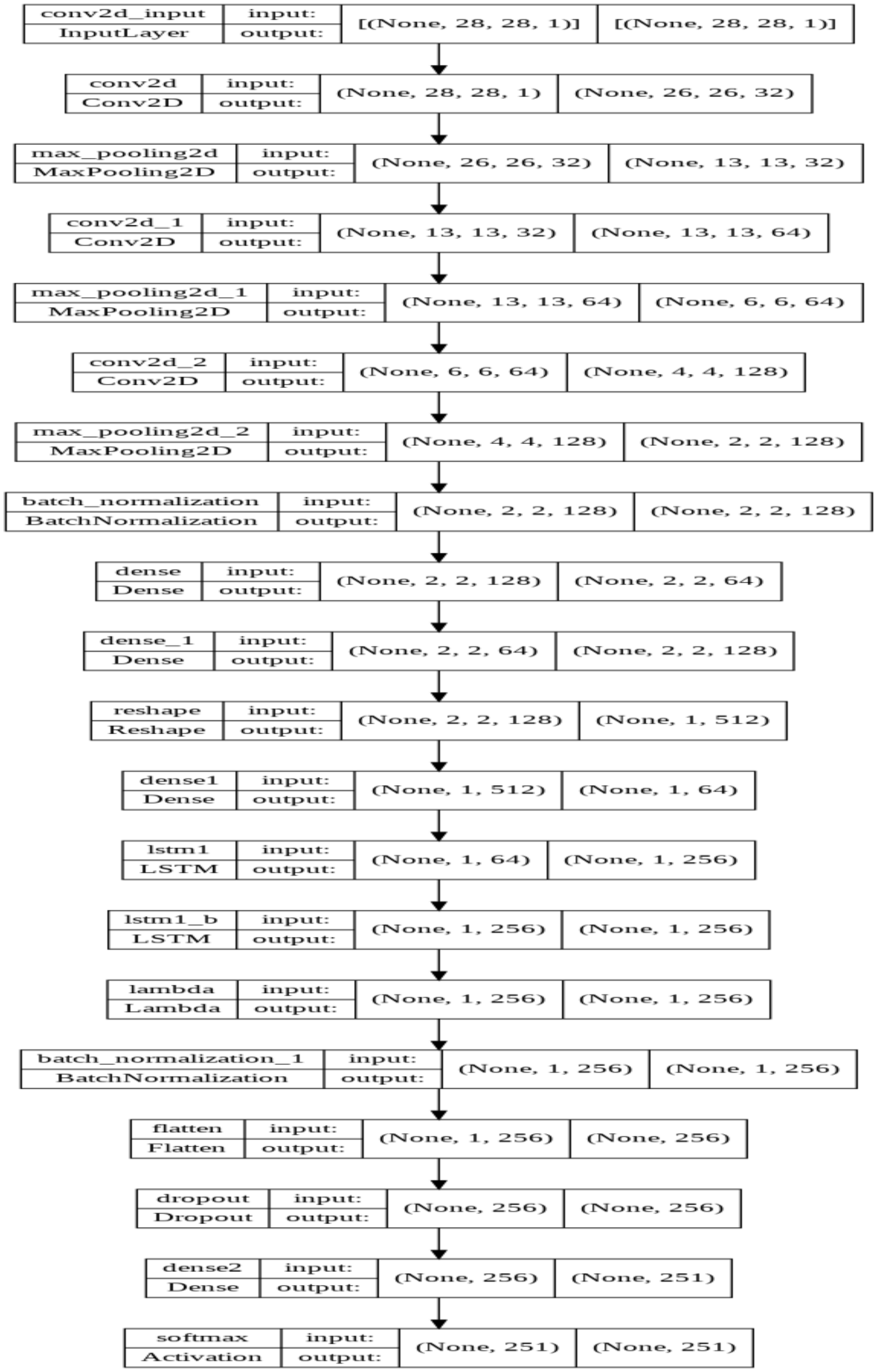
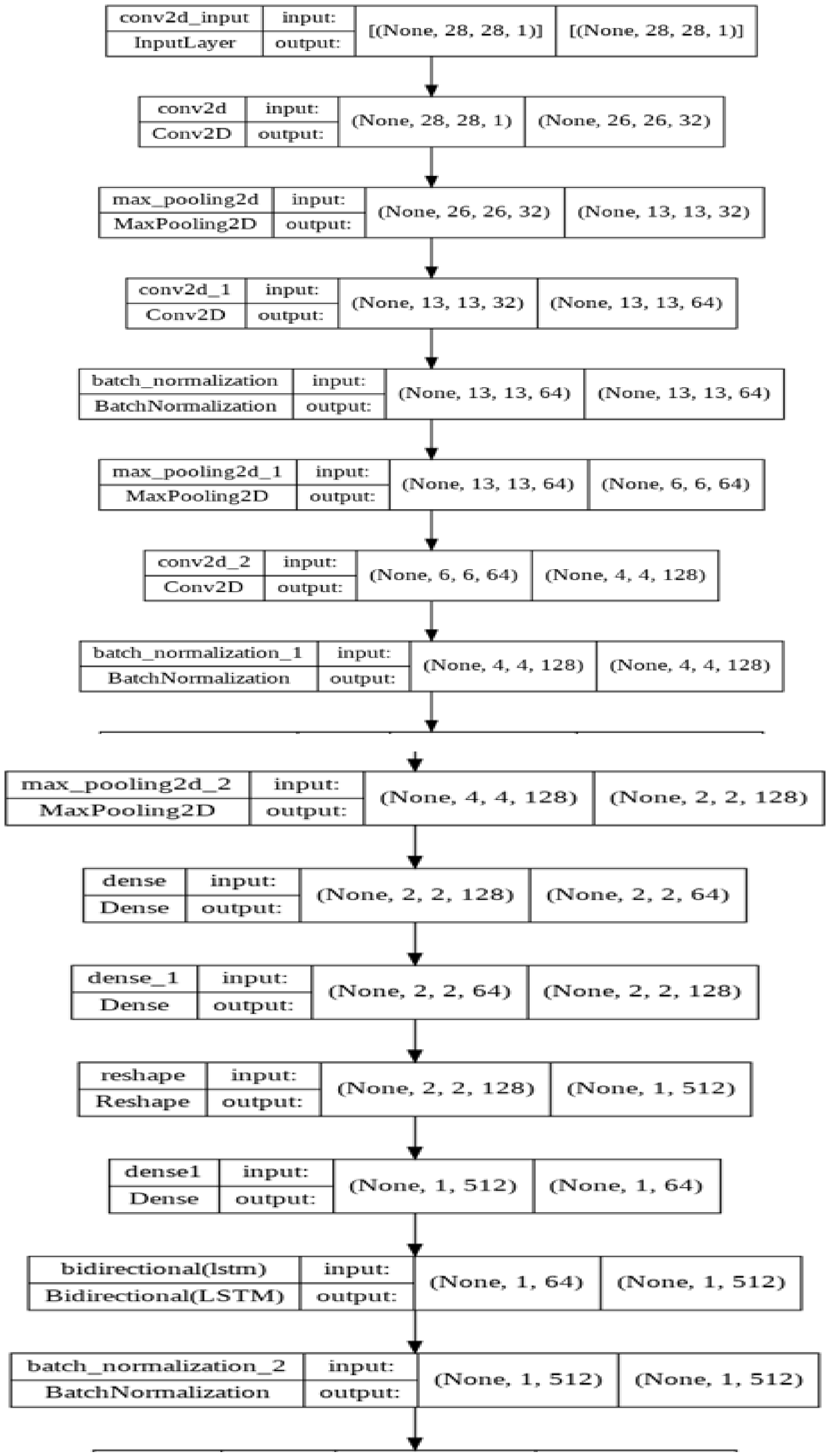


Figure 3. 6.Proposed CNN-BiLSTM (I) layers for recognition model

3.5.3. Proposed CNN-BiLSTM (II)

The last proposed model was similar construction of CNN-BiLSTM (I) except the difference of Bidirectional long-short term memory layer implementation.



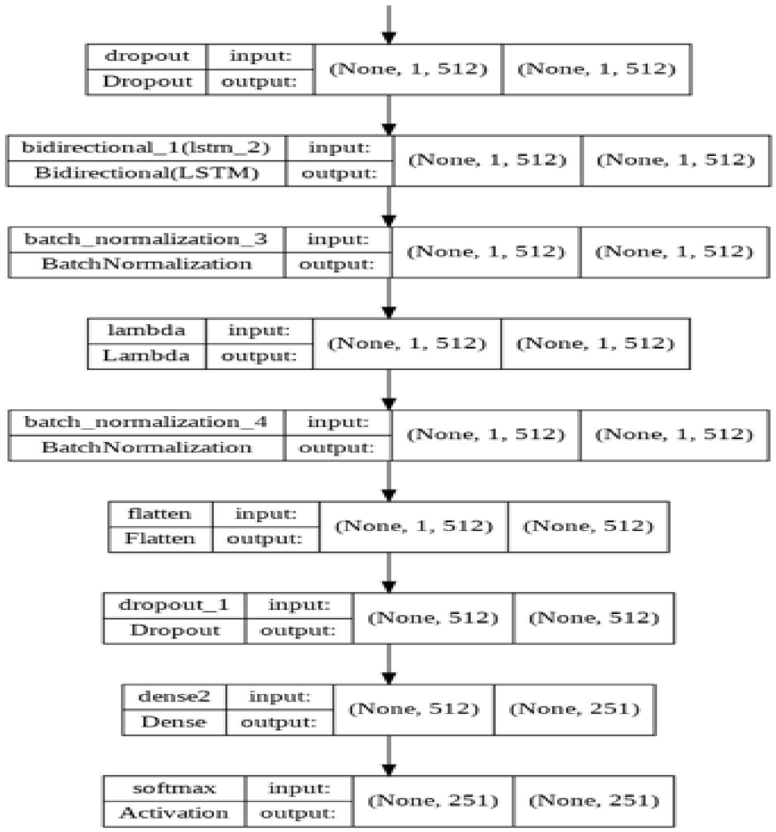


Figure 3.7.Proposed CNN-BiLSTM (II) layers for recognition model

3.6. Model Evaluation

From different experimental models, the best model will be selected based on the following evaluation or selection criteria:

Training complexity*:* by the model during training and testing is referred to as execution time.

Recognition accuracy: this metric summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions. In other words, it is the ratio of correct predictions to total sample data input. This selection parameter is one of the core evaluation criteria while we are selecting our model.

Training loss: it indicates that the number of errors is poorly recognized and it shows how the model is well fitted to the training data. This is also a major evaluation metric.

In addition to this in this study, researcher used manual evaluation system (rating scale), after the outperformer model deployed to system via end users.

CHAPTER FOUR

EXPERIMENTATIONS AND ANALYSIS

4.1 Overview

The researchers discussed the experiments they conducted to develop an OCR system for Amharic and Ge'ez manuscripts written in this chapter. The implementation of the previous chapter's preprocessing, segmentation, and recognition techniques will be presented. Thus, appropriate techniques are discussed with appropriate experiments and justification for each OCR process.

4.2. Experimental Result of Data Preprocessing

4.2.1 Image Binarization

This is the most initial image processing step; here the original image is converted to a grayscale image, and then the grayscale is again converted to a binary image. First, we should have to install the required Python library for image processing and analysis purposes. Then, after installing the required packages, the researcher first task is to convert the RGB image to a grayscale one as follows:

- ✓ First read RGB image (original image)
- ✓ Then convert RGB to grayscale image
- ✓ Finally, save as grayscale image

Here is an implementation of the above.

```
import cv2 #here is it importing python package
img = cv2.imread('DocImage2/C/C13.PNG') #read RGB image
grayscale_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #convert rgb to grayscale image
cv2.imwrite('DocImage2/grayscaled/grayscaled.jpg',grayscale_image)
cv2.imshow("orginal image",img)
cv2.imshow('grayscale image',grayscale_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Figure 4. 1. Code Snippet of RGB to Grayscale image Conversion

The original image is converted to a grayscale image and then this grayscale image is required for the next image processing steps. But before we go to the next stage of this image processing, let’s see the above code result.

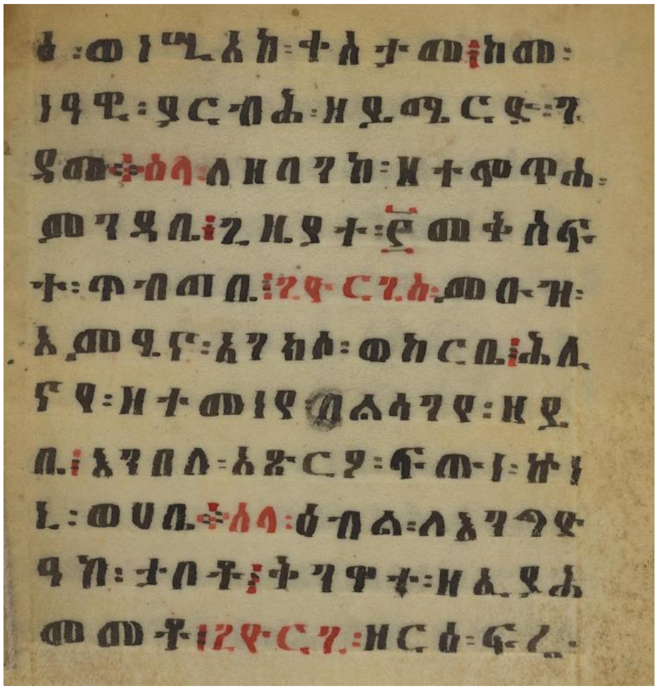


Figure 4. 2. manuscript image before converted to grayscale image

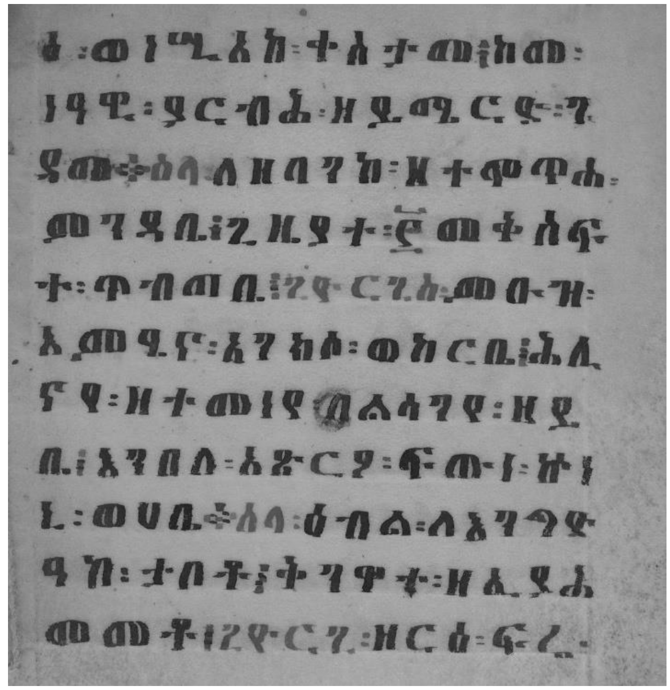


Figure 4. 3. manuscript image after converted to grayscale image

As we observe from the above image, the original image is converted to the original image by using the cvtColor () function.

The next step here that the researcher conducts is image binarization. As we have discussed in the previous chapter, Binarization is the way of converting or representing an image with 0's and 1's where 0 represents black and 1 represents white.

So here we are going to implement different image binarization algorithms, and then we are going to discuss them.

- ✓ First read grayscale image
- ✓ Define function for image binarization functions
- ✓ Feed grayscale image for each image binarization algorithms with different parameter
- ✓ Display to see what will happened via binarization algorithms.
- ✓ Save binarized image.

During image binarization, thresholding techniques play a big role, because different binarization algorithms perform their task via comparing each pixel of the image with the threshold value. If the threshold value is smaller than the threshold algorithms, then it sets the maximum value (255). So, during thresholding, we are performing the operation of separating an object from its background.

And thresholding has different parameters, such as

- ✓ Source: sources of input image array (must be grayscale)
- ✓ Thresholdvalue: the value of thresholding operations conducts
- ✓ maxValue: maximum value
- ✓ thresholding technique: the types of thresholding to be applied

Let’s implement different image thresholding algorithms.

```
import cv2 as cv
import numpy as np
def OTSU Thresholding(image):
    grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    retval2, threshold2 = cv2.threshold(grayscale_image,125,255, cv2.THRESH_BINARY_INV
+cv2.THRESH_OTSU)
    cv.imwrite ('DocImage2/threshold/OTSUThresholding28.jpg', threshold2)
img = cv2.imread('DocImage2/C/C13.PNG') #image reading
OTSU Thresholding(img)
cv.waitKey (60)
cv.destroyAllWindows ()
```

As we see in the above python code, the researcher implements OTSU Thresholding image binarization algorithms.

The binarization technique that the researcher used to conduct his binarization task is OTSU Thresholding. This thresholding algorithm is implemented with the def OTSU Thresholding (image) python function. Under OTSU, the threshold value isn’t chosen but it is automatically determined that a bimodal image (two distinct image values) is considered. The histogram generated contains two peaks. So, a generic condition would be to choose a threshold value that lies in the middle of both the histogram peak values. And its result looks like as follow via figure 4.4.

Figure 4. 4. manuscript image after OTSU Thresholding applied

4.2.2. Noise removal

When we are discussing image processing or studying images, we should have to consider image noise because it comes for different reasons during image acquisition, coding, transformation, and image processing stages. Handwritten documents are very noisy and degraded documents, so some kinds of noise removal techniques should be applied to obtain a better recognition result. Noise removal is the process of removing noise from an image by preserving the details of the image. This is performed by the filtering process.

Noise removal techniques are used to improve image quality by removing unnecessary information from images. Furthermore, noise removal techniques are used to restore historical noisy images. This study employs bi-level noise filtering techniques, as discussed by the researchers in Chapter three.

A. bi-level noise

The researcher applied the two-level noise-filtering techniques in order to get a better-quality image. In this technique, first the researchers de-noised the image using adaptive filtering and then applied the mean noise removing technique over the de-nosed image to improve the quality of the image.

```
img = cv2.resize(image-source, None, fx=0.8, fy=0.8, interpolation=cv2.INTER_CUBIC)
result = cv2.bilateralFilter(img, 0, 100, 5)
```

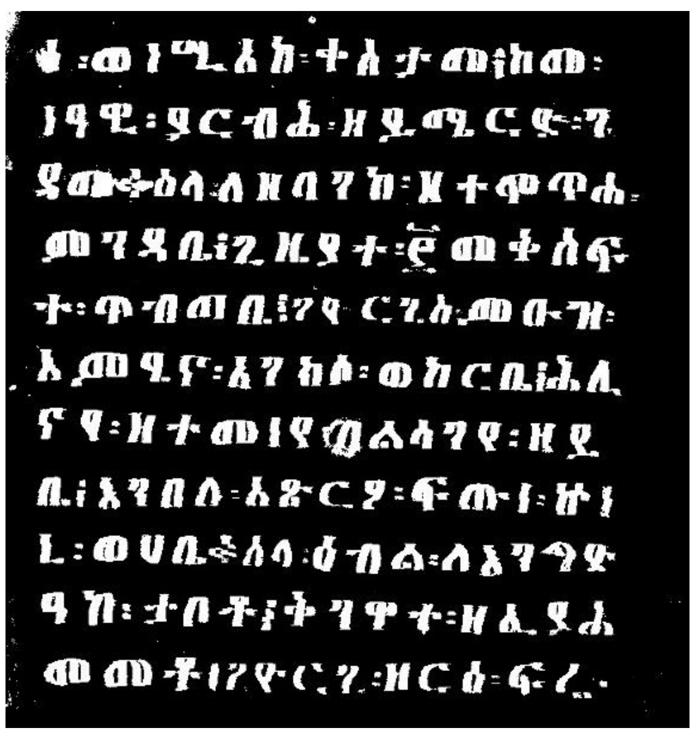


Figure 4. 5. Result of two-level noise-filter

4.2.3. Image Segmentation

In image processing, the most important steps are image segmentation and future extraction, and both of them are also the most challenging tasks. When we see character segmentation from a handwriting recognition point of view, it is the difficult task that the researcher has faced to extract characters from a manuscript document. Because difficulties come from the manuscript document, those documents have different writing styles of writers, font size, similarity in structure of characters, noise and simpler factors that increase segmentation difficulties. This stage has the ability to decide the recognition accuracy.

Under optical character recognition, there are three image segmentation techniques that are applied to segment portions of document images. Those are line segmentation. This is used to extract individual document line portions depending on line segmentation parameters implemented. The second one is word level document segmentation. This technique is implemented to extract portions of words from document image. And the third one is character level segmentation. In this technique, individual character objects are extracted from document images.

In this study, the researcher implements these three techniques from the above types of segmentation techniques. Those that are implemented by the researcher are line, word and character level image segmentation by using Vertical and Horizontal Histogram Projection.

4.2.3.1. Line Segmentation

It is the process of identifying lines in a given image. It includes horizontal scanning of the image, pixel -row by pixel-row from left to right and top to bottom. Taking its top and bottom pixel rows that are identified during transformation of pixels from background to foreground and vice versa during scanning identifies a line [12].

The researcher follows the following procedure to achieve this goal.

- ✓ Accept the list of images.
- ✓ Convert RGB image to Grayscale image
- ✓ Convert Grayscale image to Binary image
- ✓ Perform image dilation: in this process kernel is constructed with (1,200) because why researcher select this kernel size when kernel size increase it can achieve the correctness of line segmentation, and when we decrease kernel size you are going to decrease the size of your region of interest mean that it leads you character segment. And researcher also use this operation to achieve character level segmentation also.
- ✓ Find all contours/line on the image and return a list of coordinates for each contour.

- ✓ Sort image contours and Getting your region of interest (ROI)
- ✓ Crop the line from the images using the returned coordinates.
- ✓ Loop this image for all image

```
In [29]: image = cv2.imread('DocImage2/B/B44.PNG')
         #grayscale
         gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
         #binary
         ret,thresh = cv2.threshold(gray,125,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
         #dilation
         kernel = np.ones((1,200), np.uint8)
         img_dilation = cv2.dilate(thresh, kernel, iterations=1)
         cv2.imshow("image dialation",img_dilation)
         #find contours
         ctrs, hier = cv2.findContours(img_dilation.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
         #sort contours
         sorted_ctrs = sorted(ctrs, key=lambda ctr: cv2.boundingRect(ctr)[0])
         for i, ctr in enumerate(sorted_ctrs):
             # Get bounding box
             x, y, w, h = cv2.boundingRect(ctr)
             # Getting ROI
             roi = thresh[y:y+h, x:x+w]
             #show ROI
             #cv2.imshow('segment no:'+str(i),roi)
             cv2.imwrite("line segmentation/Final2/segment_no304_"+str(i)+".png",roi)
             cv2.rectangle(thresh,(x,y),( x + w, y + h ),(90,0,255),2)
             cv2.waitKey(0)
         cv2.imwrite('final_bounded_box_image.png',thresh)
         cv2.imwrite('dilation.jpg',img_dilation)
         cv2.imshow('marked areas',thresh)
         cv2.waitKey(0)
         cv2.destroyAllWindows()
```

Figure 4. 6. Code Snippet of line segmentation

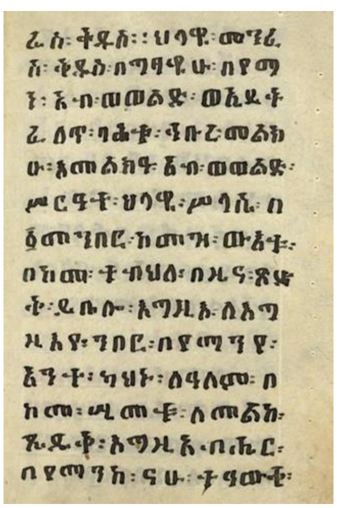


Figure 4. 7. Original image

Figure 4. 8. Dilated image

Figure 4. 9. Bounded line image (selected ROI)

Final segmented line image become

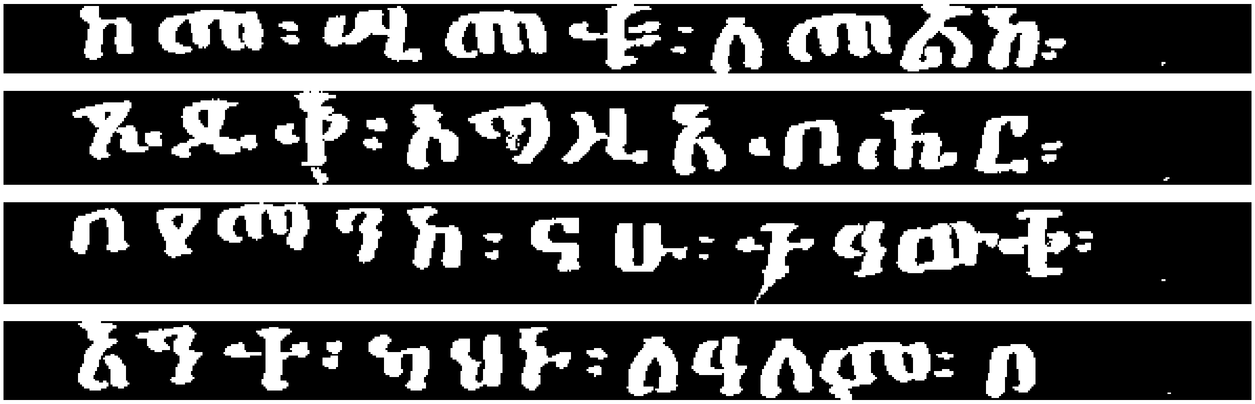


Figure 4. 10. Line Segmentation Results

4.2.3.2. Word Segmentation

At this level of segmentation, we are given an image with a single line (segmented in the previous step) containing a string of words. The goal of Word Level Segmentation is to divide an image into words. The concept is similar to the previous step, but the only difference is that we must now project the image vertically (Sum is calculated along columns) because we must segment words vertically.

If we project the binary image vertically,

- ✓ The text columns have a high number of foreground pixels, which corresponds to higher peaks in the histogram.
- ✓ Columns representing word gaps have a high number of background pixels, which correspond to lower peaks in the histogram.

Figure 4. 11. Word Segmentation Result

4.2.3.3. Character segmentation

At this level of segmentation, we are given an image with a single word (segmented in the previous step) made up of a sequence of characters. Character Level Segmentation's goal is to separate the image into individual characters. In order to segment or crop the character, we have used contour analysis. The steps followed for character segmentation are described below.

- ✓ Accept the list of word which are returned from the word segmentation stage.
- ✓ Find all contours/characters on the word and return a list of coordinates for each contour.
- ✓ Crop the characters from the threshold text word images using the returned coordinates.
- ✓ Iterate through each word

Its result becomes

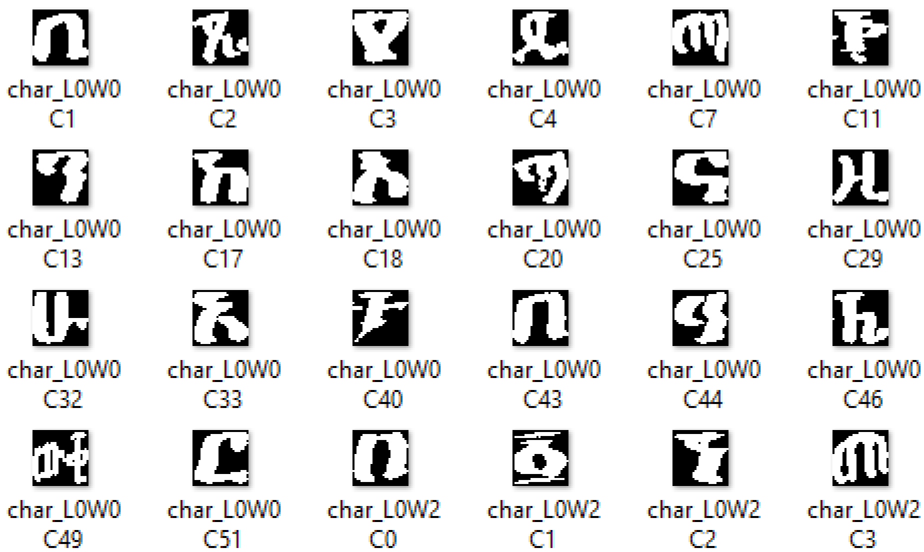


Figure 4. 12. Character Segmentation Results

During image segmentation, researchers used 705 manuscript document images from 11 different manuscript image document sources. Each manuscript has its own writing style, font, as well as quality of document that varies from document to document. So, each image processing stage is highly challenging, but all challenges increase their load in image segmentation. During character level segmentation, special characters (punctuation marks) and some characters of a manuscript document don't segment properly due to overlapped and detached characters or disconnected strokes and missed formation of characters' shape, style, and size owing to handwritten badly behaved characters like lack of standard font size and type. These factors reduce the accuracy of character segmentation and recognition systems.

4.2.4. Segmented Character image size Normalization

After the image is segmented into characters, it should be normalized because the recognition algorithm that we implemented accepts similar sized inputs. So, researchers implemented the following image resizing algorithm.

```
import cv2 as cv
import glob
import os
orginal_image = glob.glob('character segmentation/Final22/*.png')
i = 0
for f1 in orginal_image:
    img = cv.imread(f1)
    image_resized = cv.resize(img,(28,28))
    cv.imwrite('Resized_image/image%02i.jpg' %i,image_resized)
    i += 1
    cv.waitKey(30)
cv.destroyAllWindows()
```

Figure 4. 13. Code snippets for character image size normalization

As we see, the above python code input image (segmented character image) is resized to a (28,28) pixel image. Understand that your character's image is being normalized.

4.2.5. Dataset preparation

As we have discussed in the data acquisitions section, the researcher gathered 705 manuscript image documents, both from ancient Amharic and Ge’ez manuscript documents, from 11 different manuscript documents. And image preprocessing tasks are implemented by the researcher through the entire collected manuscript document image. So, while we performed character level image segmentation, lots of character images have been generated from those researchers selected 39012-character images from segmented characters. Then those selected images should be labeled into different classes. In this study, we have created 251 labeled classes. On each class label, character images are evenly distributed to prevent classification bias and improve recognition accuracy.

```python
import os
classes={}
path ="../Dataset/"
file = os.listdir(path)[:251]
print(file)
#in the following am tries to put classes in the dictionary as key value pair
classes ={'ha':0,'hu':1,'hi':2,'ha2':3,'ha2':4,'hh':5,'ho':6,'le':7,'lu':8,'li':9,'la':10,
          'lie':11,'l':12,'lo':13,'He':14,'Hu2':15,'Hi2':16,'Ha22':17,'Hie':18,'H2':19,'Ho2':20,
          'me':21,'mu':22,'mi':23,'ma':24,'mie':25,'m':26,'mo':27,'s2e':28,'s2u':29,
          's2i':30,'s2a':31,'s2ie':32,'s2':33,'s2o':34,'re':35,'ru':36,'ri':37,'ra':38,
          'rie':39,'r':40,'ro':41,'se':42,'su':43,'si':44,'sa':45,'sie':46,'s':47,'so':48,
          'Se2':49,'Su2':50,'Si2':51,'Sa2':52,'Sie2':53,'S22':54,'So2':55,'qe':56,'qu':57,'qi':58,
          'qa':59,'qie':60,'q':61,'qo':62,'be':63,'bu':64,'bi':65,'ba':66,'bie':67,'b':68,'bo':69,
          've':70,'vu':71,'vi':72,'va':73,'vie':74,'v':75,'vo':76,'te':77,'tu':78,'ti':79,'ta':80,
          'tie':81,'t':82,'to':83,'ce':84,'cu':85,'ci':86,'ca':87,'cie':88,'c':89,'co':90,
          'h2e':91,'h2u':92,'h2i':93,'h2a':94,'h2ie':95,'h22':96,'h2o':97,'ne':98,'nu':99,'ni':100,
          'na':101,'nie':102,'n':103,'no':104,'Ne2':105,'Nu2':106,'Ni2':107,'Na2':108,'Nie2':109,'N2':110,'No2':111,
          'e':112,'u':113,'i':114,'a':115,'ie':116,'A2':117,'o':118,'ke':119,'ku':120,'ki':121,'ka':122,'kie':123,
          'k':124,'ko':125,'Ke2':126,'Ku2':127,'Ki2':128,'Ka2':129,'Kie2':130,'K2':131,'Ko2':132,
          'we':133,'wu':134,'wi':135,'wa':136,'wie':137,'w':138,'wo':139,
          'Oe':140,'Ou':141,'Oi':142,'Oa':143,'Oie':144,'O':145,'Oo':146,
          'ze':147,'zu':148,'zi':149,'za':150,'zie':151,'z':152,'zo':153,
          'Ze2':154,'Zu2':155,'Zi2':156,'Za2':157,'Zie2':158,'Z2':159,'Zo2':160,
          'ye':161,'yu':162,'yi':163,'ya':164,'yie':165,'y':166,'yo':167,
          'de':168,'du':169,'di':170,'da':171,'die':172,'d':173,'do':174,
          'je':175,'ju':176,'ji':177,'ja':178,'jie':179,'j':180,'jo':181,
          'ge':182,'gu':183,'gi':184,'ga':185,'gie':186,'g':187,'go':188,
          'Te2':189,'Tu2':190,'Ti2':191,'Ta2':192,'Tie2':193,'T2':194,'To2':195,
          'Ce2':196,'Cu2':197,'Ci2':198,'Ca2':199,'Cie2':200,'C2':201,
          'Pe':202,'Pu':203,'Pi':204,'Pa':205,'Pie':206,'P':207,'Po':208,
          'xe':209,'xu':210,'xi':211,'xa':212,'xie':213,'x':214,'xo':215,
          'x2e':216,'x2u':217,'x2i':218,'x2a':219,'x2ie':220,'x2':221,'x2o':222,
          'fe':223,'fu':224,'fi':225,'fa':226,'fie':227,'f':228,'fo':229,
          'pe2':230,'pu2':231,'pi2':232,'pa2':233,'pie2':234,'p2':235,'po2':236,
          'one':237,'two':238,'three':239,'four':240,'five':241,'six':242,'seven':243,'eight':244,'nine':245,
         'ten':246,'haya':247,'selasa':248,'ariba':249,'hy':250}
```

Figure 4. 14. Code Snippet for Class Labeling

As we observe from the above python code, the researcher first reads the path of the dataset and initializes variable 'file' and displays a list of labels for the dataset, and then creates a python data dictionary as key value pair for each character class as classes.

Then let's create two python arrays X and Y, one for image and the other for class/labels of datasets as follows.

```python
X = []
Y = []
for cls in classes:
    pth = path + cls
    for img_name in os.listdir(pth):
        path_image = pth + "/" + img_name
        img = cv2.imread(path_image,0)
        X.append(img)
        Y.append(classes[cls])
print("dataset created Successfully!")
```

```
dataset created Successfully!
```

Figure 4. 15. Code snippet for dataset preparation

After we have declared X and Y arrays, we iterate on the class data dictionary, then append the image to X and class labels to Y. Now, we have created our own image datasets.

4.2.5.1. Split Datasets for Testing and Training data

Because using similar data for testing and training leads to biases in model performance, the researcher divided the dataset into training and testing sets. So, creating training and testing for the model and testing for evaluating whether the model can generalize well to new, unseen data, respectively,

Researchers classify modeling datasets by splitting the data into 80% for training and 20% for testing. The test set is a separate set of data used to test the model after completing the training. And the testing set was further split to 10% for testing and 10% for validation set. In this study, an experiment was also conducted with a data split ratio of 70% for training and 30% for testing data. The validation set is a set of data, separate from the training set, that is used to validate our model's performance during training. This validation process gives us information that helps us tune the model's hyperparameters and configurations accordingly. It is like a critic telling us whether the training is moving in the right direction or not. The model is trained on the training set, and, simultaneously, the model evaluation is performed on the validation set after every epoch. The main idea of splitting the dataset into a validation set is to prevent our model from overfitting, i.e., the model becoming really good at classifying the samples in the training set but not being able to generalize and make accurate classifications on the data it has not seen before [95]. The random state hyperparameter controls the shuffling process and is used for initializing the internal random number generator, which will decide the splitting of data into train and test indices in your case. With random state = None, we get different training and testing sets across different executions, and the shuffling process is out of control. In this study, researchers took a random state of 10. This task has been implemented as follows:

```
xtrain,xtest,ytrain,ytest = train_test_split(X,Y,test_size=.20, random_state=10)

X_train = np.asarray(xtrain)
Y_train = np.asarray(ytrain)
X_test = np.asarray(xtest)
Y_test = np.asarray(ytest)
print(X_train.shape, Y_train.shape)
print(X_test.shape, Y_test.shape)
```

```
(31209, 28, 28) (31209,)
(7803, 28, 28) (7803,)
```

Figure 4. 16. Code Snippet to Splitting dataset as training and test set of 80:20

As we have seen in the above code synopses, 80 % of the data set can be categorized as training and the remaining 20 % can be testing data.

And the second experiment was performed with the following split of data: 70% for training data and the reset of 30% of the dataset for testing data.

```
xtrain,xtest,ytrain,ytest = train_test_split(X,Y,test_size=.30, random_state=10)

X_train = np.asarray(xtrain)
Y_train = np.asarray(ytrain)
X_test = np.asarray(xtest)
Y_test = np.asarray(ytest)
print(X_train.shape, Y_train.shape)
print(X_test.shape, Y_test.shape)
```

```
(27308, 28, 28) (27308,)
(11704, 28, 28) (11704,)
```

Figure 4. 17. Code Snippet to Splitting dataset as training and test set of 70:30

4.2.6. Normalizing the data

Normalize the value of each image pixel as images are in gray level (1 channel ==> 0 to 255), not color (RGB).

```
X_train = tf.keras.utils.normalize(X_train, axis = 1)
X_test = tf.keras.utils.normalize(X_test, axis = 1)
plt.imshow(X_train[0], cmap = plt.cm.binary)
```

Figure 4. 18. Code snippet. Normalizing data

Resizing images to make them suitable for Apply Convolution operation. For multilayer perceptron model we must reduce the image in to vector pixels. So, in this research case researcher reduce the input image in to 28*28-pixel input images and its operations performed as follow.

```
IMG_SIZE=28
#increasing one dimension for Kernel operation for testing and training image
X_train = np.array(X_train).reshape(-1,IMG_SIZE,IMG_SIZE,1)
X_test = np.array(X_test).reshape(-1,IMG_SIZE,IMG_SIZE,1)
```

Figure 4. 19. Code Snippet for resizing input data for input layer

As we see in the above code block, reshape (-1, IMG_SIZE, IMG_SIZE,1) is equivalent to originalImage. reshape (orginalImage.shape, 1). This means if the image size is (28, 28), the final output image will become (28, 28, 1).

4.3. Building a Deep Neural Network models

Here, the researcher uses two different neural network algorithms to achieve objectives of the study. These are using the Convolutional neural network and the CNN_BiLSTM algorithm and evaluating their performance. From these models, the outperforming model was selected for model building of Amharic and Ge'ez manuscript character recognition.

First let's build CNN based model as follow;

1. ***Convolution layer***: a convolution layer, called Conv2D
    - ✓ It has 32 feature detectors (i.e., filters), with kernel size of 3*3
    - ✓ ReLU as activation function and
    - ✓ Input size of 28*28*1
2. ***Pooling layer***: it's a maxpooling layer, called MaxPool2D
    - ✓ it is configured with a pool size of 2*2 and stride value of 2, to determine how many steps we are moving in each step of convolution operation.
3. ***Convolution layer***: a convolution layer, called Conv2D
    - ✓ It has 64 feature detectors (i.e., filters), with kernel size of 3*3
    - ✓ ReLU as activation function and
    - ✓ Padding: The padding parameter of the Kera's Conv2D class can take one of two values: 'valid' or 'same'. Setting the value to "valid" parameter means that the input volume is not zero-padded and the spatial dimensions are allowed to reduce via the natural application of convolution. You can instead preserve spatial dimensions of the volume such that the output volume size matches the input volume size, by setting the value to the "same" [96]. In this case we are setting the padding value as "same".
4. ***Pooling layer***: it's a maxpooling layer, called MaxPool2D
    - ✓ it is configured with a pool size of 2*2 and stride value of 2.
5. ***Convolution layer***: a convolution layer, called Conv2D
    - ✓ It has 128 feature detectors (i.e., filters), with kernel size of 3*3
    - ✓ ReLU as activation function and
    - ✓ Padding: valid
6. ***Pooling layer***: it's a maxpooling layer, called MaxPool2D
    - ✓ it is configured with a pool size of 2*2 and stride value of 2.
7. ***Flattening layer***: it converts the 2D matrix data to a vector called Flatten.

- ✓ it allows the output to be processed by standard fully connected NN

6. ***Fully connected layer 1***:
   - ✓ it has 64 neurons
   - ✓ ReLU as activation function
7. **Fully connected layer 2**:
   - ✓ it has 128 neurons
   - ✓ ReLU as activation function
8. ***Output layer***:
   - ✓ it has 251 neurons for the 251 classes
   - ✓ Softmax as the activation function to output probability for each class.

```
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'valid'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Flatten())
model.add(Dense(64,activation ="relu"))
model.add(Dense(128,activation ="relu"))
model.add(Dense(251,activation ="softmax"))
```

Figure 4. 20. CNN model Building Code Snippet

Here the researcher implements 3 convolution layers with different hyper parameters, 3 pooling layers, 1 flattening layer, and 3 dense layers. we used ReLU activation during the convolutional layer and Softmax for the output layer. As we have discussed before, the input layer accepts an input image with a (28,28,1) size image. Then convolution begins at convolution and maximum pooling layer to extract features of input image data and pooling layer to decrease the size of the convolved feature map to reduce the computational costs. At the output layer, output is equivalent to the number of classes that will be recognized.

CNN model summary looks like as follow:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_3 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_4 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_4 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_5 (Conv2D)           (None, 4, 4, 128)         73856

 max_pooling2d_5 (MaxPooling  (None, 2, 2, 128)        0
 2D)

 flatten_1 (Flatten)         (None, 512)               0

 dense_3 (Dense)             (None, 64)                32832

 dense_4 (Dense)             (None, 128)               8320

 dense_5 (Dense)             (None, 251)               32379

=================================================================
Total params: 166,203
Trainable params: 166,203
Non-trainable params: 0
_________________________________________________________________
```

Figure 4. 21. CNN model Summary

The second model that researcher implement was the CNN_BiLSTM(I) algorithm for developing the second experimental model with similar steps of constructing CNN model except some additional layers. And CNN-BiLSTM(I) algorithms have been implemented as follows:

```
import keras
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'valid'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(BatchNormalization())
model.add(Dense(64,activation ="relu"))
model.add(Dense(128,activation ="relu"))
model.add(Reshape(target_shape=((1,512)), name='reshape'))
model.add(Dense(64, activation='relu', kernel_initializer='he_normal', name='dense1'))
model.add(LSTM(256, return_sequences=True, kernel_initializer='he_normal', name='lstm1'))
model.add(LSTM(256, return_sequences=True, go_backwards=True, kernel_initializer='he_normal', name='lstm1_b'))
model.add(Lambda(lambda inputTensor: K.reverse(inputTensor, axes=1)))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(251, kernel_initializer='he_normal',name='dense2'))
model.add(Activation('softmax', name='softmax'))
```

Figure 4. 22.Building CNN-BiLSTM(I) based model one

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 4, 4, 128)         73856

 max_pooling2d_2 (MaxPooling  (None, 2, 2, 128)        0
 2D)

 batch_normalization (BatchN  (None, 2, 2, 128)        512
 ormalization)

 dense (Dense)               (None, 2, 2, 64)          8256

 dense_1 (Dense)             (None, 2, 2, 128)         8320

 reshape (Reshape)           (None, 1, 512)            0

 dense1 (Dense)              (None, 1, 64)             32832

 lstm1 (LSTM)                (None, 1, 256)            328704

 lstm1_b (LSTM)              (None, 1, 256)            525312

 lambda (Lambda)             (None, 1, 256)            0

 batch_normalization_1 (Batc  (None, 1, 256)           1024
 hNormalization)

 flatten (Flatten)           (None, 256)               0

 dropout (Dropout)           (None, 256)               0

 dense2 (Dense)              (None, 251)               64507

 softmax (Activation)        (None, 251)               0

=================================================================
Total params: 1,062,139
Trainable params: 1,061,371
Non-trainable params: 768
_________________________________________________________________
```

Figure 4. 23.CNN-BiLSTM(I) model summary

As we have seen figure 4.23, the model generates 768 non-trainable parameters those are the number of weights that are not updated during training backpropagation and that have to be defined a priori, or passed as inputs.

The 3rd model that researchers implement is the CNN_BiLSTM (II) algorithm for developing the 3rd experimental model with similar steps of constructing CNN-BiLSTM(I) based model except some modification. And CNN-BiLSTM (II) algorithms have been implemented as follows:

```
import keras
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'valid'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Dense(64,activation ="relu"))
model.add(Dense(128,activation ="relu"))
model.add(Reshape(target_shape=((1,512)), name='reshape'))
model.add(Dense(64, activation='relu', name='dense1'))

forward_layer = LSTM(256, return_sequences=True)
backward_layer = LSTM(256, activation='relu', return_sequences=True,
                       go_backwards=True)
model.add(Bidirectional(forward_layer, backward_layer=backward_layer))
model.add(BatchNormalization())
model.add(Dropout(0.5))
forward_layer1 = LSTM(256, return_sequences=True)
backward_layer1 = LSTM(256, activation='relu', return_sequences=True,
                       go_backwards=True)
model.add(Bidirectional(forward_layer1, backward_layer=backward_layer1))
model.add(BatchNormalization())
model.add(Lambda(lambda inputTensor: K.reverse(inputTensor, axes=1)))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(251, kernel_initializer='he_normal',name='dense2'))
model.add(Activation('softmax', name='softmax'))
```

Figure 4. 24.Building CNN-BiLSTM (II) based model two

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496     
                                                                 
 batch_normalization (BatchN  (None, 13, 13, 64)       256       
 ormalization)                                                   
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0         
 2D)                                                             
                                                                 
 conv2d_2 (Conv2D)           (None, 4, 4, 128)         73856     
                                                                 
 batch_normalization_1 (Batc  (None, 4, 4, 128)        512       
 hNormalization)                                                 
                                                                 
 max_pooling2d_2 (MaxPooling  (None, 2, 2, 128)        0         
 2D)                                                             
                                                                 
 dense (Dense)               (None, 2, 2, 64)          8256      
                                                                 
 dense_1 (Dense)             (None, 2, 2, 128)         8320      
                                                                 
 reshape (Reshape)           (None, 1, 512)            0         
                                                                 
 dense1 (Dense)              (None, 1, 64)             32832     
                                                                 
 bidirectional (Bidirectiona  (None, 1, 512)           657408    
 l)                                                              
                                                                 
 batch_normalization_2 (Batc  (None, 1, 512)           2048      
 hNormalization)                                                 
                                                                 
 dropout (Dropout)            (None, 1, 512)             0         
                                                                 
 bidirectional_1 (Bidirectio  (None, 1, 512)            1574912   
 nal)                                                            
                                                                 
 batch_normalization_3 (Batc  (None, 1, 512)            2048      
 hNormalization)                                                 
                                                                 
 lambda (Lambda)              (None, 1, 512)             0         
                                                                 
 batch_normalization_4 (Batc  (None, 1, 512)            2048      
 hNormalization)                                                 
                                                                 
 flatten (Flatten)            (None, 512)                0         
                                                                 
 dropout_1 (Dropout)          (None, 512)                0         
                                                                 
 dense2 (Dense)               (None, 251)                128763    
                                                                 
 softmax (Activation)         (None, 251)                0         
                                                                 
=================================================================
Total params: 2,510,075
Trainable params: 2,506,619
Non-trainable params: 3,456
_________________________________________________________________
```

Figure 4. 25.CNN-BiLSTM (II) based model two Summary

4.4. Experimental Setup

4.4.1. Software and Hardware Tools Used for Experiment

During the experiment, we used the following hardware and software tools

#	Hardware	software	Purpose
1	HP laptop with Intel®core™ i7-1065G7 CPU @1.30GHZ, 1498 MHZ, 4 Core(s), 8 logical processor(s) and, 8GB RAM.	Windows 10 Pro, 64-bit operating system Jupyter Notebook (Anaconda3) with Python 3.10.4	For data preparation and preprocessing, for writing thesis report and experimentation.

Table 4. 1. Hardware and Software setting

The HP Laptop was used for data preparation and preprocessing, as well as writing the thesis report. The main laboratory environment for training the proposed model is the Jupyter notebook (anaconda3).

Furthermore, to implement the proposed model various neural network modules and libraries such as keas, OpenCV, TensorFlow, Numpy and matplotlib packages are used.

4.4.2. Experimental Setting

We conducted six experiments with two different learning rate, dataset percentage of split, number of epochs and algorithms. For this experiment, we used similar dataset.

Scenario No.	Model	Experiment	Learning Rate	Number of Epoch	Percentage of split
I	CNN	Experiment-1	0.001	10	70%:30%
		Experiment-2	0.0001	10	
		Experiment-3	0.001	15	
		Experiment-4	0.0001	15	
		Experiment-5	0.001	20	
		Experiment-6	0.0001	20	
		Experiment-7	0.001	25	
		Experiment-8	0.0001	25	
	CNN-BiLSTM(I)	Experiment-1	0.001	10	
		Experiment-2	0.0001	10	
		Experiment-3	0.001	15	
		Experiment-4	0.0001	15	
		Experiment-5	0.001	20	

		Experiment-6	0.0001	20	
		Experiment-7	0.001	25	
		Experiment-8	0.0001	25	
	CNN-BiLSTM (II)	Experiment-1	0.001	10	
		Experiment-2	0.0001	10	
		Experiment-3	0.001	15	
		Experiment-4	0.0001	15	
		Experiment-5	0.001	20	
		Experiment-6	0.0001	20	
		Experiment-7	0.001	25	
		Experiment-8	0.0001	25	
II	CNN	Experiment-1	0.001	10	80%:20%
		Experiment-2	0.0001	10	
		Experiment-3	0.001	15	
		Experiment-4	0.0001	15	
		Experiment-5	0.001	20	
		Experiment-6	0.0001	20	
		Experiment-7	0.001	25	
		Experiment-8	0.0001	25	
	CNN-BiLSTM(I)	Experiment-1	0.001	10	
		Experiment-2	0.0001	10	
		Experiment-3	0.001	15	
		Experiment-4	0.0001	15	
		Experiment-5	0.001	20	
		Experiment-6	0.0001	20	
		Experiment-7	0.001	25	
		Experiment-8	0.0001	25	
	CNN-BiLSTM (II)	Experiment-1	0.001	10	
		Experiment-2	0.0001	10	
		Experiment-3	0.001	15	

		Experiment-4	0.0001	15	
		Experiment-5	0.001	20	
		Experiment-6	0.0001	20	
		Experiment-7	0.001	25	
		Experiment-8	0.0001	25	

Table 4. 2 Experimental Setting

Scenario I: in this experiment, we used a total of 39,012-character image data. To conduct this experiment, researcher used CNN based model and two different CNN-BiLSTM(I) and CNN-BiLSTM (II) based model with different architecture by using the hybrids of CNN algorithm and BiLSTM algorithm. Moreover, the dataset was split into two different groups with 70% training and 30% of testing data. That means the dataset was split into 27,308 and 11,704 for training and testing set respectively. Besides, this experiment was conducted by 28 *28 input images with different parameters and hyperparameters from these parameters researcher train the developed model at epoch of 10,15,20 and 25, Adam optimizer with learning rate of 0.001 and 0.0001, loss function of sparse_categorical_crossentropy and default batch size. And as we observe on the above table for each developed model there were eight different experiment's conducted mean that under this scenario 24 different experiments ware conducted.

Scenario II: in this experiment, we used a total of 39,012-character image data. To conduct this experiment, researcher used CNN based model and two different CNN-BiLSTM(I) and CNN-BiLSTM (II) based model with different architecture by using the hybrids of CNN algorithm and BiLSTM algorithm. Moreover, the dataset was split into two different groups with 80% training and 20% of testing data. That means the dataset was split into 31,209 and 7803 for training and testing set respectively. Besides, this experiment was conducted by 28 *28 input images with different parameters and hyperparameters from these parameters researcher train the developed model at epoch of 10,15,20 and 25, Adam optimizer with learning rate of 0.001 and 0.0001, loss function of sparse_categorical_crossentropy and default batch size. And as we observe on the above table for each developed model there were eight different experiment's conducted mean that under this scenario 24 different experiments ware conducted.

Evaluation metrics: we used both training time, training accuracy and validation accuracy for these experiments.

4.5. Experimental Result

Experiment results under Scenario I: The researcher ran this experiment with a dataset split ratio of 70% for training and 30% for testing, as well as with different numbers of parameters and hyperparameters.

CNN model model with 70:30 training testing set ratio	Experiment	#epoch	time	loss	Accuracy	Val_accuracy	Learning_rate
	Exp-1	10	16.28s	0.3201	0.8858	0.7587	0.001
	Exp-2	10	17.28s	0.0310	0.9922	0.7962	0.0001
	Exp-3	15	31.52s	0.0308	0.9894	0.7981	0.001
	Exp-4	**15**	**30.50s**	**0.0033**	**0.9979**	**0.8209**	**0.0001**
	Exp-5	20	32.54s	0.0177	0.9939	0.8045	0.001
	Exp-6	20	31.52s	0.0031	0.9977	0.8164	0.0001
	Exp-7	25	17.28s	0.0256	0.9914	0.7886	0.001
	Exp-8	25	32.53s	0.0029	0.9978	0.8164	0.0001
CNN_BiLSTM(I) model with 70:30 training testing set ratio	Experiment	#epoch	Time	Loss	Accuracy	Val_accuracy	Learning_rate
	Exp-1	10	27.45s	0.3991	0.8708	0.7838	0.001
	Exp-2	10	34.58s	0.0429	0.9878	0.8401	0.0001
	Exp-3	15	45.75s	0.1818	0.9410	0.7954	0.001
	Exp-4	**15**	**45.75s**	**0.0138**	**0.9954**	**0.8441**	**0.0001**
	Exp-5	20	28.47s	0.1087	0.9670	0.7974	0.001
	Exp-6	20	25.41s	0.0068	0.9973	0.8390	0.0001
	Exp-7	25	44.74s	0.1017	0.9700	0.8075	0.001
	Exp-8	25	43.72s	0.0057	0.9978	0.8388	0.0001
CNN_BiLSTM (II) model with 70:30 training testing set ratio	Experiment	#epoch	Time	Loss	Accuracy	Val_accuracy	Learning_rate
	Exp-1	10	44.72s	0.6843	0.7893	0.7893	0.001
	Exp-2	10	45.74s	0.2277	0.9236	0.8626	0.0001
	Exp-3	15	43.70	0.3226	0.8942	0.8086	0.001
	Exp-4	15	44.71	0.0585	0.9811	0.8661	0.0001

	Exp-5	20	41.67	0.1829	0.9416	0.8422	0.001
	Exp-6	**20**	**70.114**	**0.0201**	**0.9941**	**0.8729**	**0.0001**
	Exp-7	25	74.121	0.1261	0.9609	0.8368	0.001
	Exp-8	25	69.112	0.0110	0.9968	0.8688	0.0001

Table 4. 3. Experiment results under scenario I

As we observe on table 4.2, the experiment is conducted via different numbers of epochs and learning rates and gets different results. Then select one of the outperformed conditions from the listed experimental results, but researchers choose a model whose result has been better at accuracy and validation accuracy. So, in the CNN based model, it outperforms on Experiment-4 at epoch 15 and learning rate 0.0001 with training accuracy of 0.9979 and 0.0033 training loss. Its validation accuracy becomes 0.8208 with a training time of 30.50 seconds. For the second CNN-BiLSTM(I) model on Experiment-4 at epoch 15 and a learning rate of 0.0001, the CNN_BiLSTM(I)-based recognition model's training and validation accuracy are 0.9954 and 0.8441, respectively, with a training loss of 0.0138 and a training time of 45.75 seconds. And the 3rd model CNN_BiLSTM (II) model, on Experment-6 with 20 epoch and a learning rate of 0.0001, the CNN_BiLSTM (II)-based recognition model's training and validation accuracy are 0.9941 and 0.8729, respectively, with a training loss of 0.0201 and a training time of 70.114 seconds.

Let's visualize the above experimental result under first experimental scenarios.

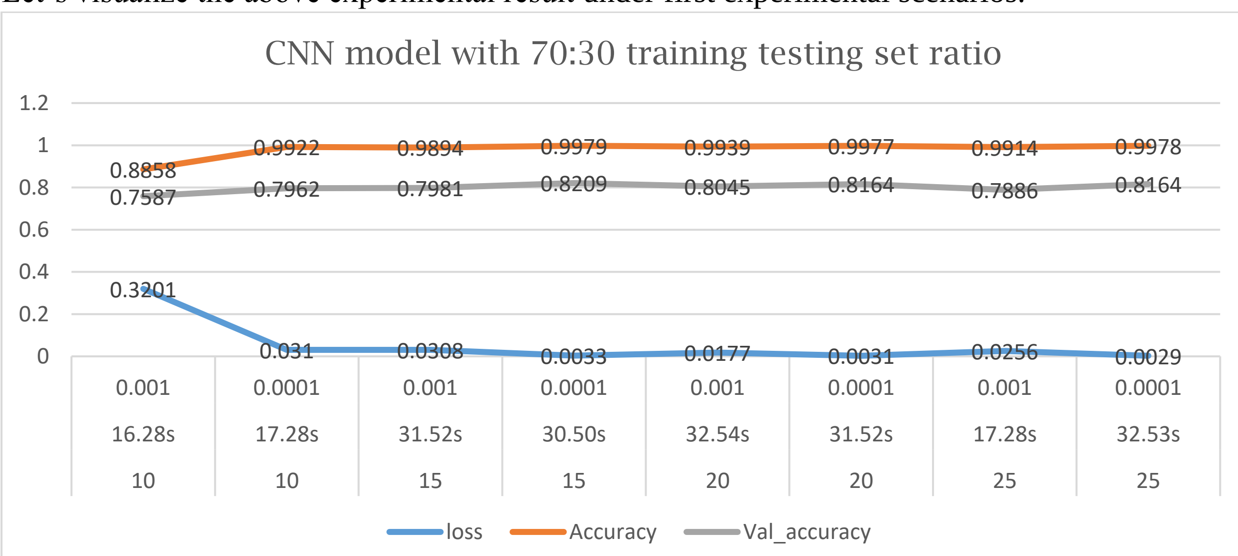


Figure 4. 26. CNN model with 70:30 training testing set ratio experimental result visualization

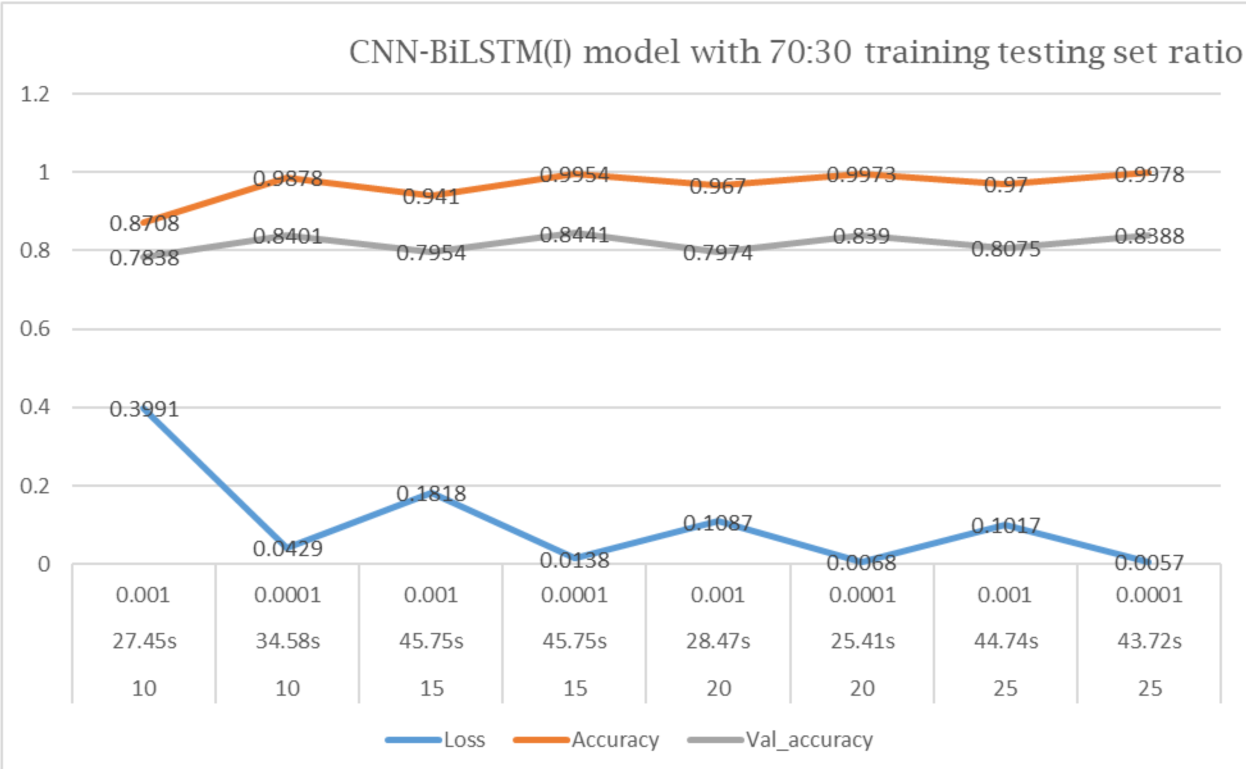


Figure 4. 27. CNN-BiLSTM(I) model with 70:30 training testing set ratio experimental result visualization

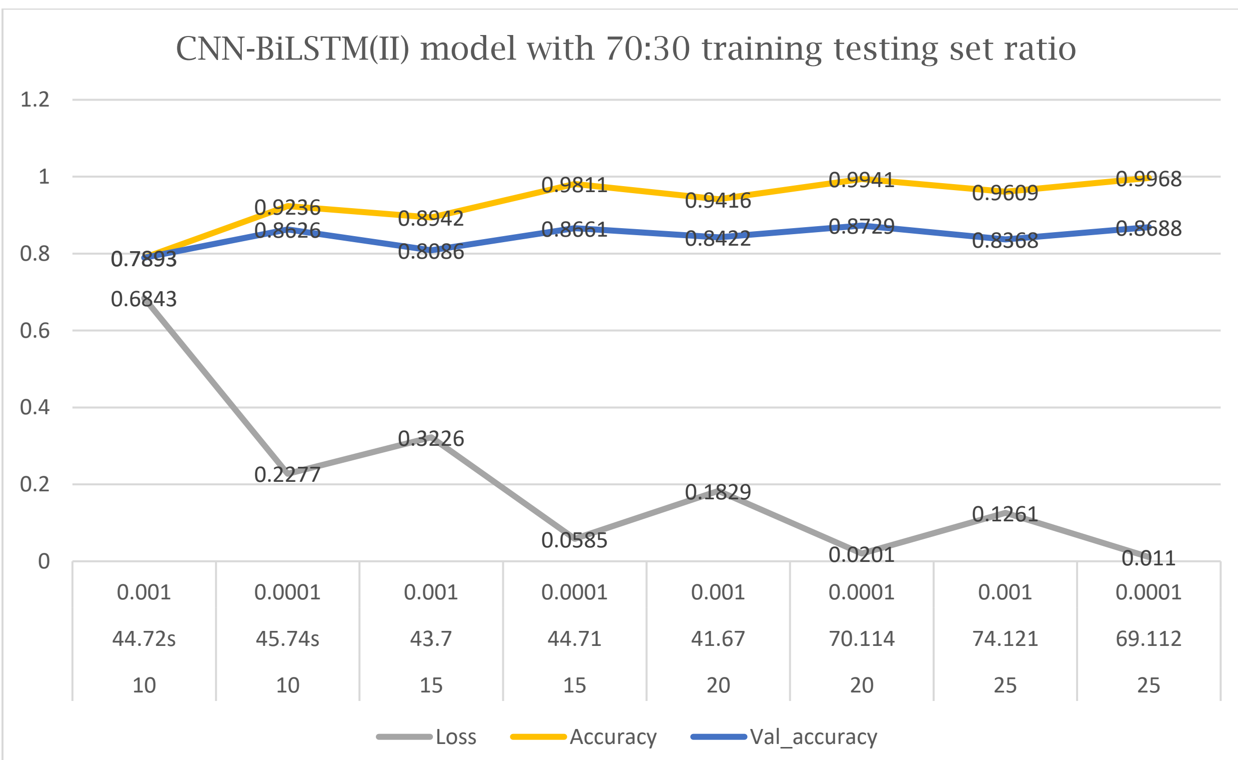


Figure 4. 28. CNN-BiLSTM (II) model with 70:30 training testing set ratio experimental result visualization

Therefore, from three experimental models in this condition/scenario CNN_BiLSTM (II) based model were selected, because our objective is to improve the performance of the recognition accuracy of the developed model with new data that has not been seen before.

In this scenario, the researcher selects a CNN_BiLSTM (II) based recognition model from Experment-6 with a 20-epoch number and a 0.0001 learning rate.

<u>Experiment results under Scenario II:</u> The researcher ran this experiment with a dataset split ratio of 80% for training and 20% for testing, as well as with different numbers of parameters and hyperparameters.

<table>
<tr><td rowspan="9">CNN model
model with
80:20 training testing set ratio</td><td>Experiment</td><td>#epoch</td><td>time</td><td>loss</td><td>Train. Accuracy</td><td>Val_accuracy</td><td>Learning_rate</td></tr>
<tr><td>Exp-1</td><td>10</td><td>35.51s</td><td>0.3133</td><td>0.8864</td><td>0.7750</td><td>0.001</td></tr>
<tr><td>Exp-2</td><td>10</td><td>20.29s</td><td>0.0340</td><td>0.9909</td><td>0.8058</td><td>0.0001</td></tr>
<tr><td>Exp-3</td><td>15</td><td>41.61s</td><td>0.1055</td><td>0.9625</td><td>0.7807</td><td>0.001</td></tr>
<tr><td>Exp-4</td><td>15</td><td>35.55s</td><td>0.0048</td><td>0.9970</td><td>0.8194</td><td>0.0001</td></tr>
<tr><td>Exp-5</td><td>20</td><td>20.29s</td><td>0.0528</td><td>0.9820</td><td>0.7892</td><td>0.001</td></tr>
<tr><td>Exp-6</td><td>20</td><td>37.55s</td><td>0.0040</td><td>0.9973</td><td>0.8164</td><td>0.0001</td></tr>
<tr><td>Exp-7</td><td>25</td><td>35.51s</td><td>0.0352</td><td>0.9879</td><td>0.7874</td><td>0.001</td></tr>
<tr><td>Exp-8</td><td>25</td><td>37.54s</td><td>0.0036</td><td>0.9973</td><td>0.8182</td><td>0.0001</td></tr>
<tr><td rowspan="9">CNN_BiLSTM (I) model with
80:20 training testing set ratio</td><td>Experiment</td><td>#epoch</td><td>Time</td><td>Loss</td><td>Accuracy</td><td>Val_accuracy</td><td>Learning_rate</td></tr>
<tr><td>Exp-1</td><td>10</td><td>29.42s</td><td>0.0886</td><td>0.9729</td><td>0.8419</td><td>0.001</td></tr>
<tr><td>Exp-2</td><td>10</td><td>36.53s</td><td>0.0436</td><td>0.9843</td><td>0.9055</td><td>0.0001</td></tr>
<tr><td>Exp-3</td><td>15</td><td>30.44s</td><td>0.1736</td><td>0.9437</td><td>0.8189</td><td>0.001</td></tr>
<tr><td>Exp-4</td><td>15</td><td>30.43s</td><td>0.0738</td><td>0.9746</td><td>0.9086</td><td>0.0001</td></tr>
<tr><td>Exp-5</td><td>20</td><td>29.43s</td><td>0.1164</td><td>0.9625</td><td>0.8326</td><td>0.001</td></tr>
<tr><td>Exp-6</td><td>20</td><td>31.46s</td><td>0.0292</td><td>0.9896</td><td>0.8913</td><td>0.0001</td></tr>
<tr><td>Exp-7</td><td>25</td><td>29.43s</td><td>0.0950</td><td>0.9707</td><td>0.8400</td><td>0.001</td></tr>
<tr><td>Exp-8</td><td>25</td><td>30.44s</td><td>0.0116</td><td>0.9957</td><td>0.8648</td><td>0.0001</td></tr>
</table>

CNN_BiLSTM (II) model with 80:20 training testing set ratio	Experiment	#epoch	Time	Loss	Accuracy	Val_accuracy	Learning_rate
	Exp-1	10	85.121	0.6664	0.7917	0.8135	0.001
	Exp-2	10	87.124	0.2362	0.9202	0.8647	0.0001
	Exp-3	15	93.132	0.3305	0.8906	0.8264	0.001
	Exp-4	15	77.110	0.0666	0.9776	0.8657	0.0001
	Exp-5	20	83.118	0.1993	0.9377	0.8426	0.001
	Exp-6	**20**	**78.111**	**0.0273**	**0.9916**	**0.8718**	**0.0001**
	Exp-7	25	84.120	0.1400	0.9553	0.8429	0.001
	Exp-8	25	116.165	0.0147	0.9957	0.8677	0.0001

Table 4. 4. Experiment results under Scenario II

In this scenario, researchers conduct similar operations and techniques with scenario one. The only difference is the percentage split of training and testing set of the dataset. So, in the CNN based model, it outperforms under Experment-4 at epoch 15 and learning rate 0.0001 with training accuracy of 0.9970 and 0.0048 training loss. Its validation accuracy becomes 0.8194 with a training time of 35.55 seconds. At epoch 15 and a learning rate of 0.0001, the CNN_BiLSTM(I)-based recognition model's training and validation accuracy are 0.9746 and 0.9086, respectively, with a training loss of 0. 0738 and a training time of 30.43 seconds under Experiment-4. And the 3rd model CNN_BiLSTM (II) based model under Experiment-6 at epoch 20 and a learning rate of 0.0001, the CNN_BiLSTM (II)-based recognition model's training and validation accuracy are 0.9916 and 0.8718, respectively, with a training loss of 0.0273 and a training time of 78.111 seconds.

Let’s visualize the above experimental result experimental result under second experimental scenarios.

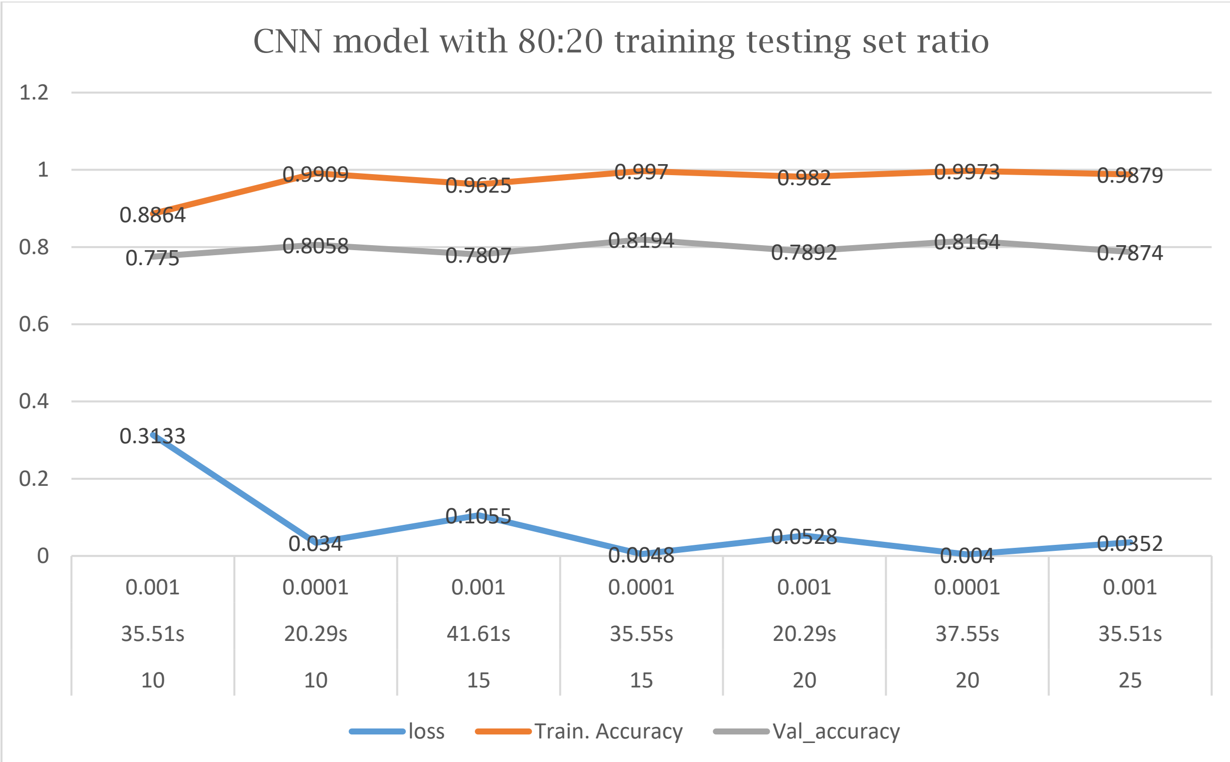


Figure 4. 29. CNN model with 80:20 training testing set ratio experimental result visualization

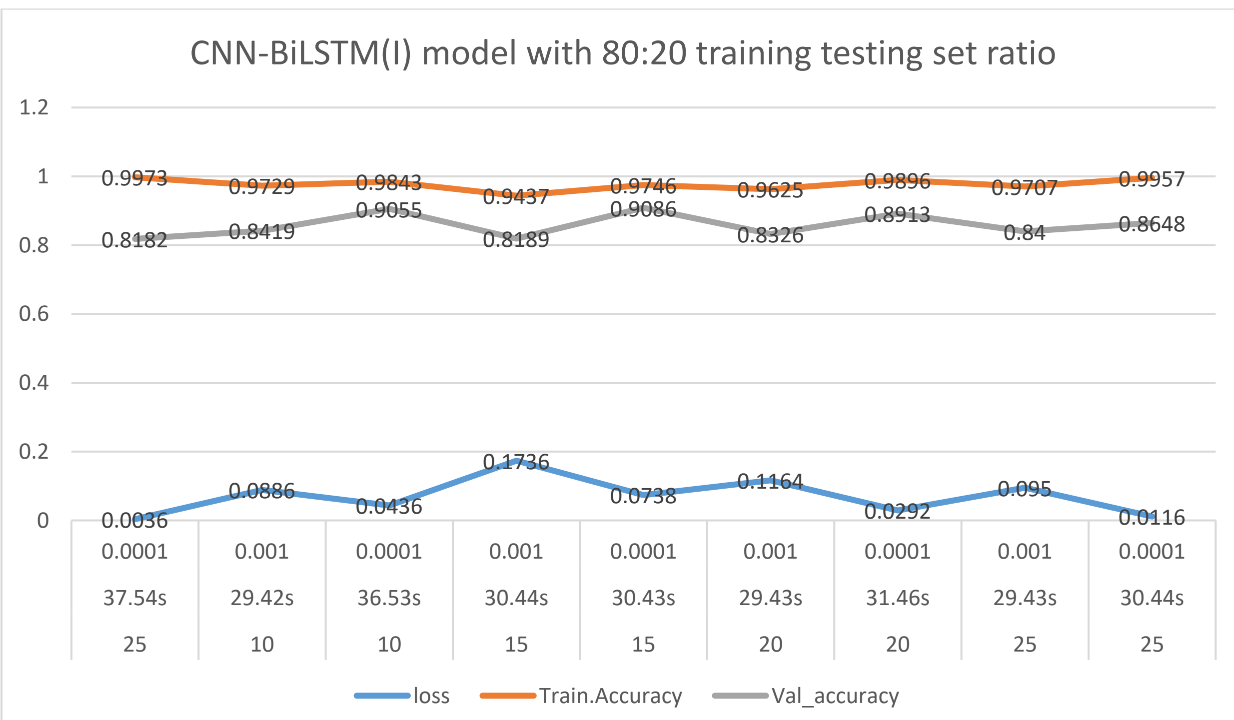


Figure 4. 30. CNN-BiLSTM(I) model with 80:20 training testing set ratio experimental result visualization

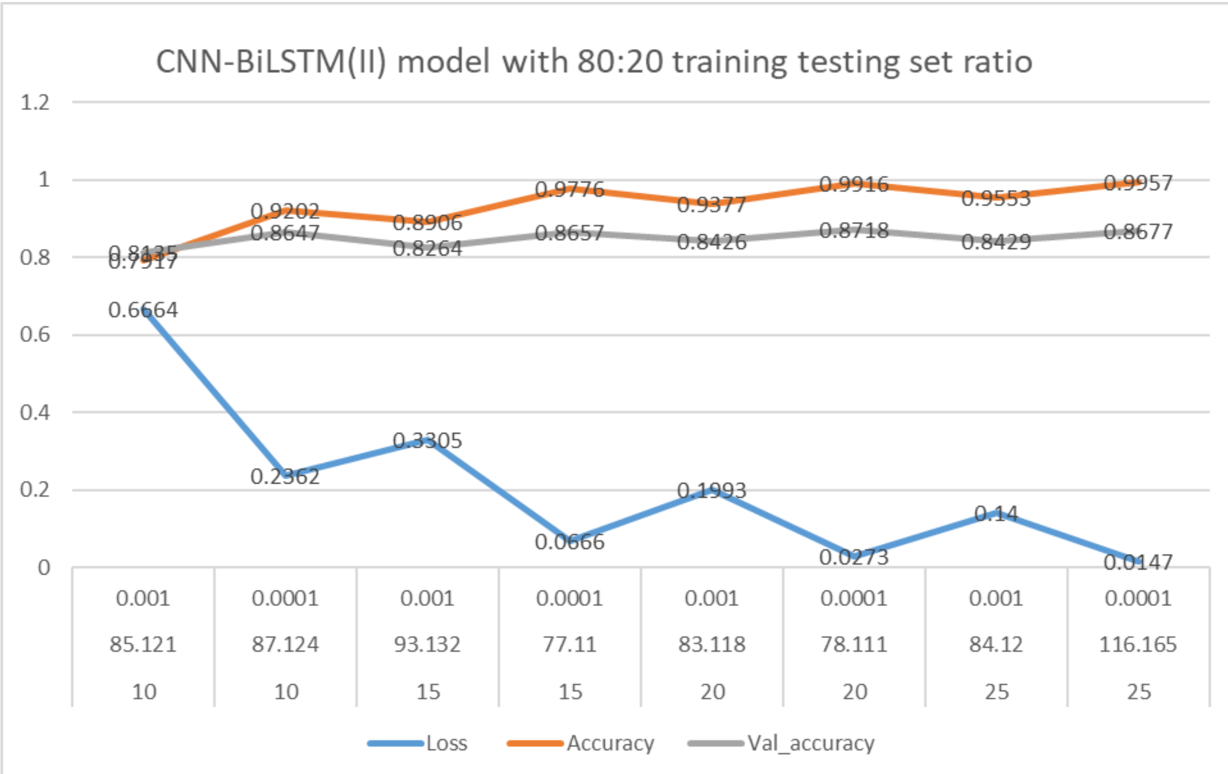


Figure 4. 31. CNN-BiLSTM (II) model with 80:20 training testing set ratio experimental result visualization

From the above two scenarios, the researcher selects the following experimental results to compete themselves for the final model selection for recognition of manuscripts.

Model	Experiment	scenario	#epoch	Time	Loss	Accuracy	Val_accuracy
CNN	4	I	15	30.50s	0.0033	0.9979	0.8209
CNN-BiLSTM(I)	4	I	15	45.75s	0.0138	0.9954	0.8441
CNN-BiLSTM (II)	6	I	20	70.114s	0.0201	0.9941	0.8729
CNN	4	II	15	35.55s	0.0048	0.997	0.8194
CNN-BiLSTM(I)	**4**	**II**	**15**	**30.43s**	**0.0738**	**0.9746**	**0.9086**
CNN-BiLSTM (II)	6	II	20	78.111s	0.0273	0.9916	0.8718

Table 4. 5. promising result of three models under each scenario

The researcher presented the models that outperformed in two different scenarios in table 4.5, going to compare them to select one of the best models from the above as the final research model recommendation for Ancient Amharic and Ge’ez manuscript recognition.

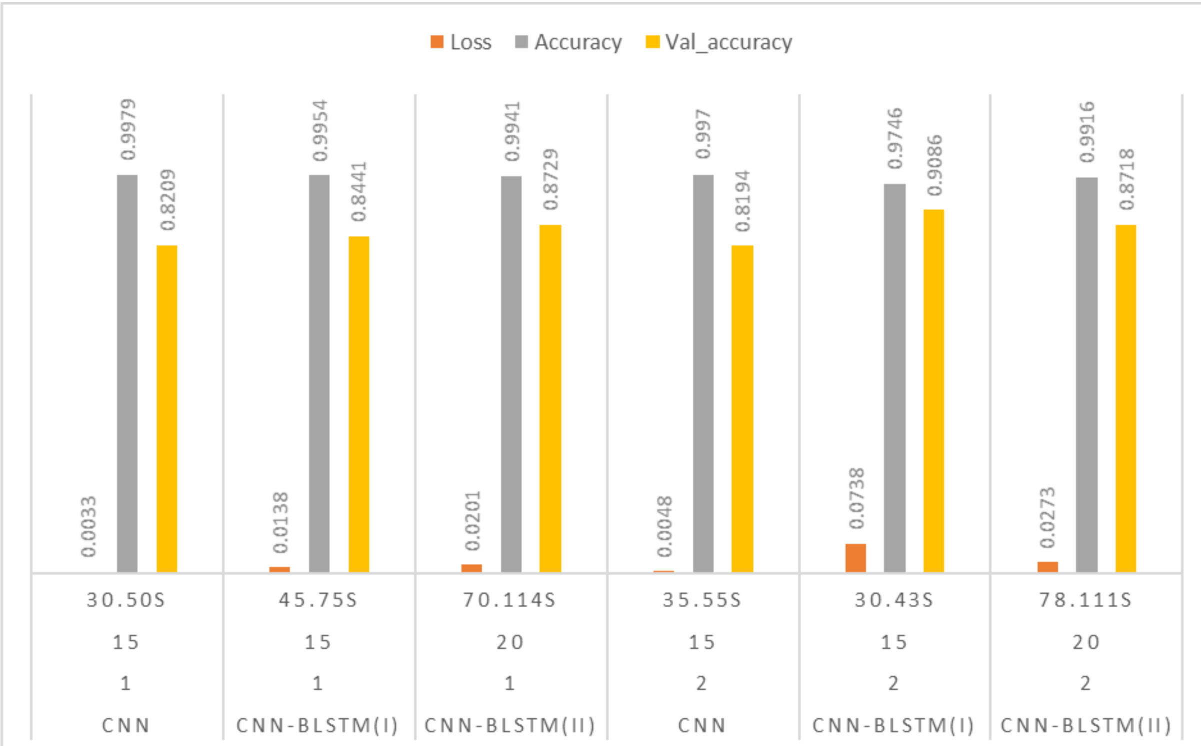


Figure 4. 32. competent Models for final model selection

As we discussed in the above section, we have selected best experimental models one from scenario one and one from experimental scenario two. These selected models were CNN-BiLSTM(I) model from scenario one and CNN-BiLSTM (II) model from scenario two. Next, the researcher chooses one of the best models from selected model from the two competent model presented. Therefore, the researcher selects **CNN_BiLSTM(I)** model from above three experimental models which have done under scenario two and Experiment-4, because one of the objectives of the study to improve the performance of the recognition accuracy of the developed model with new data that has not been seen before.

In this research, the researcher selects a **CNN_BiLSTM(I)** based recognition model that yields training accuracy of 97.46%, validation accuracy of 90.86%, training time of 30.43 seconds with 15-epoch number and a 0.0001 learning rate and percentage split of 80%:20%.

```
plt.plot(history1.history['loss'])
plt.plot(history1.history['accuracy'])
plt.plot(history1.history['val_accuracy'])
plt.title('Visualization of loss, Accuracy and validation accuracy')
plt.xlabel('epoch')
plt.legend(['loss', 'accuracy','val_accuracy'], loc='center right')
plt.show()
```

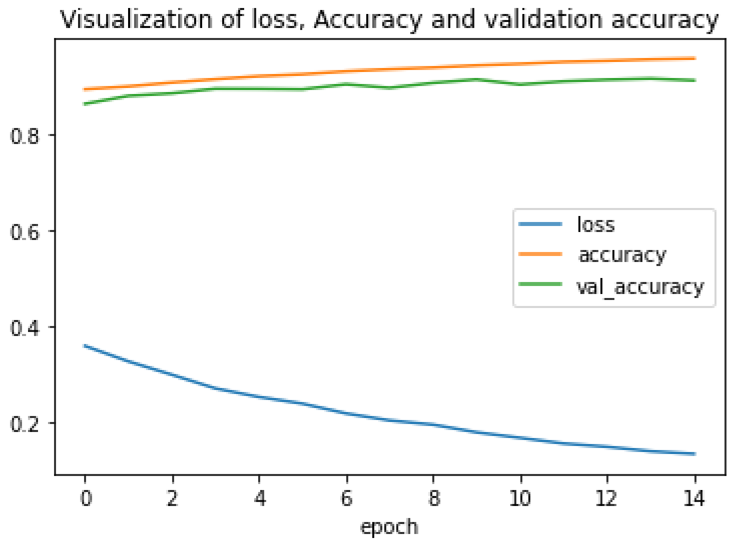


Figure 4. 33. Visualization of loss, training accuracy and validation accuracy of selected CNN-BiLSTM(I) model.

```
plt.plot(history1.history['val_accuracy'],label='val_accuracy')
plt.plot(history1.history['accuracy'],label='accuracy')
plt.legend()
plt.show()
```

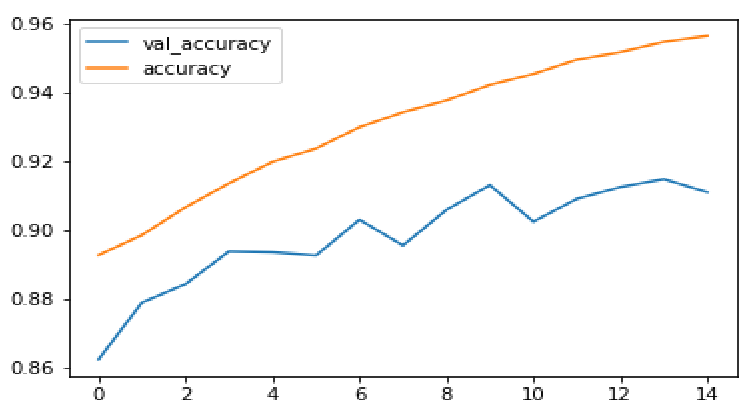


Figure 4. 34. visualization of training accuracy vs validation accuracy of selected CNN-BiLSTM(I) model.

4.6. Prototype Development

The user interface is a method of communicating with the end user that allows them to easily interact with the developed Ancient Ethiopic Manuscript Recognition system. The system's end-user can launch the web application by entering the URL of the system into any web browser; the web browser then sends HTTP requests to the web server, which responds with an HTTP response to the browser.

The recognition system provides the following functionalities: document image loading, perform preprocessing activity and recognitions of the given input document image finally the system generates editable text format of the given image. the proposed GUI is shown in figure 4.35.

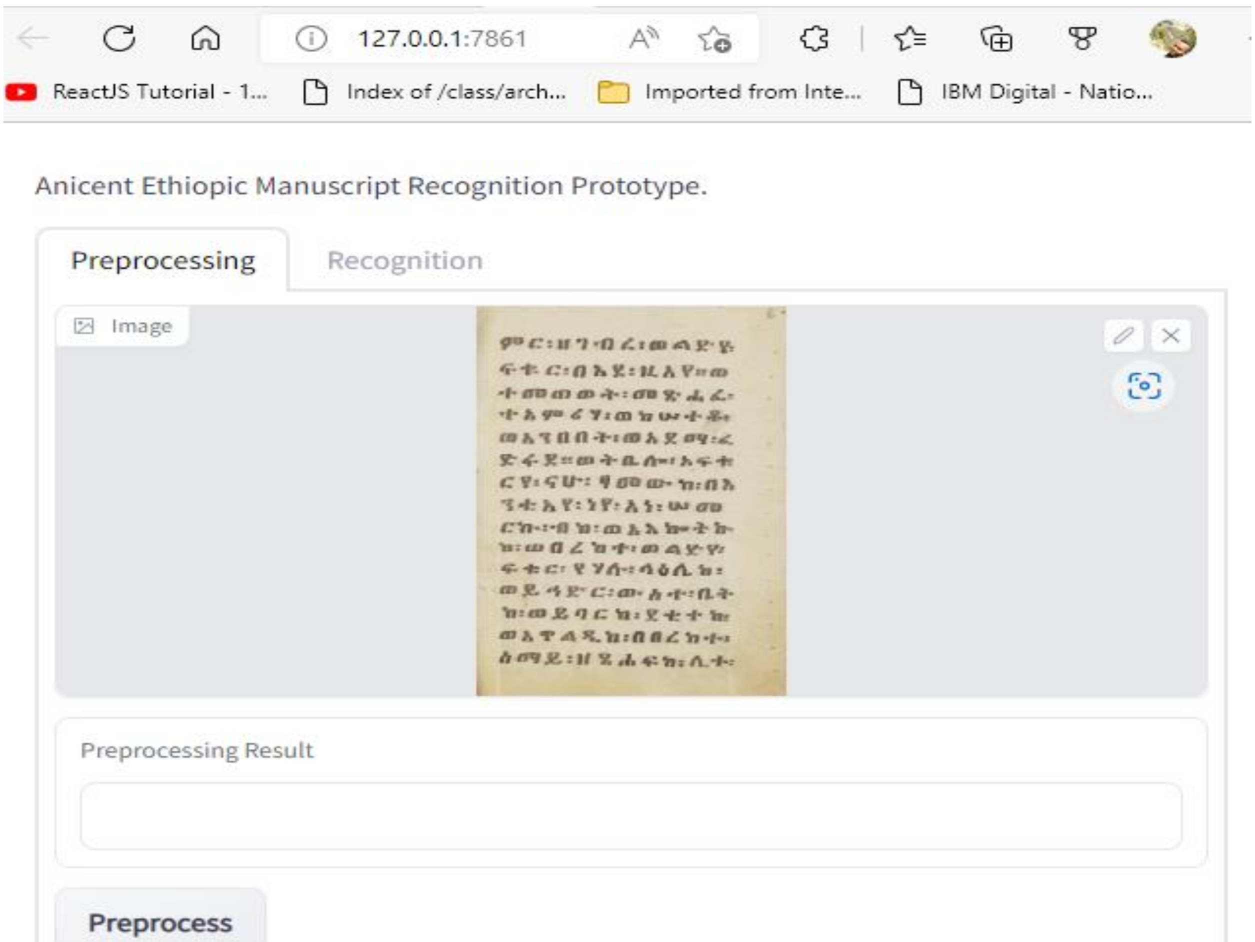


Figure 4. 35. Prototype of Ancient Ethiopic manuscript Recognition System

4.7. Result Analysis of developed prototype

The prototype developed and tested on various manuscript document images is 12-page in length. The datasets that are tested are those that are described in the dataset preparation section. The prototype system then generates the following recognition accuracy rate. Table 4.6 displays the hybrid of CNN and BiLSTM recognition accuracy with the given testing dataset.

Manuscript image code	Total number of characters	Correctly recognized Number of characters	Recognitions accuracy in percent
MI1	163	58	35.58
MI2	205	71	34.63
MI3	243	61	25.10
MI4	60	12	20
MI5	251	79	31.47
MI6	103	22	21.35
MI7	201	67	33.33
MI8	159	47	29.55
MI9	173	58	33.52
MI10	137	43	31.38
MI11	213	70	32.86
MI12	197	64	32.48
Average recognitions accuracy on testing data			**30.10%**

Table 4. 6. Recognitions Accuracy rate of recognition system

The average accuracy of various document images is presented in this table. In the case of recognition, performance is reduced due to unremoved noise on the document images during the noise removing tasks; morphological similarity of some characters in nature; in the case of segmentation problems, as shown in Table 4.6.

Therefore, the proposed recognition system has a testing accuracy score of 30.1% on new and unknown testing data.

The researchers conducted experiments on various preprocessing techniques in this study. OTSU's threshold for binarizing the digitized image, bi-level filtering for noise removal, and three-level segmentation, which is line, word, and character. The main factors influencing the result of recognition raised from the quality of the manuscript image, inconsistency of characters' shapes, font size and style, corrosiveness of writing style and morphological similarity of characters. In addition to this, segmentation also plays an important role for performance recognition because during segmentation it may segment important strokes from one part of a character image that is important for recognition as well as this leads to variation of character features for certain character images, which leads to misrecognition.

4.8. Manual Quality Evaluation

As previously stated, we evaluated the proposed model's recognition results using a training, validation, and testing accuracy evaluation metric. A manual evaluation is used in addition to this automatic evaluation to determine whether the proposed model is acceptable. As a result, we asked and interviewed seven domain experts to evaluate the recognition system prototype. These scholars were specifically chosen because they are familiar with Ethiopian Orthodox Church manuscript documents and have sufficient knowledge to evaluate results and corrective actions.

Table 4.7 demonstrates how the testing images are prepared for manual evaluation, as well as the Points of Recognition Quality (PRQ) and their corresponding values.

1. ምንም እውቅና የለውም = 0 (Nothing recognized = 0) Or Bad recognized (abbreviated as NR)
2. ብዙው አልታወቀም = 1 (Most not recognized = 1) (abbreviated as MnR)
3. በከፊል አልታወቀም = 2 (Partially not recognized = 2) (abbreviated as PnR)
4. ታውቋል ማለት ይቻላል = 3 (Almost recognized = 3) (abbreviated as AR)
5. በትክክል እውቅና አግኝቷል = 4 (Correctly recognized = 4) Or Nice recognized (abbreviated as CR)

The identification code of the images used for evaluation	የእውቅና ደረጃ					
	Points of Recognition Quality (PRQ)					Average (4pts)
	NR (0)	MnR (1)	PnR (2)	AR (3)	CR (4)	
MI1			1(14.28%)	6 (85.71%)		2.86 (71.5%)
MI2		1(14.28%)	1(14.28%)	5 (71.42%)		2.57 (64.25%)
MI3			5(71.42%)	2 (28.57%)		2.28 (57%)
MI4		1(14.28%)	3(42.85%)	3 (42.85%)		2.28 (57%)
MI5			1(14.28%)	6 (85.71%)		2.86 (71.5%)
MI6		1(14.28%)	4(57.14%)	2 (28.57%)		2.14 (53.5%)
MI7			1(14.28%)	6 (85.71%)		2.86 (71.5%)
MI8			2(28.57%)	5 (71.42%)		2.71 (67.75%)
MI9		1(14.28%)		6 (85.71%)		2.71 (67.75%)
MI10			2(28.57%)	5 (71.42%)		2.71 (67.75%)
MI11			1(14.28%)	6 (85.71%)		2.86 (71.5%)
MI12		1(14.28%)	1(14.28%)	5 (71.42%)		2.57 (64.25%)
Total Averages		5(5.95%)	22(26.2%)	57 (67.86%)		**2.62 (65.5%)**

Table 4. 7 proposed system manual quality evaluation

The average is computed as shown in the equation.

$$Avarage = \frac{\sum(PRQ * ValueGivenByEvaluator)}{Total\ number\ of\ Evaluator}$$

In Table 4.7, for example, the average of the " MI2" could be calculated as 1*1 +1*2 + 5*3 (1 evaluator gave it 1, 1 evaluator gave it 1 and 5 evaluators gave it 3) divided by the total number of evaluators, which is 7, and 18/7 equals 2.57. Finally, the table shows that an average of 65.5 percent of manual recognition quality was recorded. And evaluators commented that the study produced their best results yet and that the study demonstrates how we can shift our way of preserving manuscripts and converting them to computer systems. However, they observe that the recognition performance needs to be improved.

4.9. Discussions

This section presents the discussion based on the finding of this study. Therefore, the researcher has discussed the result about the research questions are answered or not which is sated under chapter one. As remembered, the major objectives of the study were developed Amharic and Ge’ez manuscript recognitions model using deep learning approaches with convolutional neural network and the hybrids of convolutional neural network and bidirectional Long-short term memory. This study was developed three different model architecture with the stated deep learning algorithms. Finally, the researcher chooses hybrid of CNN and BiLSTM, as a result of both training, validation, and testing recognition accuracy outperform under this algorithm.

In this study, works have been made to find an answer to the following listed questions

- Which deep learning technique best work for the recognition of Ancient Ge’ez and Amharic manuscript?

In this study, two different deep learning algorithms are applied to develop ancient Amharic and Ge’ez manuscript recognitions model, namely CNN and hybrid of CNN and BiLSTM algorithms. Accordingly, researcher proposed three different model architecture based on these deep learning algorithms by using different parameters and hyperparameters. comparatively, the Hybrid of CNN and BiLSTM model has given very encouraging performance for the recognitions of Ancient Amharic and Ge’ez manuscripts.

- To what extent does the developed model correctly recognize manuscripts in Amharic and Ge’ez scripts?

Following the development of any recognition model, it is always necessary to determine whether or not the final recognition power of the developed model is correct. The developed recognizer model's performance was evaluated in this study using training complexity, training and validation accuracy. The experimental results showed 97.46 percent accuracy, 90.86 percent of validation accuracy, 30.1 percent of testing accuracy.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

This study aimed to propose a recognition system for Ancient Amharic and Ge’ez ancient manuscript recognition using deep learning techniques, which is a hybrid of two deep learning algorithms, those being CNN and BiLSTM Algorithms. During developing this model, there are different tasks. Those are, from the beginning, one data accusation up to the recognition stage. The following stages should be implemented. First should perform image data gathering, image preprocessing, image segmentation, feature extraction, image classification and recognition.

During the experiment, for image binarization, different image binarization algorithms were implemented, and we selected OTSU’s global thresholding algorithms for other implemented algorithms. For noise removal techniques, bi-level noise filtering methods are implemented. And line, word and character segmentation have been implemented in order to extract lines of manuscript images, word level segmentation in order to segment segmented line to word segment and character level segmentation in order to extract character images from manuscript images. While extracting characters from manuscript images, there are lots of character images extracted from those images. Researchers constructed 39084 image datasets for the study from collected Ancient Amharic and Ge’ez manuscript documents and 705 collected image data from those 11 different image document sources.

Finally, the implemented model outperformed with a 97.46% accuracy and an 90.84% validation accuracy with testing Accuracy of 30.10%. validation accuracy is affected by similarity of manuscript character images, noise, inappropriate segmentation, and inconsistency of writing style across the manuscript document. Character shapes that may be classified or recognized wrongly due to their shape are ደ፣ይ፣ጰ, ተ፣ቀ, ጠ፣ወ፣ጬ, ሰ፣ለ, and ነ፣ኀ etc.

In this research, we are focused on ancient Amharic and Ge’ez manuscript recognition collected from different document sources. Some of the researcher's contributions are the following:

1. Analyze the recognition process and challenges, and respond to those challenges or apply the researcher’s knowledge to address those challenges and report challenges that will occur during image recognition or optical character recognition.

2. In image processing, the most difficult task is image dataset preparation, especially for manuscript recognition datasets. During dataset preparation, the hardest task is image segmentation. Meanwhile, this Ancient Amharic and Ge’ez document dataset has been constructed.

5.2. Recommendations

The study focused on ancient Amharic and Ge’ez manuscript document recognition. Although the study achieves better results, the researcher believes that further study could be conducted on this study area to improve this result of recognition in the coming future.

- The researcher recommend that a study conducted further with word and sentence level image segmentation to eliminate inappropriate image segmentation processes and inappropriate character image segmentation.
- The researcher recommends that future studies be conducted by including characters that are included in Amharic script but not under Ge’ez script called labialized character (27 in number such as ቷ, ሟ, ሏ etc) as well as 8 punctuation marks.
- Further study is required with different CNN and BiLSTM deep learning algorithms or other deep learning algorithms with different parameters and hyperparameters.
- Further study is required for Ethiopic manuscript recognition using end to end recognition.

Reference

[1] C. U. Prof. Swapna Banerjee, DLIS, “Overview of manuscripts.”

[2] “Sewasew | Manuscripts (የእጅ ጽሑፎች).” https://en.sewasew.com/p/manuscripts-(የእጅ-ጽሑፎች) (accessed Jun. 01, 2022).

[3] F. Abdurahman, “Handwritten Amharic character recognition using a convolutional neural network,” *NWSAENS*, vol. 7231, no. November 2018, pp. 71–87, 2019, doi: 10.12739/NWSA.2019.14.2.1A0433.Abdurahman.

[4] A. T. Hurisa, “Handwritten Ancient Saint Yared Huymn Script Recognition System Using Artifical Neural Network,” 2018.

[5] J. Albrecht and W. Lehner, “On-line analytical processing in distributed data warehouses,” in *International Database Engineering and Applications Symposium*, 1998, no. figure 1, pp. 78–85. doi: 10.1109/IDEAS.1998.694361.

[6] F. Ashagrie and D. Boran, “Ancient Geez script recognition using deep learning,” Springer International Publishing, 2019. doi: 10.1007/s42452-019-1340-4.

[7] S. Kelly, “A Companion to Medieval Ethiopia and Eritrea,” *A Companion to Mediev. Ethiop. Eritrea*, vol. 1350, pp. 8–9, 2019, doi: 10.1163/9789004419582.

[8] D. N. and A. Bausi, “Cataloguing Ethiopic manuscripts : update and overview on ongoing work,” 2018.

[9] www.tutorialspoint.com, “Artificial Intelligence Tutorial.” https://www.tutorialspoint.com/artificial_intelligence/index.htm (accessed Jul. 16, 2022).

[10] Sudhir Allam, “Artificial Intelligence and its Applications,” *Int. Eng. J. Res. Dev.*, vol. 2, no. 1, pp. 1–6, 2020.

[11] “What is Optical Character Recognition and How Can AI Give it a Boost?” https://appen.com/blog/optical-character-recognition/ (accessed Jun. 01, 2022).

[12] S. Getu, “Addis Ababa Institute of Technology School of Electrical and Computer Engineering Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Network,” *J. Addis abeba Univ.*, pp. 1–109, 2016.

[13] S. Abebe, “Bilingual Script Identification for Optical Character Recognition of Amharic and English Printed Document,” *J. Addis abeba Univ.*, pp. 1–102, 2011.

[14] B. Y. Reta, D. Rana, and G. V. Bhalerao, “Amharic Handwritten Character Recognition Using Combined Features and Support Vector Machine,” *Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018*, no. Icoei, pp. 265–270, 2018, doi: 10.1109/ICOEI.2018.8553947.

[15] N. K. Tamiru, “Amharic Sign Language Recognition based on Amharic Alphabet Signs,” *J. Addis abeba Univ.*, pp. 1–90, 2018.

[16] P. Singh and S. Budhiraja, “Feature Extraction and Classification Techniques in O.C.R. Systems for Handwritten Gurmukhi Script-A Survey,” *Int. J. Eng. Res. Appl. www.ijera.com*, vol. 1, no. 4, pp. 1736–1739, [Online]. Available: www.ijera.com

[17] C. Tensmeyer and T. Martinez, “Historical Document Image Binarization: A Review,” *SN Comput. Sci.*, vol. 1, no. 3, 2020, doi: 10.1007/s42979-020-00176-1.

[18] C. C. Tappert and S. H. Cha, *English Language Handwriting Recognition Interfaces*. Elsevier Inc., 2007. doi: 10.1016/B978-012373591-1/50006-1.

[19] “Ethiopian Languages - Semitic, Cushitic, Omotic and Nilo-Saharan.” http://www.ethiopiantreasures.co.uk/pages/language.htm (accessed Jun. 16, 2022).

[20] Shiferaw Tegen, “Optical Character Recognition for Ge’ez Scripts Written on the Vellum,” *J. gonder Univ.*, pp. 1–93, 2017.

[21] L. I. Lied, “Ancient Manuscripts in Digital Culture,” *Anc. Manuscripts Digit. Cult.*, pp. 15–29, 2019, doi: 10.1163/9789004399297.

[22] F. A. Demilew, “Ancient Geez Script Recognition Using Deep Convolutional Neural Network,” Near East University, 2019.

[23] Y. Yilma, “Ethiopian Manuscript Heritage: Ethiopic Manuscript and the Role of ARCCH.”

[24] T. G. Mariam, “Ethiopian Manuscripts and Archives: Challenges and Prospects,” *J. Archaeol. Egypt/Egyptology*, vol. 17, no. 10, pp. 4214–4227, 2020.

[25] G. Haile, “The Limits of Traditional Methods of Preserving Ethiopian Ge’ez Manuscripts,” *Libri*, vol. 68, no. 1, pp. 33–42, 2018, doi: 10.1515/libri-2017-0004.

[26] data-flair, “Top 25 Computer Vision Project Ideas for 2021 - DataFlair,” *data-flair*. https://data-flair.training/blogs/computer-vision-project-ideas/ (accessed Jan. 04, 2021).

[27] data-flair, “AI - Python Computer Vision Tutorial with OpenCV - DataFlair,” *data-flair*. https://data-flair.training/blogs/ai-python-computer-vision/ (accessed Jan. 04, 2021).

[28] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, “Computer vision technology in agricultural automation —A review,” *Inf. Process. Agric.*, vol. 7, no. 1, pp. 1–19, 2020, doi: 10.1016/j.inpa.2019.09.006.

[29] R. B. milan sonka , vaclav Hiavac and, *Milan Sonka_ Václav Hlavác_ Roger Boyle-Image processing, Analysis, and Machine Vision-Cengage Learning (2015),* Fourth edi. united kingdom, australia,brazil, japan,mexico,singapore,united states: cengage learning.

[30] V. Wiley and T. Lucas, “Computer Vision and Image Processing: A Paper Review,” *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 28–36, 2018, doi: 10.29099/ijair.v2i1.42.

[31] O. Babatunde, L. Armstrong, D. Diepeveen, and J. Leng, “A survey of computer-based vision systems for automatic identification of plant species,” vol. 6, no. 1, pp. 61–71, 2015, doi: 10.17700/jai.2015.6.1.152.

[32] “Ashwin Pajankar - Raspberry Pi Computer Vision Programming_ Design and implement computer vision applications with Raspberry Pi, OpenCV, and Python 3, 2nd Edition-Packt Publishing (2020).”

[33] S. Shawal, M. Shoyab, and S. Begum, “Fundamentals of Digital Image Processing and Basic Concept of Classification,” *Int. J. Chem. Process Eng. Res.*, vol. 1, no. 6, pp. 98–108, 2014, doi: 10.18488/journal.65/2014.1.6/65.6.98.108.

[34] 磯崎行雄, 丸山茂徳, and 柳井修一, “Image processing techniques. Part 1,” 地学雑誌, vol. 119, no. 2, pp. 187–189, 2010, [Online]. Available: http://ci.nii.ac.jp/naid/40017128637

[35] M. Bhat, “Digital Image Processing,” vol. 3, no. 1, pp. 272–276, 2014.

[36] I. Young, J. Gerbrands, and L. van Vliet, “Fundamentals of Image Processing,” pp. 1–85, 2009, doi: 10.1201/9781420046090-c13.

[37] W. I. Nahy, “Types of Digital Images,” pp. 1–13.

[38] P. P. and S.Muthuselvi, “Digital image processing Techniques - A survey,” *Golden Res. Thoughts J.*, no. May 2016, 2017.

[39] S. V Khedaskar, M. A. Rokade, B. R. Patil, and P. N. Tatwadarshi, “A Survey of Image Processing and Identification Techniques,” vol. 1, no. 1, pp. 1–10, 2018.

[40] S. Mathur, R. Purohit, and A. Vyas, “A Review on basics of Digital Image Processing,” vol. 4, no. 12, pp. 1–3, 2016.

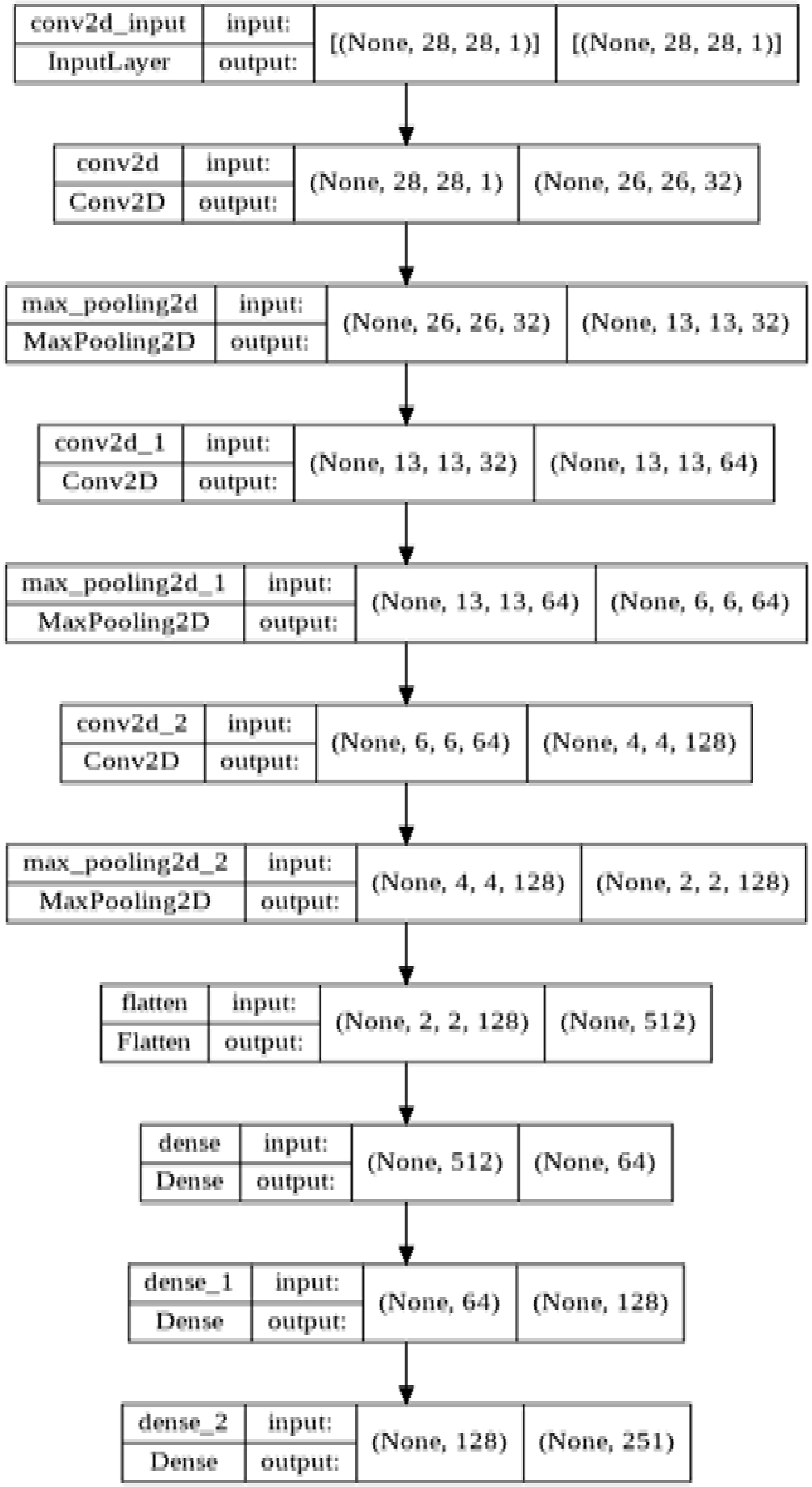
[41] S. G. Dafe and S. S. Chavhan, “Optical Character Recognition Using Image Processing,” *Int. Res. J. Eng. Technol.*, pp. 962–964, 2018, [Online]. Available: www.irjet.net

[42] A. Purohit and S. S. Chauhan, “A Literature Survey on Handwritten Character Recognition,” vol. 7, no. 1, pp. 1–5, 2016.
[43] S. G. Dedgaonkar, A. A. Chandavale, and A. M. Sapkal, “Survey of Methods for Character Recognition,” *Int. J. Eng. Innov. Technol.*, vol. 1, no. 5, pp. 180–189, 2012.
[44] M. Sonkusare, R. Gandhi, and P. Vishwavidyalaya, “A Study on Optical Character Recognition Techniques,” *Int. J. Comput. Sci. Inf. Technol. Control Eng.*, vol. 4, no. January 2017, pp. 1–14, 2019, doi: 10.5121/ijcsitce.2017.4101.
[45] R. Jana and A. R. Chowdhury, “Optical Character Recognition from Text Image,” *Int. J. Comput. Appl. Technol. Res.*, vol. 3, no. 4, pp. 239–243, 2014.
[46] S. R. Zanwar, U. B. Shinde, A. S. Narote, and S. P. Narote, “A Comprehensive Survey On Soft Computing Based Optical Character Recognition Techniques,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 978–987, 2019.
[47] A. Shimeles, “Online Handwriting Recognition for Ethiopic Characters for Ethiopic Characters By Abnet Shimeles,” *J. Addis abeba Univ.*, pp. 1–119, 2005.
[48] S. Tomar and A. Kishore, “A Review: Optical Character Recognition,” *Int. J. Eng. Sci. Res. Technol.*, vol. 7, no. 4, pp. 233–238, 2018.
[49] “Difference between Optical Character Recognition (OCR) and Magnetic Ink Character Reader (MICR) - GeeksforGeeks.” https://www.geeksforgeeks.org/difference-between-optical-character-recognition-ocr-and-magnetic-ink-character-reader-micr/ (accessed Jun. 02, 2022).
[50] R. Y. Choi, A. S. Coyner, J. Kalpathy-cramer, M. F. Chiang, and J. P. Campbell, “Introduction to Machine Learning , Neural Networks , and Deep Learning,” *Transl. Vis. Technol.*, vol. 9, no. special issue 2, pp. 1–12, 2020.
[51] R. B. and T. S. 2018 Dipanjan Sarkar, “Machine Learning Basics,” 2018, pp. 3–65.
[52] R. A. Trishan Panch , Peter Szolovits, “Artificial intelligence, machine learning and health systems,” vol. 8, no. 2, pp. 1–8, 2018, doi: 10.7189/jogh.08.020303.
[53] V. Jatana, *Dive Into Machine Learning*, no. October. 2018, pp. 1–30.
[54] J. Kirsch, Daniel and Hurwitz, *Machine Learning for dummies*, IBM Limite. John Wiley & Sons, Inc, 2018.
[55] O. Theobald, *Machine Learning For Absolute Beginners*, 2nd ed. 2017.
[56] Y. Baştanlar and M. Özuysal, “Introduction to machine learning,” *Methods Mol. Biol.*, vol. 1107, pp. 105–128, 2014, doi: 10.1007/978-1-62703-748-8_7.
[57] S. Vieira, W. H. Lopez Pinaya, and A. Mechelli, “Introduction to machine learning,” *Mach. Learn. Methods Appl. to Brain Disord.*, pp. 1–20, 2019, doi: 10.1016/B978-0-12-815739-8.00001-8.
[58] A. H. and D. Fleet, *Machine Learning and Data Mining Lecture Notes*. Computer Science Department University of Toronto, 2012.
[59] “Time Series Forecasting as Supervised Learning.” https://machinelearningmastery.com/time-series-forecasting-supervised-learning/ (accessed Jun. 14, 2022).
[60] “Active learning machine learning: What it is and how it works - DataRobot AI Cloud.” https://www.datarobot.com/blog/active-learning-machine-learning/ (accessed Jun. 14, 2022).
[61] “4 Types of Classification Tasks in Machine Learning.” https://machinelearningmastery.com/types-of-classification-in-machine-learning/ (accessed Jun. 14, 2022).

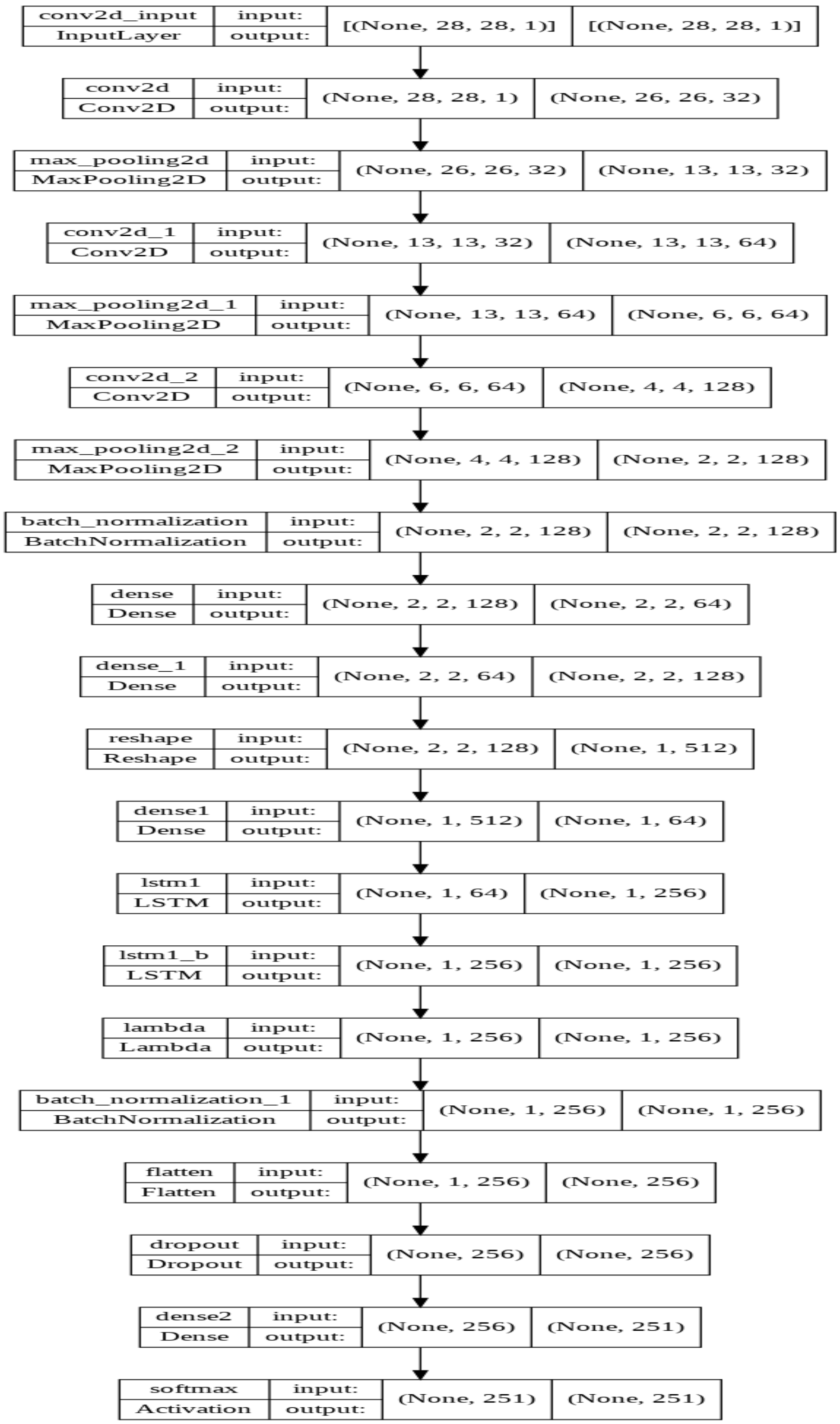
[62] "Supervised vs Unsupervised Learning: Key Differences." https://www.guru99.com/supervised-vs-unsupervised-learning.html (accessed Jun. 14, 2022).
[63] "Supervised, Unsupervised, And Semi-Supervised Learning With Real-Life Usecase." https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning (accessed Jun. 14, 2022).
[64] "Supervised and Unsupervised Machine Learning Algorithms." https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (accessed Jun. 14, 2022).
[65] A. beleete Mekonnen, "A Deep Learning Based Approch for Potato Leaf Diseases Classification," 2020.
[66] guru99, "Difference Between Deep Learning and Machine Learning Vs AI," *guru99.com*. https://www.guru99.com/machine-learning-vs-deep-learning.html (accessed Jan. 04, 2021).
[67] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision : A Brief Review," *Comput. Intell. Neurosci.*, pp. 1–13, 2018.
[68] P. Ongsulee, "Artificial Intelligence , Machine Learning and Deep Learning," *Fifteenth Int. Conf. ICT Knowl. Eng. Artif.*, pp. 1–12, 2017.
[69] T. W. Guide, *Introduction to Machine Learning The Wikipedia Guide*.
[70] R. B. and T. S. Dipanjan Sarkar, "Machine Learning Basics," *Springer Link*, pp. 3–65, 2020.
[71] S. K. Shetty and A. Siddiqa, "Deep Learning Algorithms and Applications in Computer Vision Savita," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 7, pp. 195–201, 2019.
[72] S. Skansi, "Machine Learning Basics," *Springer Int. Publ. AG*, pp. 51–77, 2018, doi: 10.1007/978-3-319-73004-2.
[73] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 2020.
[74] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," vol. 521, pp. 1–9, 2015, doi: 10.1038/nature14539.
[75] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting," *Sci. Direct Procedia Comput. Sci.*, vol. 135, pp. 89–98, 2018, doi: 10.1016/j.procs.2018.08.153.
[76] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting," in *Science Direct Procedia Computer Science*, 2018, vol. 135, pp. 89–98. doi: 10.1016/j.procs.2018.08.153.
[77] Özal Yildirim, "A Novel Wavelet Sequences Based on Deep Bidirectional LSTM Network Model for ECG Signal Classification," 2019.
[78] A. Melaku, "preprocessing of mobile captured document images." doi: 10.4000/etudesafricaines.5928.
[79] A. T. Birhanu and A, "Amharic Character Recognition System for Printed Real-Life Documents," *J. Addis abeba Univ.*, pp. 1–124, 2010.
[80] B. Mengistu, "Restoration and Retrieval of Historical Amharic document images," 2014.
[81] S. N. Ayalew, "recognition of the Ethiopian car plate," *J. Addis abeba Univ.*, pp. 1–79,

2013.
[82] E. G. Beyene, “Handwritten and Machine printed OCR for Geez Numbers Using Artificial Neural Network,” *arXiv*, no. Nips, 2019.
[83] B. Asnake, “Retrieval From Real-Life Amharic Document Images,” 2012.
[84] L. Koneru, “Using Design Science Research to Develop a Conceptual Solution for Improving Knowledge Sharing in a Virtual Workspace,” 2018.
[85] A. Alturki, G. G. Gable, and W. Bandara, “The design science research roadmap: In progress evaluation,” *Proc. - Pacific Asia Conf. Inf. Syst. PACIS 2013*, 2013.
[86] Ł. Ostrowski, M. Helfert, and F. Hossain, “A conceptual framework for design science research,” *Lect. Notes Bus. Inf. Process.*, vol. 90 LNBIP, pp. 345–354, 2011, doi: 10.1007/978-3-642-24511-4_27.
[87] T. tuunanen Ken Peffers, “The design science research process: A model for producing and presenting information Systems research,” *J. Manag. Inf. Syst.*, pp. 1–24, [Online]. Available: h
[88] K. Peffers, “A Design Science Research Methodology for Information Systems Research,” no. August 2014, 2008, doi: 10.2753/MIS0742-1222240302.
[89] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007, doi: 10.2753/MIS0742-1222240302.
[90] “Pre-Processing in OCR!!!. A basic explanation of the most widely… | by Susmith Reddy | Towards Data Science.” https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7 (accessed Jul. 16, 2022).
[91] R. Chamchong *et al.*, “Comparing Binarisation Techniques for the Processing of Ancient Manuscripts To cite this version : Comparing Binarisation Techniques for the Processing,” 2014.
[92] J. T. Barron, “A Generalization of Otsu’s Method and Minimum Error Thresholding,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12350 LNCS, pp. 455–470, 2020, doi: 10.1007/978-3-030-58558-7_27.
[93] W. Bhimji, S. A. Farrell, T. Kurth, M. Paganini, Prabhat, and E. Racah, “Deep Neural Networks for Physics Analysis on low-level whole-detector data at the LHC,” *J. Phys. Conf. Ser.*, vol. 1085, no. 4, 2018, doi: 10.1088/1742-6596/1085/4/042034.
[94] “What is Feature Extraction? Feature Extraction in Image Processing | Great Learning.” https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/ (accessed May 25, 2022).
[95] “Splitting Machine Learning Data: Train, Validation, Test Set Split.” https://www.v7labs.com/blog/train-validation-test-set (accessed May 25, 2022).
[96] “Keras.Conv2D Class - GeeksforGeeks.” https://www.geeksforgeeks.org/keras-conv2d-class/ (accessed Jun. 06, 2022).

Appendix

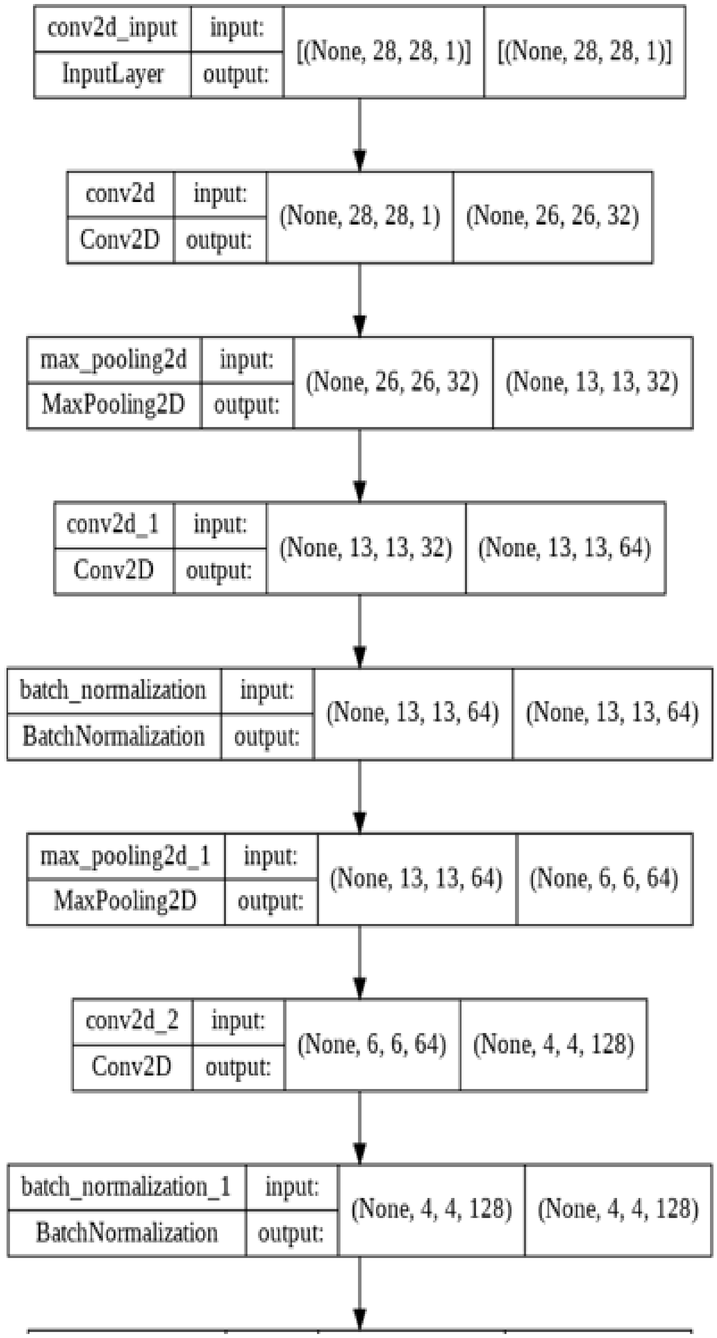


The first model by Convolutional Neural network

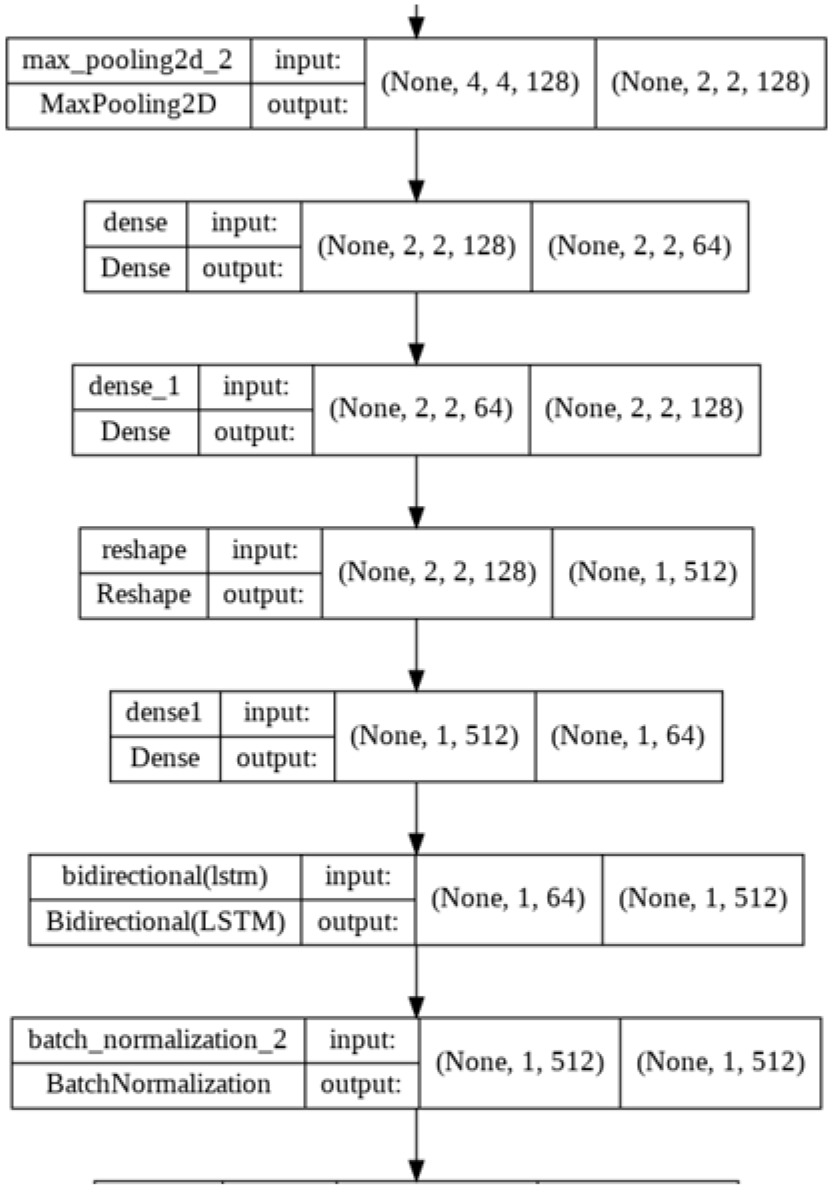


The Second Model that the researcher builds using CNN and BiLSTM

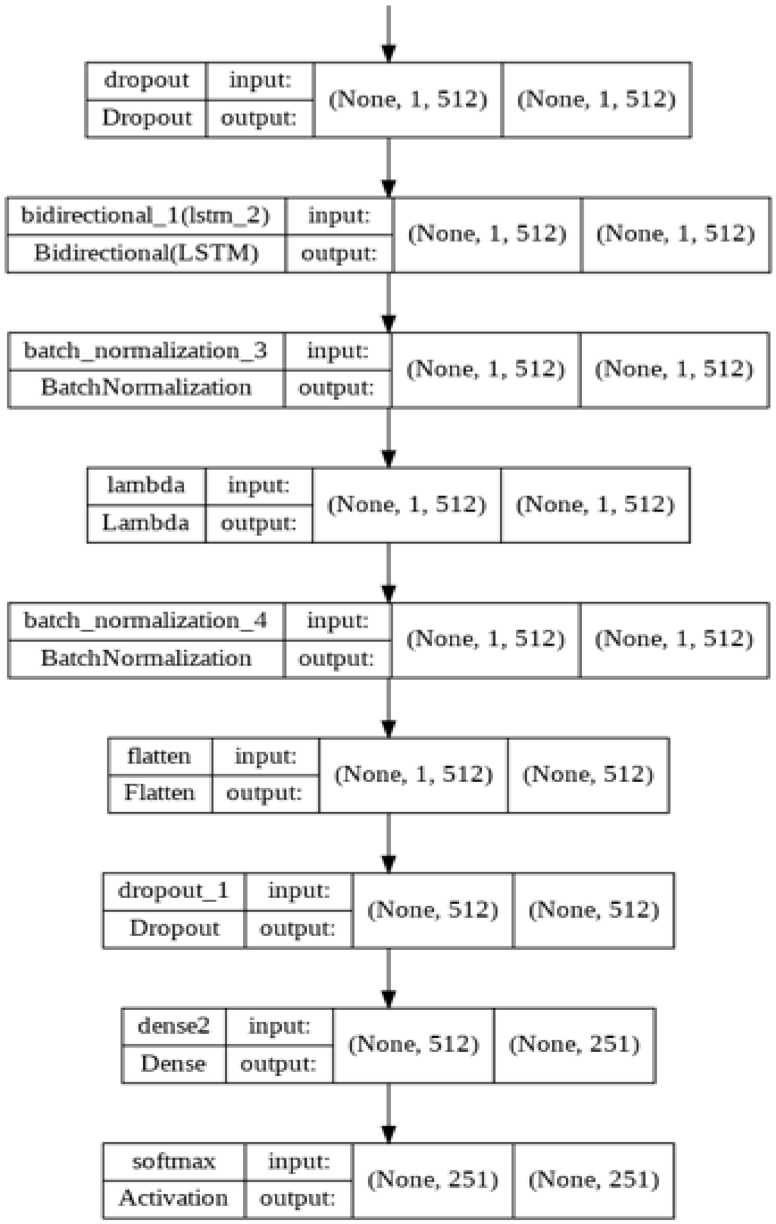
The third Model that the researcher builds using CNN and BiLSTM as Second model (CNN-BLSTM(II)), as follow with the following 3 visualization portions.



Part 1


max_pooling2d_2
MaxPooling2D
input:
output:
(None, 4, 4, 128)
(None, 2, 2, 128)
dense
Dense
input:
output:
(None, 2, 2, 128)
(None, 2, 2, 64)
dense_1
Dense
input:
output:
(None, 2, 2, 64)
(None, 2, 2, 128)
reshape
Reshape
input:
output:
(None, 2, 2, 128)
(None, 1, 512)
dense1
Dense
input:
output:
(None, 1, 512)
(None, 1, 64)
bidirectional(lstm)
Bidirectional(LSTM)
input:
output:
(None, 1, 64)
(None, 1, 512)
batch_normalization_2
BatchNormalization
input:
output:
(None, 1, 512)
(None, 1, 512)

Part 2


dropout
Dropout
input:
output:
(None, 1, 512)
(None, 1, 512)
bidirectional_1(lstm_2)
Bidirectional(LSTM)
input:
output:
(None, 1, 512)
(None, 1, 512)
batch_normalization_3
BatchNormalization
input:
output:
(None, 1, 512)
(None, 1, 512)
lambda
Lambda
input:
output:
(None, 1, 512)
(None, 1, 512)
batch_normalization_4
BatchNormalization
input:
output:
(None, 1, 512)
(None, 1, 512)
flatten
Flatten
input:
output:
(None, 1, 512)
(None, 512)
dropout_1
Dropout
input:
output:
(None, 512)
(None, 512)
dense2
Dense
input:
output:
(None, 512)
(None, 251)
softmax
Activation
input:
output:
(None, 251)
(None, 251)

Part 3

Manual Evaluation Sample

ለሙከራ የቀረቡ የእጅ ጽሑፍ ምስሎች መለያ ኮድ	የእውቅና ደረጃ				
	(0) ምንም እውቅና የለውም	(1) ብዙም አልታወቀም	(2) በከፊል አልታወቅም	(3) ታውቋል ማለት ይቻላል	(4) በትክክል እውቅና አግኝቷል
MI1				✓	
MI2				✓	
MI3			✓		
MI4			✓		
MI5			✓		
MI6			✓		
MI7				✓	
MI8			✓		
MI9				✓	
MI10				✓	
MI11				✓	
MI12				✓	

ጥልቅ ትምህርትን በመጠቀም ስለተሰራው የእጅ ጽሑፍ ማወቂያ ስርዓት ምን አይነት አስተያየት አለዎት፡

[illegible]
[illegible]
[illegible]

አዲስ ስልተሰራው የማሽን ሞዴል ምን ቢስተካከል ብለው ያስባሉ፡

[illegible]
[illegible]
[illegible]

ለሙከራ የቀረቡ የእጅ ጽሑፍ ምስሎች መለያ ኮድ	የእውቅና ደረጃ				
	(0) ምንም እውቅና የለውም	(1) ብዙም አልታወቀም	(2) በከፊል አልታወቅም	(3) ታውቋል ማለት ይቻላል	(4) በትክክል እውቅና አግኝቷል
MI1				✓	
MI2				✓	
MI3			✓		
MI4			✓		
MI5				✓	
MI6			✓		
MI7				✓	
MI8			✓		
MI9				✓	
MI10				✓	
MI11				✓	
MI12				✓	

ጥልቅ ትምህርትን በመጠቀም ስለተሰራው የእጅ ጽሑፍ ማወቂያ ስርዓት ምን አይነት አስተያየት አለዎት፡

የተዘጋጀው የእጅ ጽሑፍ ፊደሎችን የማወቂያ ሲስተም በጣም ጥሩ እና አስፈላጊ ነው:: [illegible] ሰው የቤቱ እና የመረጃት [illegible] ላይ [illegible] ለመቀየር [illegible] በጣም አስፈላጊ ስራ ነው::

አዲስ ስለተሰራው የማሽን ሞዴል ምን ቢስተካከል ብለው ያስባሉ፡

የተዘጋጀው የእጅ ጽሑፍ ማወቂያ ሲስተም ለእጅ የተቀረፁ ፊደሎችን [illegible] እንዲሆን በስክሪን መውጣት ነው