MEKELLE UNIVERSITY
Ethiopian Institute of Technology - Mekelle
School of Computing
Department of Computer Science

A Machine Learning Framework for Amharic Sentiment Analysis in Social Media Images Using OCR and NLP Techniques

By
Halefom Desta Fitsum

A Thesis
Submitted in Partial Fulfillment of the Requirements for The Master of Science Degree in Computer Science

Advisor: Shishay Welay (PhD)

Nov 2025
Mekelle, Ethiopia

MEKELLE UNIVERSITY
Ethiopian Institute of Technology-Mekelle
School of Computing
Computer Science Department

CERTIFICATION

This is to certify that the thesis entitled "A Machine Learning Framework for Amharic Sentiment Analysis in Social Media Images Using OCR and NLP Techniques" is submitted in partial fulfillment of the requirements for the degree of Masters in Computer Science and has been carried out by **Halefom Desta Fitsum**, **ID No: Eitm/pr175991/12,** under my supervision. Therefore, I recommend that the student has fulfilled the requirements and hence can submit the thesis to the Department.

_____	_____	_____
Name of Major Advisor	Signature	Date
_____	_____	_____
Name of Co-Advisor	Signature	Date

DECLARATION

I hereby declare that this Masters thesis is my original work and has not been presented for a degree in any other university and all sources of material used for this thesis have been duly acknowledged.

Name: _____

Signature: _____

Date: _____

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

Name: _____

Signature: _____

Date: _____

EXAMINERS' APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defense by **Halefom Desta Fitsum**, have read and evaluated his thesis **"A Machine Learning Framework for Amharic Sentiment Analysis in Social Media Images Using OCR and NLP Techniques"** and evaluated the candidate. This is therefore to certify that the thesis has been accepted in partial fulfillment of the requirements for the Masters Degree in Computer Science.

Name of Chairperson	Signature	Date
Name of Major Advisor	Signature	Date
Name of Internal Examiner	Signature	Date
Name of Major Advisor	Signature	Date
Name of External Examiner	Signature	Date

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the candidate's Department through the office of the Department Graduate Program Coordinator. Thesis Approved by:

Graduate Program Coordinator	Signature	Date

Certification of the Final Thesis

I hereby certify that all the corrections and recommendations suggested by the Board of Examiners are incorporated into the final thesis entitled “A Machine Learning Framework for Amharic Sentiment Analysis in Social Media Images Using OCR and NLP Techniques” by Halefom Desta Fitsum.

_____ _____ _____

Department Head Signature Date

Stamp of the Department of _____

Acknowledgements

Above all, I am deeply grateful to my advisor, Shishay Welay (PhD), for his invaluable guidance, constructive feedback, and constant encouragement throughout the course of this thesis. His support has been essential in shaping the direction and quality of my work. I would also like to extend my heartfelt thanks to my family and friends for their unwavering support and assistance whenever I needed it.

Abstract

In Ethiopia, social media platforms are increasingly used as spaces for public communication, with much of the opinion-rich content embedded in images containing Amharic text. Conventional sentiment analysis methods are designed for plain text and they fail to capture this significant portion of online discourse. Complexity of the Amharic script, scarcity of language processing tools, and limitations in computational resources further restrict automatic analysis of image-based text. So, this study develops an integrated framework that combines Optical Character Recognition (OCR) and Natural Language Processing (NLP) techniques to extract and classify Amharic text from social media images into Positive, Negative, and Neutral categories using machine learning classifiers. A balanced dataset of 600 annotated images was compiled and preprocessed with OpenCV for image enhancement and Tesseract OCR for text extraction. The extracted texts underwent different text preprocessing stages, including normalization, character unification, and stopword removal. Then the preprocessed texts are vectorized using Term Frequency–Inverse Document Frequency (TF-IDF). Four machine learning classifiers Support Vector Machine, Logistic Regression, Naive Bayes, and Random Forest were implemented, and the performance of each classifier were evaluated by different evaluation metrics such as, accuracy, precision, recall, F1-score and confusion matrices. The results from the evaluation metrics showed that SVM achieved the highest accuracy of 86%, Logistic Regression (83%) and Naive Bayes (82%), while Random Forest performed less by achieving 75%. These findings highlights that linear classifiers are suitable for Amharic sentiment analysis under resource-constrained conditions. The study demonstrates the feasibility of integrating OCR and NLP techniques for sentiment analysis of Amharic social media images and provides a solid baseline for future work in morphologically rich language processing.

Keywords:

Amharic, sentiment analysis, social media images, OCR, NLP, TF-IDF, machine learning, SVM.

List of Abbreviations and Acronyms

CSV	Comma-Separated Values
F1	F1-Score (Harmonic Mean of Precision and Recall)
ML	Machine Learning
NLP	Natural Language Processing
OCR	Optical Character Recognition
OpenCV	Open Source Computer Vision Library
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency
LR	Logistic Regression
NB	Naive Bayes

Table of Contents

List of Tables

List of Figures

Chapter One

1. Introduction

1.1 Background of Study

Recently social media platforms have become powerful tools for communication, self-expression, and sharing information especially in developing countries like Ethiopia. Millions of users actively share posts, memes and engage by giving comments that often contain embedded text within images rather than just typed texts. This is particularly common for content created in local languages such as Amharic, where text is frequently stylized in cultural, religious, or political memes, screenshots, digital flyers, and other visual media [1].

Amharic, the Ethiopian official working language is widely used online, but it remains a low-resource language in OCR and NLP research, with limited datasets, tools, and prior studies available [1,2,4,7]. OCR systems, particularly open-source tools such as Tesseract, have achieved high accuracy for Latin-script languages. However, their performance on non-Latin scripts such as Geʿez (used in Amharic) remains inconsistent due to challenges such as limited annotated datasets and insufficient font support [2]. As a result, extracting accurate Amharic text from images remains a technical challenge.

On the other hand, sentiment analysis, the process of determining whether a piece of text expresses a positive, negative, or neutral emotion has gained substantial attention in recent years, especially in areas such as social media monitoring, political discourse analysis, and public opinion mining [3]. While English and other high-resource languages enjoy advanced sentiment analysis models powered by deep learning and large pre-trained language models, Amharic suffers from resource constraints such as lack of labeled corpora, preprocessing tools, and robust morphological analyzers [13].

In the Ethiopian context social media has become not just a tool for interaction but a major platform for political debate, activism, and crisis communication, especially during periods

of conflict, tragic events and national emergency. During such times, a significant portion of user-generated content takes the form of emotional appeals, criticism or support shared through image posts. Unfortunately these texts are often overlooked by existing sentiment analysis tools that depend on typed comments or posts. This limitation creates a gap in digital monitoring and analysis systems, especially for government agencies, media researchers, or civil society organizations seeking to understand public opinion dynamics [4].

From a technological perspective combining OCR with NLP techniques offers an innovative yet method in low-resource language settings. When carefully trained and optimized, OCR systems can serve as gateway for extracting valuable textual data hidden inside visual content. Once extracted, this text can be analyzed by using lightweight machine learning algorithms such as Logistic Regression or Support Vector Machines (SVM) which are suitable for environments with limited computing power and small datasets. Recent advancements in open-source tools and the availability of benchmark datasets for Amharic scene text recognition such as those proposed in [14], provides an opportunity to build efficient sentiment classification pipelines for Amharic without requiring massive computational infrastructure.

Despite its challenges the integration of OCR and NLP techniques presents a promising pathway for capturing and analyzing public sentiment expressed through image-based content. Few existing systems have attempted this kind of pipeline in low-resource settings, and none have provided a robust solution for Amharic image-based sentiment analysis. The purpose of this study is to bridge that gap by developing a system that combines optimized Amharic OCR with machine learning-based sentiment classification that is specifically tailored for social media images.

This approach is expected to support applications such as digital opinion tracking, online hate speech monitoring, and even public mood analysis in different events like elections or national crises. The proposed system will contribute both to the development of language technologies for underrepresented languages and to the broader field of social media analytics by automating analysis of Amharic content in visual form.

1.2 Motivation

In the digital information age one of the most pressing challenges is extracting meaningful insights from visual content shared on social media platforms. In Ethiopia, where Amharic language is widely used, much of the opinionated and emotionally charged content is embedded within images rather than plain text. This presents challenge for computational analysis, as traditional sentiment analysis models are mainly built to process plain, typed text and cannot directly analyze textual content embedded in images. As a result significant amount of user generated sentiment remains untapped, limiting the effectiveness of digital opinion mining.

The current manual approach for analyzing such visual content involves human inspection, which is time-consuming, subjective, error-prone, and unscalable given the vast volume of image-based content generated on social media platforms like Facebook, Instagram, LinkedIn and X. This has motivated the development of an automated system capable of extracting Amharic text from images using Optical Character Recognition (OCR) and analyzing the sentiment using Natural Language Processing (NLP) techniques. By automating this process, the system can enable faster, more consistent, and scalable analysis of public opinion embedded in images.

Despite the fact that there is great advancement in OCR and NLP techniques, most of the progress has been made in high-resource languages like English and Chinese. Most languages such as Amharic still face challenges like limited labeled datasets, weak OCR support for complex scripts, and underdeveloped sentiment models. The existing systems often demand high computational resources. This makes them impractical for real-world applications in developing countries. These limitations have driven a need for lightweight, modular and efficient sentiment analysis pipeline for Amharic language and capable of operating under constrained computing environments.

By combining lightweight OCR using Tesseract and NLP Techniques with traditional machine learning algorithms such as Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest for sentiment classification, this study addresses those

limitations. The system is designed to perform well even with small datasets and limited computational power. This research fills the gap in Amharic sentiment analysis by targeting image-based text that are largely overlooked source of data. it also provides a cost-effective and easily deployable solution for extracting and interpreting public sentiment from social media images, offering substantial benefits to researchers, policymakers, and digital communication analysts.

This study advances Amharic language technology by testing open-source OCR and NLP tools in low-resource settings. It also faced challenges, including poor OCR accuracy on low-quality or stylized text and the difficulty of analyzing informal social media language with inconsistent spelling and structure.

By overcoming these difficulties this research presents reliable and efficient system for sentiment analysis from Amharic image-based content. It demonstrates that with smart integration of open-source technologies and careful system design, it is possible to make meaningful progress in sentiment mining for low-resource languages, particularly in image-rich, digitally active environments like modern social media.

1.3 Problem Statement

In the digital era, social media platforms have become dominant medium for people to express opinions, emotions and sentiments. A significant portion of user generated content, especially in Ethiopian context is embedded in images rather than plain text. These images often contain Amharic text which is a language with a complex script and limited digital resources, making the automatic extraction and sentiment analysis of such data extremely challenging. Traditional sentiment analysis models fundamentally rely on textual input and overlook valuable sentiment rich visual content, thereby failing to capture the full scope of user opinions [1,2,4].

Moreover, the limited research on Amharic scene text recognition has left technological gap in developing robust Optical Character Recognition (OCR) system capable of accurately extracting Amharic text from low-quality, noisy, or stylistically varied social

media images [14,17]. Existing OCR systems for Latin or other widely used scripts often fail to perform effectively on Ge'ez-based languages like Amharic because of script complexity and dataset scarcity [18,19]. Similarly, most of the sentiment analysis studies for Amharic are based on small, manually collected datasets and traditional machine learning methods that struggle with linguistic nuances, morphology, and domain-specific slang [7,8,11].

Consequently, there is need for end-to-end framework that combines Amharic OCR and Natural Language Processing (NLP) techniques to extract and analyze sentiment from image-based content on social media. This study aims to fill this gap by integrating Tesseract-based OCR and tailored NLP techniques for Amharic, allowing for the automated extraction and classification of sentiments from social media images. So, by addressing the issues of script recognition and sentiment understanding, the proposed system provides a practical solution for sentiment monitoring in Amharic language.

1.4 Research questions

The research questions that are addressed in this study are:

1. How effectively can Optical Character Recognition (OCR) extract Amharic text from social media images with diverse fonts, layouts, and image quality?
2. What natural language processing techniques are suitable for performing sentiment analysis on noisy or informal Amharic text extracted from images?
3. How does the performance of the integrated OCR and NLP system compare when using different classification models based on the same TF-IDF feature representation?
4. What metrics are appropriate to evaluate the accuracy and reliability of Amharic sentiment classification from image-based text?

1.5 Objectives

1.5.1 General objective

The general objective of this study is to develop a machine learning framework for Amharic sentiment analysis in social media images using OCR and NLP techniques.

1.5.2 Specific objectives

To achieve the general objective of the study, this study includes the following specific objectives:

1. To review existing literatures related to Amharic OCR and sentiment analysis, highlighting the strengths and limitations of key approaches.
2. To collect dataset of Amharic social media images containing embedded textual content.
3. To develop a preprocessing pipeline that enhances image quality and prepares data for OCR processing.
4. To test the capability and accuracy of Tesseract OCR (with image preprocessing) in extracting Amharic text from social media images.
5. To preprocess the extracted Amharic text through normalization, tokenization, and other NLP techniques suitable for sentiment analysis.
6. To train and evaluate sentiment classification models using machine learning techniques on the extracted and preprocessed text.
7. To analyze the performance of the proposed system using standard evaluation metrics such as accuracy, precision, recall, and F1-score.
8. To compare the performance of different sentiment classification approaches and determine the most effective method for Amharic text in image-based social media content.

1.6 Scope and Limitation

1.6.1. Scope

The scope of this study is limited to developing an integrated system that applies Optical Character Recognition (OCR) to extract Amharic text from social media images and utilizes Natural Language Processing (NLP) techniques to classify the extracted text into sentiment categories. The OCR component employs the Tesseract engine supported by OpenCV-based preprocessing to enhance recognition accuracy for printed Amharic text. The extracted text is transformed using TF-IDF and classified into positive, negative, and neutral sentiment using traditional machine learning algorithms, including Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest. Model performance is evaluated using accuracy, precision, recall, and F1-score to determine the most effective classification approach. The study excludes handwritten text, video content, and audio-based content, focusing solely on static social media images containing printed Amharic text.

1.6.2. Limitation

This study has the following limitations. The sentiment classification model is restricted to three sentiment categories: positive, negative, and neutral, excluding complex emotional expressions like sarcasm or mixed sentiments. The OCR component may result in low accuracy when it processes images with noisy background, low resolution, or poor quality. The study relies on traditional machine learning approaches rather than deep learning approaches for sentiment classification due to limited computational resources and dataset constraints. The current system is primarily trained and optimized for the Amharic language. Since Amharic and Tigrigna share the Ge'ez (fidel) script and contain overlapping vocabulary, the system may incorrectly process Tigrigna text as Amharic. The study does not currently include a dedicated Language Identification (LID) module to distinguish between these two languages prior to sentiment classification.

1.6.3. Constraints Affecting Scope, Validity and Generalizability

The scope, validity, and generalizability of this study are all impacted by a number of constraints. Dataset limitations arise because the study utilizes a relatively small, manually labelled dataset of Amharic social media images, which may not fully represent all dialectal variations or sentiment expressions, potentially affecting how the model applies to more extensive real-world scenarios. A further constraint is image variability, since variations in lighting, noisy backgrounds, and image quality can lower OCR accuracy and have an indirect impact on sentiment classification results. In Addition, language processing challenges emerge from the scarcity of advanced Amharic NLP tools such as tokenizers, stemmers, and stopword lists, which may reduce the accuracy of text preprocessing and affect sentiment classification. Limited computing capacity constrained the use of deep learning models, restricting experimentation to lightweight traditional machine learning algorithms, which may underperform on larger datasets. These constraints outline the contextual and technical boundaries within which this research was conducted.

1.7 Significance of study

This study contributes to the advancement of sentiment analysis in Amharic text embedded in social media images. As online platforms increasingly become the space for public expression in Ethiopia, much of the sentiment-rich content is shared in the form of images containing Amharic script. Even though little effort is made to extract and analyze such textual data, this study bridges the gap by building an integrated system that uses OCR to extract text from images and NLP techniques to analyze sentiments. The outcome of this study can support digital media monitoring, public opinion analysis, and sociopolitical insight generation, particularly in local contexts where Amharic is mainly used. It also provides a foundation for future enhancements in Amharic text recognition and sentiment classification using more advanced models.

1.8 Organization of thesis

The thesis is structured as follows: Chapter One provides an overview of the study's background, problem statement, objective, research questions, scope, limitations, and significance of the study. Chapter Two reviews several existing research studies focusing on the strengths and limitations of each approach. Chapter Three outlines the methodologies employed in the study, including the tools, techniques and processes used for system development. Chapter Four discusses the experimental result of this study. Finally, Chapter five summarizes the main findings of this study, provides detailed answers to each research question, and focuses on our future implementation.

Chapter Two

2. Literature review

Nowadays the world wide web and social media platforms have become a source of information, public and individual opinion, emotional expression, and political discourse. The use of social media such as Facebook, Instagram, LinkedIn and X is growing rapidly in Ethiopia. This is leading to a massive increase in user-generated Amharic content. So, analyzing the sentiment behind those contents is very important to understand public mood, political attitudes, and social behavior. However, sentiment analysis in Amharic is challenging due to limited resources, the morphological complexity of the language, and the use of informal or code-mixed writing styles in online communication. This challenge becomes even more difficult when the Amharic text appears in the form of images, such as memes or screenshots, which require Optical Character Recognition (OCR) before natural language processing can be applied [6,17,19].

Sentiment analysis, also referred to as opinion mining, is the task of identifying and classifying the emotional tone behind a piece of text. In Amharic, most sentiment analysis studies have used either traditional machine learning or more recent deep learning techniques. Machine learning approaches are typically based on handcrafted features such as word frequency, term weighting (TF, TF-IDF), or n-grams, and classifiers such as Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and K-Nearest Neighbors (KNN). Machine learning models are simple to implement and require less computational power, but they depend mainly on feature engineering and may have difficulties to capture semantic or contextual meaning. Several researchers have applied these methods to analyze sentiment in various domains, including COVID-19-related Facebook comments [7], multi-scale sentiment classification [8], political discourse [9], restaurant reviews [11], and filtering offensive content [12]. Some studies also addressed challenges such as subjectivity detection, data imbalance, and the lack of annotated corpora by introducing ensemble models [10] and oversampling techniques like SMOTE.

In contrast, deep learning models have become more popular due to their ability to automatically extract features and learn hierarchical representations from raw data. Convolutional neural networks (CNN) recurrent neural networks (RNN), bidirectional long short-term memory (BiLSTM) and CNN-BiLSTM hybrid architectures have all been used to classify emotions and sentiment in Amharic [1,3,5]. To capture semantic relationships in the language, these models often employ embedding methods like Word2Vec or FastText [3,13]. Utilizing multilingual embeddings like LaBSE and sentence transformers for cross-lingual sentiment analysis has also been explored in some studies [2]. In addition, recent studies have combined OCR with deep learning to extract Amharic text from memes and image-based posts for hate speech detection and sentiment classification [6].

Even though both approaches achieved promising results, there are still limitations in the development of sentiment analysis systems for Amharic, including a lack of standardized tools, limited datasets, spelling variations, and difficulties handling sarcasm, figurative speech, and code-mixed expressions [3,4]. Nevertheless, ongoing research has demonstrated that with proper preprocessing, embedding strategies and model tuning, both machine learning and deep learning techniques can provide valuable insight into public opinion expressed in Amharic.

This chapter presents comprehensive review of existing literatures related to Amharic sentiment analysis. The studies are categorized into two main sections. Section 2.1 reviews deep learning approaches and Section 2.2 focuses on traditional machine learning methods.

2.1. Deep Learning Approaches

Amharic Political Sentiment Analysis Using Deep Learning Approaches [1].

This study employed four deep learning algorithms; Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) as well as employed the CNN -BiLSTM model to detect the sentiment expressed on the Amharic text on social media. The experimental results indicated that the hybrid CNN-BiLSTM is superior to other models and the proposed method reaches an accuracy of 91.60%. Unlike conventional ML algorithms which depend on manual labor intensive feature engineering, this method automatically learned semantic and context information in Amharic. The research emphasized the significance of dataset size,

sentence length, word embedding, and hyperparameter selection in the model performance. Despite the challenges, like distinguishing sarcasm, figurative speech and limited data, the study demonstrated the potential of deep learning in processing morphologically rich languages like Amharic. It also recommended future work in multilingual, multi-class, and sarcasm-aware sentiment classification systems.

Deep Learning-based Sentiment Classification in Amharic using Multi-lingual Datasets [2].

This research explored Amharic sentiment classification using both monolingual and cross-lingual approaches by using deep learning techniques. These techniques are: CNN, LSTM, BiLSTM, FFNN, and sentence transformers. As there is limited availability of Amharic datasets, the researchers multiplied the data using machine translation and also tested both machine-translated and the original datasets. By combining sentence transformers with FFNN and cosine similarity a good performance was achieved, providing 82.2% accuracy for 2-class and 62.0% for 3-class classification. The study discovered a small performance difference between monolingual and cross-lingual setups. This indicates that cross-lingual models like LaBSE can capture Amharic semantics very well. Even though it is lower in accuracy than English models, the results show significant progress for sentiment analysis in languages like Amharic. The research suggests that high-quality machine translation and transformer-based embedding can help to overcome resource limitations in Amharic NLP.

Deep Learning Based Emotion Detection Model for Amharic Text [3].

This study proposed a deep learning-based model to detect emotions in Amharic text using Convolutional Neural Networks (CNN) with Word2Vec word embedding techniques. The model was designed to categorize social media comments into four emotional categories: these are sadness, anger, disgust, and happiness. It achieved 71.11% accuracy for the four emotion classes (sadness, anger, disgust, and happiness) detection and 87.46% accuracy for binary classification (positive and negative). The

research highlighted the importance of preprocessing, word vector quality, and model tuning to achieve robust emotion classification for morphologically rich languages like Amharic. When compared to RNN, CNN outperformed it in this context. The study highlighted challenges such as the lack of efficient morphological analyzers, symbol redundancy, and the need for improved preprocessing methods to handle inconsistent language patterns in Amharic social media content, while demonstrating the growing importance of emotion mining for decision-making in Ethiopian social and political contexts.

Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models [4].

By developing both classification models and annotation tools, this study presents a comprehensive approach for Amharic sentiment analysis. Despite Ethiopia's limited bandwidth, a Telegram-based annotation system called ASAB was introduced, allowing many users to label 9.4k Amharic tweets. The research tested classical supervised and deep learning models, revealing that network embedding-based deep learning models, particularly Role2Vec, outperformed other approaches. The study emphasized the difficulty of handling sarcasm, figurative speech, mixed scripts, and informal grammar in social media content. In addition, it found that removing the mixed sentiment class improved overall model accuracy. ASAB proved successful in enabling scalable data collection in low-resource environments, and the authors released all datasets, source code, and tools publicly to support future research.

Sentiment Analysis for Amharic-English Code-Mixed Sociopolitical Posts Using Deep Learning [5].

This study developed deep learning based system for classifying sociopolitical posts into positive, negative, and neutral sentiments in response to the increasing amount of Amharic-English code-mixed content on social media. The authors curated an 8,819 instance dataset from Twitter and Facebook and annotated for multiclass classification. To handle linguistic complexity, the study incorporated language detection, code-

switching handling, and Amharic text normalization during preprocessing. Four deep learning architectures (CNN, LSTM, BiLSTM, and a CNN-BiLSTM hybrid) were trained using both TF-IDF and Count Vector representations with n-gram ranges. Results indicated that, with performance reaching 84.46% accuracy using Count Vectorizer and preprocessing that included code-switching, CNN consistently outperformed others across multiple experiments. While the hybrid CNN-BiLSTM also performed well, CNN stood out for its stability and simplicity. The authors underlined the impact of code-switch-aware preprocessing on model accuracy and suggested more research into vectorization methods for code-mixed texts.

Identification of Hateful Amharic Language Memes on Facebook using Deep Learning Algorithms [6].

This study used deep learning and Amharic OCR technology to address the growing spread of hate speech in Amharic Facebook memes. A dataset of 5,000 annotated image-based posts was constructed, extracting textual content using OCR and preprocessing it through normalization and tokenization tailored to Amharic morphology. Several deep learning models, including BiLSTM, BiGRU, and hybrid variants, were compared using both traditional (BoW, TF-IDF) and advanced (Word2Vec, FastText) embeddings. With an accuracy of 94%, the BiLSTM and Dense model outperformed both classical machine learning baselines and standalone deep models. The study highlighted how embedding layers and dense connections mitigated overfitting, which was a challenge for standard BiLSTM setups. This research offers a novel direction by combining image-based text extraction with sequential deep learning, emphasizing the potential of OCR-integrated NLP approaches for Amharic hate speech detection.

2.2. Machine Learning Approaches

Sentiment Analysis on Amharic Language-Based COVID-19 Discourse from Facebook social media comments [7].

This study explored sentiment analysis of Amharic-language Facebook comments related to COVID-19 using traditional supervised machine learning techniques. The author focused on addressing the limited exploration of sentiment in Ethiopian social

media discourse during the pandemic. After preprocessing, 7,309 of 15,000 comments that were collected were kept and classified using Naive Bayes, SVM, and Maximum Entropy algorithms. The study compared multiple feature extraction methods, including Bag of Words, TF-IDF, and Word2Vec, and found that Naive Bayes with TF-IDF yielded the highest accuracy (83.3%). The work also emphasized the challenges of Amharic morphology, unnormalized characters, and spelling variations, which complicate text classification. Unlike many previous studies that rely on English-language data, this research contributed to the underexplored domain of Amharic sentiment analysis by developing a custom dataset and evaluating model performance specifically on COVID-19 discourse. The study highlighted how important morphological processing and text normalization are to improving classifier performance in morphologically rich languages like Amharic. Moreover, it underscored the relevance of local-language sentiment analysis for public health communication and crisis response.

A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts [8].

A supervised machine learning framework for sentiment analysis of Amharic texts is presented in this study to capture not just polarity but also intensity through a multi-scale classification scheme. A manually annotated dataset of 608 social media posts was developed and categorized into five sentiment classes: strongly negative, negative, neutral, positive, and strongly positive. The authors employed Naive Bayes as classification algorithm and compared unigram, bigram, and hybrid n-gram features. Their findings showed that the bigram model outperformed the others in accuracy, particularly for extreme sentiment classes, due to its ability to capture valence shifters like intensifiers and diminishers. One of the work's main contributions is the lemmatization of Amharic text using a morphological analyzer (HornMorpho), which helps deal with sparsity from morphological complexity. Although limited by a small dataset and the lack of NLP tools for Amharic, the model achieved promising results and set the groundwork for future multi-scale sentiment classification in Semitic languages.

Subjectivity and Sentiment Analysis of Amharic Comments on Social Media: The Case of Ethiopia Political Discourse [9].

This study investigates how subjectivity and sentiment are expressed in Amharic political discussions on Facebook. A manually gathered dataset containing 5,250 Amharic comments was annotated into four sentiment classes: strongly positive, negative, strongly negative, and neutral. The approach used a traditional machine learning pipeline, with preprocessing included stemming, tokenization, and stop-word removal. Decision Tree, K-Nearest Neighbors and Naive Bayes classifiers were used in the study; Naive Bayes outperformed others, with an accuracy of 80.7%. The work also attempted to capture the distinction between subjective and objective comments, which is often overlooked in Amharic sentiment research. A key contribution of this thesis is its domain-specific focus on political discourse, which highlights the significance of tailoring sentiment models to context-specific expressions in morphologically complex languages like Amharic. The study contributes to the limited amount of research on local-language political sentiment on social media platforms, though constrained by the absence of publicly accessible datasets and standardized tools.

Meta-Learner for Amharic Sentiment Classification [10].

The study investigates sentiment classification for Amharic texts using a stacked ensemble learning framework. Recognizing the challenges posed by Amharic as a low-resource language, especially the lack of labeled corpora and linguistic tools, it propose a meta-learning approach to improve classification performance. It integrates multiple base learners (SVM, Random Forest, and Naive Bayes) and uses Logistic Regression as a meta-learner. It adopt TF-IDF character n-gram features (1–7 range) combined with SMOTE to handle class imbalance by achieving 90% accuracy. Ensemble classifier consistently outperforms its individual components across different evaluation metrics. According to a significant finding of the study, character-level features particularly when paired with oversampling, are more effective in capturing morphological patterns in Amharic than word-level features. This research contributes

to growing effort of adapting robust machine learning frameworks to low resource languages like Amharic by demonstrating how meta-learners can mitigate performance issues tied to data scarcity and feature representation.

Design Amharic Text Sentiment Analysis Model Using Machine Learning Techniques: In Case of Restaurant Reviews [11].

This study presents a sentiment classification model tailored for Amharic restaurant reviews using supervised machine learning techniques. The researchers constructed a dataset of 1,068 reviews and experimented with three classifiers: Naive Bayes, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN). Using both unigram and bigram features, it evaluated each model using several feature selection schemes, such as TF, TF-IDF, binary term occurrence and n-grams. SVM outperformed other classifiers, achieving up to 80.43% accuracy with bigram features and term frequency. In addition, Naive Bayes and KNN also delivered competitive results, particularly with TF-IDF. The study underlined the importance of preprocessing, such as tokenization and normalization, in enhancing performance. Distinguishing between subjective and objective expressions was challenge particularly in reviews where sentiment was subtly embedded. This work stands out for its domain-specific application and thorough comparative evaluation of multiple ML techniques under varied feature engineering settings.

Information filtering of social media Amharic Texts Based on Sentiment Analysis [12].

The main goal of this study was to address the increasing issue of offensive and toxic content being shared on social media platforms using Amharic. It uses a machine learning-based information filtering system that classifies Facebook posts into categories such as politically offensive, religiously offensive, socially offensive and non-offensive. The study used word2vec for feature representation and variety of supervised learning approaches, including Naive Bayes, Support Vector Machines (SVM), Logistic Regression, and Decision Trees, on Amharic textual data. Unlike conventional sentiment analysis studies that aim to capture user opinions at the polarity level (positive, negative, or neutral), this work extended sentiment classification to

support content moderation goals. The system was designed to help in filtering out harmful or inflammatory posts that could potentially incite conflict or hate, aligning with the broader objective of using NLP tools to mitigate the misuse of digital platforms in Ethiopia. According to the evaluation results, SVM with word2vec features performed the best, with an average precision of 72% and a recall of 63.4%.

The study shows how sentiment analysis may be used practically in information governance and policy enforcement, especially in the sociopolitical environment of Ethiopia, where tensions can be quickly escalated by digital content. It supports the need for tailored machine learning solutions for under-resourced languages and also clarifies the unique challenges associated with processing Amharic text, such as morphological complexity and a lack of standardized resources.

Table 1: Literature Review Summary

No	**Title**	**Algorithms used**	**Authors**	**Critics**	**Efficiency**
1	Amharic Political Sentiment Analysis Using Deep Learning Approaches	CNN, BiLSTM, GRU, CNN-BiLSTM	Alemayehu, Fikirte & Meshesha, Million & Abate, Jemal	Challenges in detecting sarcasm and limited dataset size.	CNN-BiLSTM achieved 91.6% accuracy.
2	Deep Learning-based Sentiment Classification in Amharic using Multi-lingual Datasets	Sentence Transformers, FFNN, Cosine Similarity	Gebremichael Tesfagergish, Senait & Damaševičius, Robertas & Kapočiūtė-Dzikienė, Jurgita	Performance gaps in multilingual vs monolingual settings.	82.2% accuracy (binary); 62.0% (3-class)

3	Deep Learning Based Emotion Detection Model for Amharic Text	CNN, Word2Vec	Eyob Tesfu	Preprocessing and redundancy in symbols were challenges.	87.46% (binary), 71.11% (multi-class).
4	Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models	Role2Vec, Supervised DL Models	Seid Muhie Yimam, Hizkiel Mitiku Alemayehu, Abinew Ayele, and Chris Biemann	Handling mixed scripts and figurative speech was difficult.	Role2Vec showed superior performance.
5	Sentiment Analysis for Amharic-English Code-Mixed Sociopolitical Posts Using Deep Learning	CNN, LSTM, BiLSTM, CNN-BiLSTM	Ebabu, Yitayew & Chalie, Minalu	Complex code-switching handling required.	CNN achieved 84.46% accuracy with Count Vector.
6	Identification of Hateful Amharic Language Memes on Facebook using Deep Learning Algorithms	BiLSTM, BiGRU, Word2Vec, FastText, OCR	Belete, Mequanent & Kassa, Girma	Integration of OCR introduced additional complexity.	BiLSTM with Dense achieved 94% accuracy.
7	Sentiment Analysis on Amharic Language-Based COVID-19 Discourse from	Naive Bayes, SVM, MaxEnt, TF-IDF, Word2Vec	Eyasu Tekle	Issues with unnormalized characters and morphology.	NB with TF-IDF achieved 83.3% accuracy.

	Facebook Social Media Comments				
8	A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts	Naive Bayes classifier + lemmatization using HornMorpho + unigram/bigram features	Philemon Wondwossen, and Wondwossen Mulugeta	Small dataset and limited NLP tools.	Bigram model yielded best accuracy among tested.
9	Subjectivity and Sentiment Analysis of Amharic Comments on Social Media	Naive Bayes, Decision Tree, KNN	GEBRIEL GIZATE MOLLA	Lack of standardized tools and public datasets.	Naive Bayes achieved 80.7% accuracy.
10	Meta-Learner for Amharic Sentiment Classification	SVM, RF, NB, Logistic Regression (Meta), SMOTE	Alemneh, Girma Neshir & Rauber, Andreas & Atnafu, Solomon	Still limited by low-resource constraints.	Meta-Learner achieved 90% accuracy.
11	Design Amharic Text Sentiment Analysis Model Using Machine Learning Techniques	SVM, Naive Bayes, KNN, TF-IDF, Bigrams	B. Gedif, A. Alemu, Y. Assefa and S. Nibret	Challenges distinguishing subjective/objective sentiment.	SVM achieved 80.43% accuracy.
12	Information Filtering of Social Media Amharic Texts Based on Sentiment Analysis	SVM, NB, LR, Decision Tree, Word2Vec	Hiwot Wonago Kululo	Insufficient Amharic NLP tools and data bias.	SVM achieved 72% precision, 63.4% recall.

2.3. Challenges and Research Gap Analysis

Our literature review reveals several research gaps and challenges in Amharic sentiment analysis. These gaps are discussed as follows: Our literature review reveals several research gaps and challenges in Amharic sentiment analysis. One of the primary gaps is the limited language resources, as the scarcity of extensive annotated Amharic datasets challenges most of the reviewed studies. This limitation affects both the training and evaluation stages, especially in deep learning-based approaches where data size strongly influences model performance. Another challenge is the insufficient handling of code-mixed and informal texts. Amharic social media content frequently appears in Amharic-English code-mixed form and informal structures, yet only a few studies discuss preprocessing techniques for such cases, and even those rely on rule-based normalization that may not generalize well across variations.

A further gap concerns the underexplored use of OCR for text appearing in images. Although sentiment-rich content increasingly appears in image formats, few studies integrate OCR with NLP, creating a gap in applying sentiment analysis to real-world, image-heavy social media environments. In addition, inadequate preprocessing tools for dealing with Amharic's morphological complexity remain a challenge. While morphological analyzers like HornMorpho have been employed in some studies, there is still a need for robust and scalable preprocessing pipelines that can handle script variations and spelling inconsistencies.

Another problem relates to evaluation metric limitations. Accuracy alone is insufficient for performance measurement, especially in unbalanced or multi-class settings, yet precision, recall, and F1-score are frequently disregarded or underreported, risking misrepresentation of real-world model effectiveness. Generalization and overfitting issues in deep learning models also persist; despite high training accuracy, several studies report performance drops during validation, particularly when using complex architectures or limited datasets. This raises concerns about model robustness when deployed.

In addition, although prior studies have shown that character and bigram n-grams can improve sentiment classification for Amharic by capturing morphological patterns, this study adopts unigram-based TF-IDF features to balance performance with dataset size and computational feasibility. Lastly, a notable gap is the lack of OCR-NLP integrated pipelines for sentiment analysis. While some studies apply OCR for hate speech detection, sentiment analysis from text embedded in images remains unexplored.

The purpose of this study is to fill these gaps by extracting and analyzing the sentiment of Amharic text embedded in social media images through an integrated OCR and NLP pipeline. This approach emphasizes the development of an end-to-end system tailored to the linguistic challenges of Amharic while examining traditional and deep learning models in a resource-constrained setting.

Chapter Three

3. Methodology

This study aims to develop an effective and accurate system for performing sentiment analysis of Amharic texts extracted from social media images. The proposed system integrates Optical Character Recognition (OCR) and Natural Language Processing (NLP) techniques with classical machine learning models to identify and classify sentiments into positive, negative, and neutral categories. This chapter presents the research methodology, including the software and hardware tools utilized and the step-by-step procedures followed to design, implement, and evaluate the system.

3.1 Type of Research

This research is categorized as both quantitative and experimental. It is quantitative because the study gathers numerical data from the classification of text extracted from images, which can then be statistically analyzed to determine performance measures such as accuracy, precision, recall, and F1-score. This quantitative approach enables an objective evaluation of how well the model classifies sentiment.

It is also experimental in nature. The study's experimental design applies identical feature extraction using TF-IDF to implement multiple machine learning algorithms, including Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest, on the same dataset. By systematically evaluating the performance of these algorithms, the study identifies the most suitable method for Amharic sentiment analysis and compares the models under the same preprocessing and training conditions.

3.2 Materials

3.2.1 Software tools

This research was implemented using the Python programming language, which is widely used in scientific computing because of its clear structure, object-oriented design, and extensive ecosystem of libraries. Python provides robust support for data analysis, image processing, natural language

processing, and machine learning model development, making it a suitable choice for this study [15].

Visual Studio Code (VS Code) was used as the primary integrated development environment (IDE) for writing, testing, and debugging the Python code. Its lightweight design, built-in debugging capabilities, and rich ecosystem of Python extensions made it an appropriate development environment for the project.

OpenCV served as the core library for computer vision and image preprocessing tasks. Its broad functionality allowed efficient implementation of operations such as grayscale conversion and thresholding (including Otsu's method), which were applied to enhance text regions within social media images and improve OCR performance [16].

Tesseract OCR, accessed through the Pytesseract Python wrapper, was employed for text extraction. As an open-source OCR engine with extensive multilingual support, it is capable of recognizing complex scripts such as Amharic [17–19]. Its compatibility with preprocessing techniques was essential for obtaining reliable text recognition from morphologically rich Amharic content.

Pandas was used for data manipulation and analysis, particularly for managing training and test splits, organizing OCR outputs, and storing results in CSV format. Its high-level data structures, such as DataFrames, supported efficient handling of the experimental workflow [20].

The machine learning components of the system were implemented using scikit-learn (sklearn), an open-source library that offers a standardized API for classification, regression, clustering, model validation, and feature engineering [21]. In this study, it was used for TF-IDF vectorization, model training, and performance evaluation.

Matplotlib was utilized to generate visual representations of model performance. It provided the plotting functionality needed to visualize evaluation metrics, including accuracy trends and confusion matrices, thereby supporting interpretation of classifier behavior [22].

Seaborn, built on top of Matplotlib, was used to create statistical visualizations such as confusion matrix heatmaps and performance comparison charts. Its high-level interface enabled clear visualization of accuracy, precision, recall, and F1-score comparisons across Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest models [23].

Together, these software tools enabled a complete workflow for image preprocessing, OCR implementation, text preprocessing, feature extraction, classifier training, and performance evaluation.

3.2.2 Hardware tools

A personal computer with the following specifications was used for conducting the experiments: an Intel(R) Core(TM) i5-7200U CPU running at 2.50 GHz (with a maximum frequency of 2.71 GHz), 12 GB of RAM, and a 64-bit Windows 10 operating system. This hardware configuration was adequate for training and evaluating the traditional machine learning models applied in this study.

Because the study relied on computationally efficient algorithms, Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest, no specialized hardware such as GPUs was required. Unlike deep learning models, these methods can be executed effectively on mid-range personal computers without significant performance limitations.

3.3. Methodology

3.3.1 Dataset Collection

A dedicated dataset was needed in order to develop an accurate sentiment analysis model for Amharic text in social media images. Since a dataset that combines Amharic Optical Character Recognition (OCR) with sentiment classification is not publicly available, this study manually collected and prepared its own dataset. Images that contain Amharic text were downloaded from social media platforms, focusing on posts and shared content that convey different sentiments.

Initially, a larger pool of images was gathered: 408 positive, 452 negative, and 393 neutral samples. However, a large number of these had non-Amharic content, unclear characters, noisy backgrounds or duplicate entries. For each sentiment category, 200 high-quality images were

retained in order to guarantee OCR reliability and maintain balanced class sizes, resulting in a final dataset of 600 images. This includes 200 positive images, 200 negative images, and 200 neutral images.

Although the final dataset contains 600 images, this size is appropriate when compared with publicly available Amharic sentiment datasets. Existing resources such as the Amharic news sentiment dataset (approximately 1,500–2,000 text samples) and several social media–based datasets used in prior studies typically contain between 300 and 1,500 labeled instances, often collected from pre-existing text rather than images. Moreover, none of the publicly available datasets combine OCR with sentiment labels, which makes text-in-image datasets significantly harder to collect and annotate. Given the additional challenges of OCR noise, image variability, and manual labeling, a balanced dataset of 600 high-quality Amharic text-in-image samples is consistent with dataset sizes used in earlier Amharic NLP research and is appropriate for evaluating machine learning models in resource-constrained settings.

Two things were taken into consideration in the selection process. First, clean-background images were prioritized to ensure reliable OCR extraction. Second, in order to simulate real-world social media conditions, a limited number of images with noisy or complex-background were included, while excessively noisy samples that could not be processed effectively by the Tesseract OCR engine were excluded.

The finalized dataset was divided into training and testing subsets. The training set consisted of 80% (480 images), while the testing set consisted of 20% (120 images). This division allows the models to learn from a sufficient number of samples while retaining unseen data for objective performance evaluation.

Figure 1 shows sample images from the dataset corresponding to each sentiment category (Positive, Negative, Neutral).

Figure 1: Sample Images from Each Sentiment Category

3.3.2 Dataset Preprocessing

To improve the quality of input data and enhance the performance of sentiment classification models, the collected dataset underwent a series of preprocessing steps. Unlike deep learning–based image classification tasks that rely heavily on augmentation, this study required a dual-level preprocessing approach: **image preprocessing** to prepare raw social media images for OCR and **text preprocessing** to refine the extracted Amharic text before feature extraction.

3.3.2.1 Image Preprocessing

OpenCV was used to process images to improve the clarity of the text region and enhance the accuracy of the Tesseract OCR engine. The procedures listed below were used to preprocess an image.

Grayscale conversion was first applied, where original RGB images were converted to grayscale, reducing computational complexity while emphasizing character boundaries. Following this, thresholding using Otsu's method was performed as a binarization process to separate foreground text from noisy or complex backgrounds. This step proved especially useful in handling images with uneven lighting or low contrast. In some cases, noise reduction was also applied, where

morphological operations were used to smooth small irregularities, making characters more distinguishable for OCR.

These steps enabled more reliable recognition during the OCR phase by improving the quality of Amharic characters.

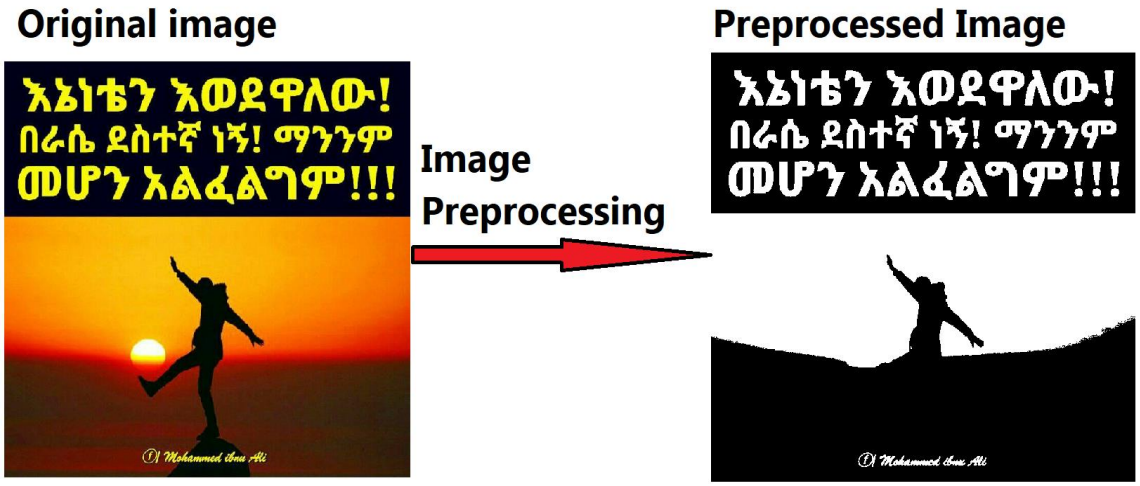

Figure 2: Image Preprocessing

3.3.2.2 Text Extraction (OCR)

Following image preprocessing, text was extracted using **Tesseract OCR** (via the Pytesseract wrapper). Tesseract was configured with the Amharic language model (amh), --oem 3 (neural network-based OCR engine mode), and --psm 6 (assumes a block of text).

Tesseract was selected because it is the most reliable open-source OCR engine with support for complex scripts, including Amharic [17–19]. This step produced raw Amharic text, which often contained noise due to font variation, background interference, or OCR misrecognition.

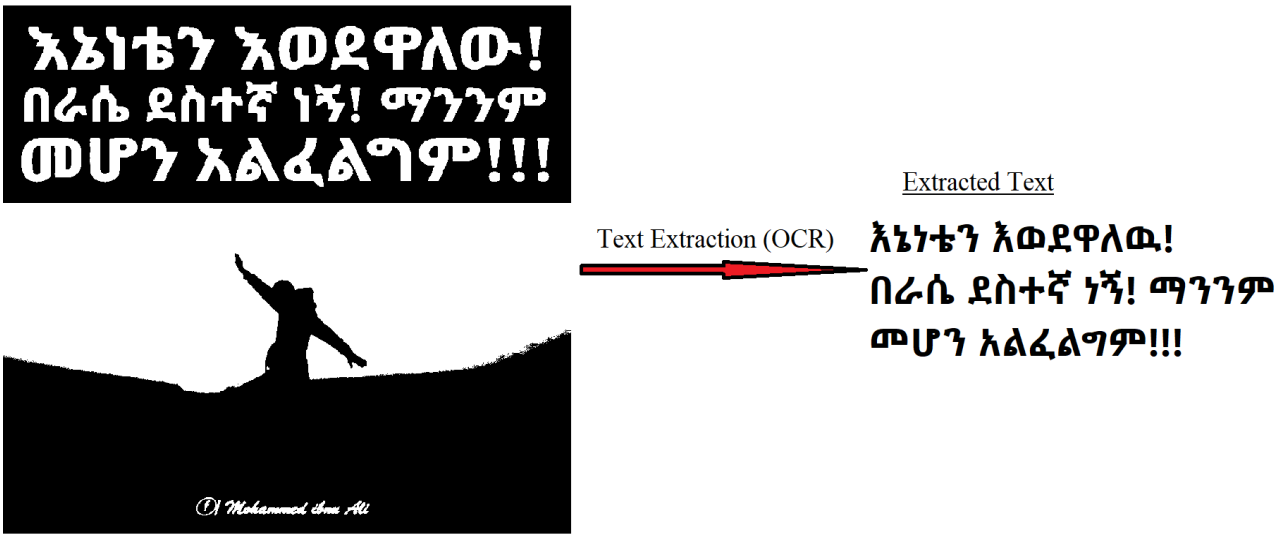

Figure 3: OCR Text Extraction from a Sample Amharic Image

3.3.2.3 Text Preprocessing

Extensive text preprocessing was carried out to clean and normalize the extracted Amharic text because OCR output is rarely perfect. The text preprocessing steps are as follows.

Character normalization was applied first, where equivalent Amharic characters with multiple orthographic representations were unified into a single canonical form. Examples include ሐ mapped to ሀ, ሠ mapped to ሰ, ጸ mapped to ፀ, and ጎ mapped to ሀ. Following this, punctuation and numeral removal was performed. Both Amharic-specific punctuation marks (።, ፣, ፡) and digits, including Ge'ez numerals (፩–፺), were removed to reduce noise. Whitespace normalization was then applied, where multiple consecutive spaces, tabs, or line breaks were replaced with a single space.

Tokenization was carried out by segmenting sentences into individual words for further analysis. Stopword removal followed, eliminating frequently occurring but sentiment-irrelevant words such as "እኔ", "አንተ", "አንቺ", "እሱ", "እሷ", "እኛ", "እናንተ", "እነሱ", "ይህ", "ያ", "ነው", "ናቸው", "ነኝ", and "ነህ", thereby improving the discriminative quality of the dataset. Removal of single characters was also performed, discarding isolated single-character tokens that often result from OCR misrecognition. Finally, removal of English words was applied so that non-Amharic tokens were excluded, retaining only language-specific (Amharic) content.

These steps ensured that the textual data fed into the TF-IDF feature extraction was consistent, noise-free, and linguistically normalized. So, from the text extraction image above, the extracted text is:

“እኔነቴን እወደዋለው

በራሴ ደስተኛ ነኝ! ማንንም

መሆን አልፈልግም!!!”

Preprocessed Text is: “እኔነቴን እወደዋለው በራሴ ደስተኛ ማንንም መሆን አልፈልግም”

3.3.2.4 Dataset Selection and Distribution

An initial pool of social media images containing Amharic text was collected for each sentiment category. Because OCR performance depends heavily on image quality, all collected images were evaluated using the preprocessing and OCR pipeline. Images were retained if the OCR produced readable Amharic text suitable for sentiment classification. On the other hand, Images with illegible text, excessive background noise, non-Amharic content, or severe distortions were excluded.

Maintaining equal representation across sentiment categories (200 images per sentiment class), a balanced dataset of 600 images was selected after quality screening. This ensured that the dataset was large enough to train and test the classifiers.

Table 2: Initial and Final Dataset Distribution Across Sentiment Classes

Sentiment Class	**Images Collected**	**Images Retained (Used)**
Positive	408	200
Negative	452	200
Neutral	393	200
Total	1,253	600

The retained dataset was then split into training and testing subsets as follows. training set 80% (480 images) and test set 20% (120 images). This procedure ensured that only high-quality OCR outputs were used for modeling, while still reflecting the diversity of real-world social media images.

3.3.3 Building the Model

3.3.3.1 Feature Extraction

The Term Frequency–Inverse Document Frequency (TF-IDF) is one of the most widely adopted feature extraction methods in text mining and information retrieval, as it balances the importance of a term within a document against its overall distribution across the corpus [24,25]. The preprocessed Amharic text obtained from OCR was transformed into numerical features using TF-IDF vectorization.

In this approach, the term frequency (TF) measures how often a word appears in a given document, while the inverse document frequency (IDF) reduces the weight of terms that occur frequently across many documents. Because of this, TF-IDF assigns higher weights to terms that are distinctive for particular documents and lower weights to common words that carry little discriminative power. This property makes TF-IDF particularly suitable for sentiment analysis tasks, as it highlights keywords and expressions that signal positive, negative, or neutral sentiment [24].

The TF-IDF vectorizer was applied to the clean and normalized Amharic text that was extracted from social media images. The process generated a high-dimensional but sparse feature matrix, where each row corresponds to a document (an extracted text) and each column corresponds to a vocabulary term. These sparse representations are ideal for machine learning algorithms and are computationally efficient, since they reduce memory usage while allowing fast model training [25]. In this configuration, only unigram features were considered, reflecting a trade-off between capturing linguistic detail and managing the limited dataset size and computational resources. The resulting The TF-IDF vectors were further refined through feature selection before being used as input to the classifiers, such as Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest.

TF-IDF was selected for this work because it provides an effective and computationally efficient way for representing text in Amharic, particularly when working with limited and noisy OCR-extracted data. TF-IDF can be used directly on small datasets and yet capture informative term weights for sentiment classification, unlike deep contextual embeddings that require extensive training resources. Along with producing sparse feature vectors that align well with linear classifiers like Support Vector Machine (SVM) and Logistic Regression, it is also robust to spelling variations and noise caused by OCR. Additionally, TF-IDF has been widely applied in prior Amharic sentiment analysis research, making it a reliable and comparable baseline for this work.

Following feature extraction, a feature selection step was incorporated to reduce the dimensionality of the feature space. Feature selection was included not only for performance improvement but also to evaluate dimensionality reduction strategies for Amharic text, where high sparsity is common.

3.3.3.2 Feature Selection (Chi-Square with SelectKBest)

After feature extraction using TF-IDF, an additional feature selection step was implemented to reduce dimensionality and retain the most informative features. The SelectKBest method with the Chi-square (χ^2) statistical test was applied to rank and select top features based on their relevance to sentiment classes. Different values of *k* (1000, 2000, and 3000) were evaluated to analyze the impact of feature selection on model performance. Then balanced configuration k=3000 was set.

3.3.3.3 Overview of Selected Models

This study employed classical supervised machine learning algorithms in combination with Term Frequency–Inverse Document Frequency (TF-IDF) for feature extraction.

To evaluate performance on Amharic sentiment analysis, four classifiers were implemented:

1. **Logistic Regression:** Logistic Regression is one of the most commonly employed linear models in text classification due to its simplicity, efficiency, and strong interpretability. It estimates the probability that a given input belongs to a particular class using the logistic (sigmoid) function, which maps real-valued features into probabilities between 0 and 1. This makes the model well-suited for binary classification tasks, while extensions such as the multinomial logistic regression with a softmax function allow it to effectively handle

multiclass problems. Due to its capacity to handle high-dimensional feature spaces, logistic regression is frequently preferred in the context of classification of texts, such as those generated by TF-IDF, while maintaining computational efficiency and robustness against overfitting [26].

2. **Multinomial Naive Bayes:** This classification approach is based on Bayes' theorem and assumes conditional independence among terms. Despite its strong assumptions, it often performs very well in text classification tasks when feature representations are sparse and high-dimensional. Because of its simplicity and efficiency, this model was included among the classifiers evaluated in this study [26].
3. **Support Vector Machine (SVM):** It is a maximum-margin classifier that identifies the optimal hyperplane separating data points into distinct classes. With linear kernels, SVM is especially effective in handling high-dimensional and sparse data, which is common in text classification tasks using feature extraction methods like TF-IDF. Due to its stability and efficiency, it has become one of the most frequently used techniques in natural language processing applications [27]. Moreover, recent research has shown that SVM continues to deliver strong results on text datasets with many features, especially when integrated with topic modeling or dimensionality reduction methods, demonstrating its effectiveness for sentiment classification of sparse Amharic text [28].
4. **Random Forest:** Random Forest is an ensemble learning algorithm that constructs multiple decision trees and aggregates their predictions to improve accuracy and control overfitting. When compared to a single decision tree, it produces stable and generalizable results by combining the outputs of several trees trained on bootstrapped subsets of the data [29]. Random Forest has shown promise in handling high-dimensional and sparse feature spaces in the context of text classification. For document classification tasks where interpretability and robustness are crucial, it is an appropriate alternative due to its ability to rank feature importance and reduce variance [30].

These models were selected because they are computationally efficient, theoretically well-grounded, and widely used in sentiment analysis and text classification studies. In addition, they can be trained effectively on the available dataset and hardware without requiring GPUs or large-scale deep learning infrastructure.

3.3.3.4 Hyperparameter Selection

In this study, most classifiers were trained using default hyperparameters provided by scikit-learn library, which are generally well-suited for small to medium-sized datasets. Minor changes were made, nevertheless, in order to ensure the stability and convergence of the model. The maximum number of iterations for logistic regression was increased to 1000 to guarantee convergence on the TF-IDF feature space, and the number of estimators for the Random Forest classifier was set to 200 with a fixed random seed to improve prediction stability and reproducibility.

For the Naive Bayes and Support Vector Machine classifiers, default hyperparameters were used, as they provided stable performance without further adjustments. No systematic hyperparameter optimization (e.g., grid search or randomized search) was conducted, since the primary objective of this study was to compare the performance of different classifiers under consistent experimental conditions rather than to exhaustively fine-tune each model.

3.3.4 Evaluation Metrics

To assess the performance of the sentiment classification models, this study employed four widely used evaluation metrics: **accuracy, precision, recall, and F1-score**. These evaluation metrics provide an extensive understanding of the models' performance in identifying and classifying Amharic texts into neutral, negative, and positive sentiment categories. In addition to these metrics, confusion matrices were generated for each classifier to provide a visual representation of correctly and incorrectly classified samples.

Before presenting the metrics, we first define the fundamental terms used in classification evaluation. True Positive (TP) is the number of samples belonging to a specific class that are correctly classified as such. True Negative (TN) is the number of samples not belonging to a specific class that are correctly identified as not belonging. False Positive (FP) is the number of samples incorrectly predicted as belonging to a class when they do not (Type I Error). False Negative (FN) is the number of samples incorrectly predicted as not belonging to a class when they do (Type II Error).

Using these definitions, the following metrics were computed:

(i) Accuracy (ACC): Accuracy measures the overall proportion of correctly classified samples across all classes. It is expressed as:

$$ACC = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Accuracy provides a general indication of performance, although it can be misleading in cases of imbalanced datasets.

(ii) Precision (P): Precision quantifies the proportion of correctly predicted positive samples out of all samples predicted as positive. It evaluates the reliability of the classifier when labeling a sample as belonging to a particular sentiment class:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

High precision indicates that the model makes few false positive errors.

(iii) Recall (R): Recall, also known as **sensitivity**, measures the proportion of actual positive samples correctly identified by the classifier:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

High recall indicates that the model successfully identifies most of the relevant samples, minimizing false negatives.

(iv) F1-Score (F1): The F1-score is the harmonic mean of precision and recall, balancing both metrics in a single measure:

$$P = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is particularly useful when dealing with class imbalances, as it penalizes extreme differences between precision and recall.

These four metrics were selected because they provide complementary perspectives: **accuracy** gives an overall measure, **precision** emphasizes correctness, **recall** emphasizes completeness, and **F1-score** balances the two. Together, they form a robust framework for evaluating the effectiveness of sentiment classification models in this study.

3.3.5 Confusion Matrix

The model predictions were analyzed using the confusion matrix in addition to numerical performance metrics. It is a tabular representation that compares the classifier's predicted sentiment labels with the actual sentiment labels. A 3x3 confusion matrix, representing the three sentiment categories of positive, negative, and neutral, was employed in this study.

Each cell of the matrix shows the number of instances classified into a particular category. The diagonal elements represent correctly classified samples. And the elements that are out of diagonal indicate misclassifications. So, it is possible to identify which classes are more prone to misclassification and assess the strengths and weaknesses of each classifier, by examining these values.

The confusion matrix therefore provides a more comprehensive view of model performance than single-value metrics, since it highlights not only overall accuracy but also the distribution of errors across sentiment classes.

3.4 System Architecture

The proposed Amharic sentiment analysis system architecture is organized into sequential modules, each of which is responsible for a distinct pipeline stage:

- **Image Input:** Social media images containing Amharic text were gathered and used as an input.
- **Image Preprocessing***:* Images are converted into grayscale and binarized using Otsu's method to enhance text visibility.
- **Text Extraction (OCR):** Amharic text is extracted from images using the Tesseract OCR engine with the Pytesseract wrapper.
- **Text Preprocessing***:* To generate clean textual inputs, extracted text is normalized, stopwords are removed, punctuation and numerals are removed, and tokenization is used.
- **Feature Extraction:** The preprocessed text is transformed into numerical features using the TF-IDF vectorization technique, enabling the models to capture discriminative word-level information.
- **Feature Selection:** Feature selection using SelectKBest method with the Chi-square (χ^2) test was implemented to reduce dimensionality and retain the most informative features.
- **Building Model:** Classical machine learning classifiers are employed for sentiment classification, rather than deep learning approaches.
- **Sentiment Classification:** The features are classified using machine learning classifiers into three sentiment classes Positive, Negative and Neutral**.**

The overall workflow of the proposed Amharic sentiment analysis system is illustrated in Figure 4.

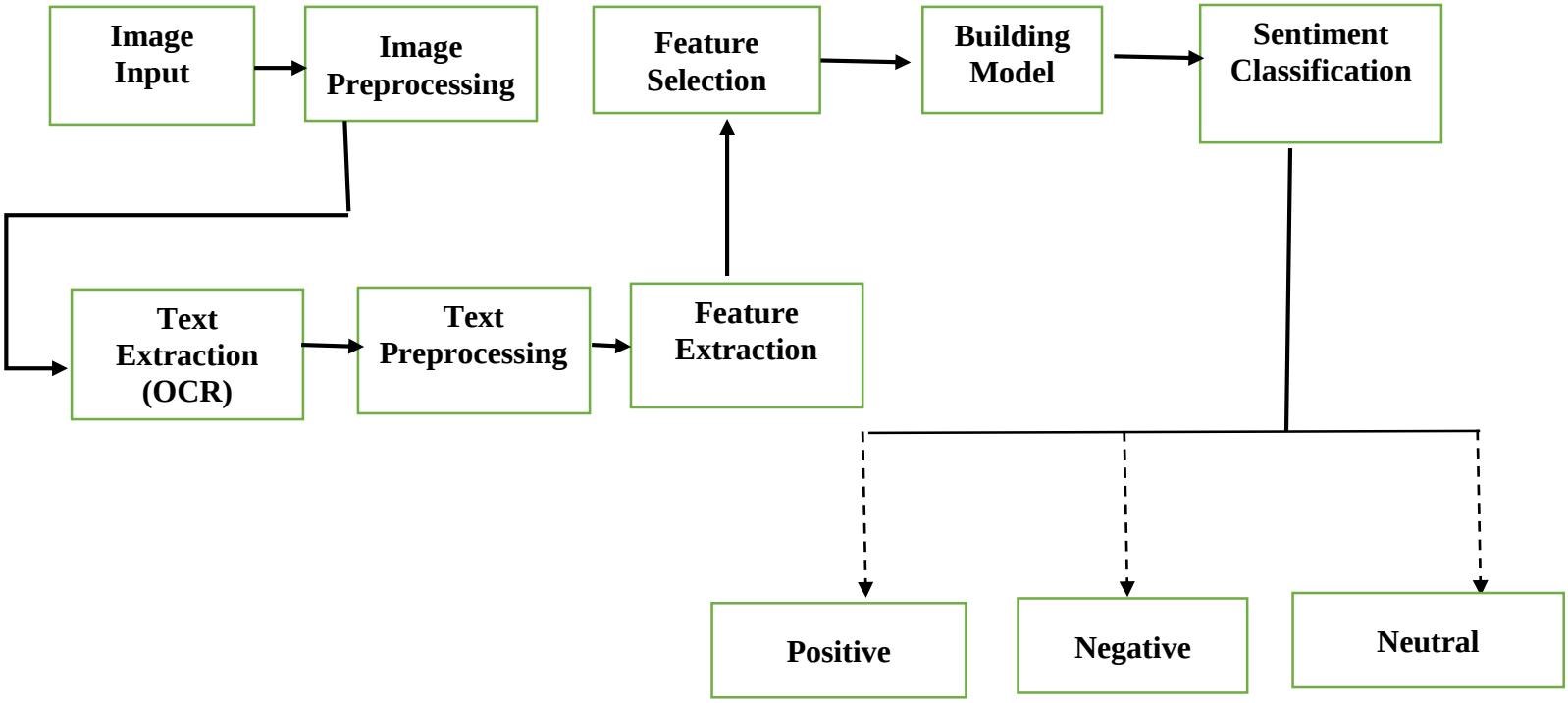

Figure 4: System architecture of the proposed pipeline

Chapter Four

4. Results and Discussion

4.1 Dataset Overview

The performance of any machine learning or natural language processing model is highly dependent on the quality and representativeness of the dataset. Since no publicly available dataset combines Amharic text extracted from images with sentiment labels, this study developed its own dataset through a systematic collection, manual labeling, and preprocessing.

A total of 1,253 candidate images were collected (408 positive, 452 negative, 393 neutral); after quality screening, 600 high-quality images (200 per class) were retained. The data was primarily collected from widely used platforms such as Facebook, Instagram, LinkedIn, and X. In these platforms, Amharic is widely used to express opinions, emotions, and sociopolitical sentiments. To ensure coverage across a range of expressions, images were selected to reflect a variety of subjects, such as politics, social issues, cultural critiques, and general discussions.

The dataset was manually annotated into three sentiment categories:

- **Positive sentiment (200 images):** Included words of encouragement, peace, happiness, optimism, or support. Messages of support, hope, kindness, expressions of joy, and congratulatory posts are a few examples.
- **Negative sentiment (200 images):** Conveyed dissatisfaction, criticism, sadness, anger, or rejection. Common cases involved complaints about social or political conditions, tragic events, and emotionally charged expressions such as "ጦርነት" (war) or "ግድያ" (massacre).
- **Neutral sentiment (200 images):** Included texts expressing factual information, announcements, or general commentary without strong emotional content.

The dataset was deliberately balanced with equal representation of 200 samples per sentiment class. This balance is crucial to prevent model bias toward a majority class, a common issue in sentiment analysis [4]. Balanced datasets ensure that each classifier receives adequate examples for learning patterns across all classes.

The dataset was further divided into training and testing sets using an 80:20 split. Accordingly, 480 images were used for training the models, while 120 images were reserved for testing. This ratio is widely adopted in sentiment analysis research as it provides sufficient training data while preserving enough samples for reliable evaluation [1,3]. The dataset distribution across training, test and OCR results is shown in Figure 5.

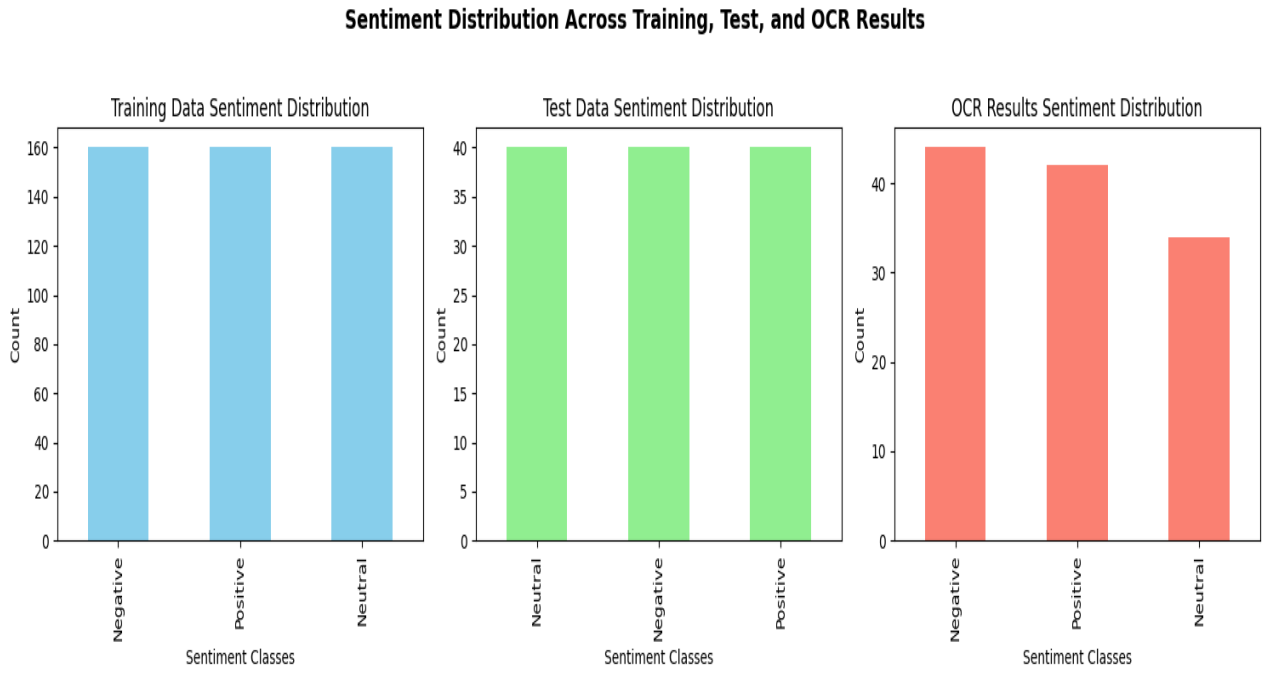

Figure 5: Sentiment distribution across training, test, and OCR results

During annotation, special care was taken to avoid ambiguity in labeling. Annotation was performed by the researcher, and careful cross-checking was performed to maintain consistency. To minimize subjectivity, the labeling guidelines were applied uniformly across all samples, which is particularly important in a morphologically rich and context-dependent language like Amharic [6].

To illustrate the composition of the dataset, representative examples from each class are shown in Figure 6.

Figure 6: Sample images representing all sentiment categories in the dataset

4.2 OCR and Text Preprocessing Results

After the dataset was finalized, the Amharic text that was embedded in the images had to be converted into a machine-readable format. Optical Character Recognition (OCR) was used to achieve this process, followed by a sequence of text preprocessing steps designed to handle the linguistic complexities of Amharic.

4.2.1 OCR Extraction

The images were first subjected to OCR using a pipeline that combined OpenCV for image preprocessing and Tesseract OCR as the primary text recognition engine [6].

Before text extraction, images underwent preprocessing to improve recognition accuracy. These are:

- Grayscale conversion: to reduce color channels and improve contrast
- Thresholding: binarizing the image to make text detection clearer.
- Noise removal: eliminating small background distortions that often confuse OCR engines.

Preprocessing was essential since raw OCR without these procedures frequently resulted in characters that were distorted or incomplete, particularly in images with overlapping graphics, background textures or shadows. By applying preprocessing, the recognition rate improved significantly, making the text more suitable for further analysis [1,7].

But some difficulties persisted. Some images with very noisy backgrounds, decorative Amharic fonts, or overlapping watermark symbols resulted in incomplete or incorrect extractions. Occasionally, these errors added rare or nonsensical tokens to the dataset. For example, OCR might misinterpret the word "ፍቅር" (love) as "ፍትር" or "ርቅር". This produces variations that had no intended or real meaning. Fortunately, because such junk tokens appeared infrequently, their impact on the overall classification process was minimal, especially after TF-IDF transformation, which naturally reduces the weight of rare words.

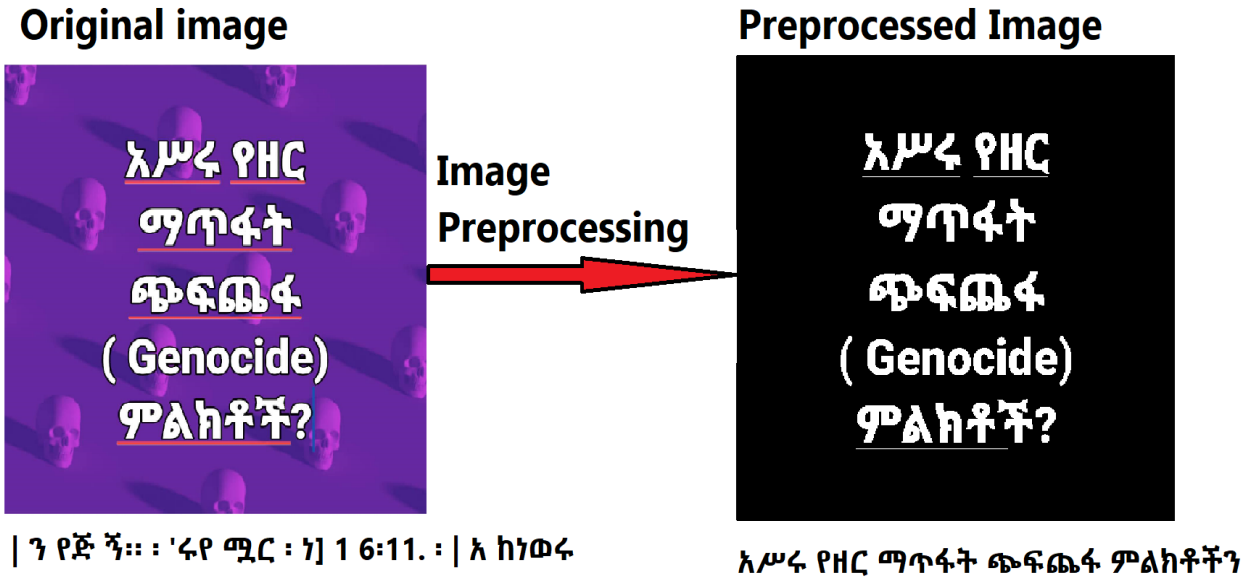

Figure 7: Sample output of OCR before preprocessing and after preprocessing

4.2.2 Text Normalization and Cleaning

After OCR, the raw text underwent linguistic preprocessing to ensure uniformity and reduce noise. For Amharic, which is characterized by script variations, redundant characters, and morphologically complex words, this phase is very important [4].

The preprocessing pipeline included the following steps:

1. **Punctuation removal** – eliminating non-textual characters (.,!?፡።/ etc.).
2. **Digit and Geez numeral removal** – excluding Arabic numbers (0–9) and Ethiopic numerals (፩, ፪, etc.).
3. **Stopword removal** – removing frequent function terms that don't have a standalone meaning and provide little sentiment information, “ይህ,” “እንደ,” “እሱ,” or “ነው,”.
4. **Character normalization** – unifying equivalent Amharic characters into their canonical forms to reduce orthographic inconsistency. Examples include:

   ሐ → ሀ
   ሠ → ሰ
   ጸ → ፀ
   ኀ → ሀ

5. **Tokenization** – separating sentences into words that can be used as tokens to extract features.

This process follows earlier Amharic NLP works, where preprocessing was shown to improve sentiment classification by reducing ambiguity caused by script variations [9,12].

An example of this process is shown in Table 3, where an OCR-extracted text containing noise was transformed into a normalized, sentiment-ready format.

Table 3: Example of OCR text before and after normalization and cleaning.

S.No	Raw OCR Output	After Cleaning & Norm
1	አሥሩ የዘር ማጥፋት ጭፍጨፋ ምልክቶችን?	አስሩ የዘር ማጥፋት ጭፍጨፋ ምልክቶችን
2	እውነተኛ ወዳጅ ማለት ቀን ሲጨልም መብራት ይዞ የሚመጣ ነው።	እውነተኛ ወዳጅ ማለት ቀን ሲጨልም መብራት ይዞ የሚመጣ
3	የአማራ ጠላት ማን ነው?	የአማራ ጠላት
4	በኢትዮጵያ ለገና ልብስ አይገዛልህም የተባለው የ25 አመት ወጣት ራሱን አጠፋ!!	በኢትዮጵያ ለገና ልብስ አይገዛልህም የተባለው አመት ወጣት ራሱን አጠፋ

4.2.3 TF-IDF Feature Representation

The cleaned text was vectorized using the Term Frequency–Inverse Document Frequency (TF-IDF) method. This technique converts text into numerical form by emphasizing words that are frequent within a document but rare across the corpus, which is highly effective for sentiment analysis [10,11].

For instance, TF-IDF vectors showed a high frequency of sentiment-bearing words like “ደስታ” (happiness) or “ግድያ” (killing), which allowed classifiers to acquire discriminative features for sentiment prediction.

As is common for text classification tasks, the resulting feature space was sparse and high-dimensional. This representation was particularly well-suited for linear models such as Logistic

Regression and SVM, which rely on separating hyperplanes in high-dimensional spaces, as confirmed by their superior performance in later sections [3,4].

4.3 Experimental Setup

The experimental framework for evaluating the performance of different classifiers on Amharic sentiment analysis was designed after the dataset was prepared and textual features were extracted. This subsection outlines the feature extraction method, classifiers, and evaluation metrics used in the study.

4.3.1 Feature Extraction with TF-IDF

The preprocessed Amharic texts were converted into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) method. By giving terms that appear frequently in a document but less frequently throughout the dataset higher weights, TF-IDF highlights sentiment-bearing words and downweights common or uninformative ones. This method works well for handling high-dimensional and sparse text data and has been used extensively in Amharic sentiment analysis tasks [10,11].

TF-IDF was applied in this study in the following setup:

- **Tokenizer**: Preprocessing and normalization were performed separately before applying TF-IDF.
- **N-grams**: Only unigrams (single words) were considered. Although character n-grams have been shown to capture Amharic morphology more effectively, unigram features were chosen in this study due to the relatively small dataset size and to reduce computational complexity.
- **Feature size**: Without limiting the maximum number of features, the corpus automatically determined the vocabulary.
- **Weighting scheme**: Standard TF-IDF weighting with L2 normalization (default setting).

This representation produced a high-dimensional and sparse feature space. This is particularly well-suited for linear classifiers such as Logistic Regression and Support Vector Machine (SVM).

4.3.2 Feature Selection using Chi-Square

In addition to the baseline TF-IDF representation, a feature selection technique was implemented to analyze its effect on classification performance. The SelectKBest method with the Chi-square (χ^2) statistical test was applied to select the most informative features based on their association with sentiment classes. Different values of *k* (1000, 2000, and 3000) were experimented to reduce the dimensionality of the TF-IDF feature space and balanced configuration set to k=3000.

4.3.3 Classifiers

Four classifiers were used on the identical TF-IDF representations to evaluate the effectiveness of different approaches. To ensure consistency in feature representation across models, all classifiers were trained using unigram-based TF-IDF features.

1. **Logistic Regression (LogReg):** A popular linear model for text classification because of its robustness and effectiveness in high-dimensional domains. Earlier Amharic sentiment studies also confirmed Logistic Regression as a strong baseline for classification tasks [1]. In order to ensure model convergence, the maximum number of iterations in this study was set at 1000.
2. **Naive Bayes (NB):** A probabilistic classifier depending on Bayes' theorem and the assumption of conditional independence among features. Despite its simplicity, it has been effectively used on Amharic datasets and frequently offers competitive baseline performance in sentiment analysis [4,5]
3. **Support Vector Machine (SVM):** A linear kernel SVM was selected, as this method is particularly effective for sparse TF-IDF vectors and has consistently shown strong results in Amharic sentiment classification [1,3].
4. **Random Forest (RF):** An ensemble of decision trees was included to evaluate whether tree-based methods could generalize well on Amharic sentiment data. nevertheless, previous studies have reported Random Forest tends to perform less effectively on high-dimensional sparse TF-IDF features [10].

4.3.4 Evaluation Metrics

Standard evaluation metrics were used to assess each classifier's performance:

- **Accuracy**: The proportion of correctly predicted instances among all test instances.
- **Precision**: The proportion of correctly predicted cases among all predictions for a given class.
- **Recall**: The proportion of correctly identified cases among all actual cases of that class.
- **F1-score**: The harmonic mean of recall and precision, which balances the two.

Every metric was calculated for each of the three classes (positive, negative, and neutral), and both weighted averages (which take class sizes into consideration) and macro-averages (which treat classes equally) were used to summarize the results.

Additionally, confusion matrices were generated for each classifier to visualize patterns of misclassification. These matrices provided deeper insight into class-specific challenges, particularly in distinguishing Negative from Neutral sentiment, a difficulty also noted in previous Amharic sentiment studies [1,6].

Four classifiers—Logistic Regression, Naive Bayes, SVM, and Random Forest—as well as evaluation metrics including accuracy, precision, recall, F1-score, and confusion matrices comprised the experimental setting for this work. Table 4 provides a summary of the information.

Table 4: Summary of experimental setup (TF-IDF configuration, classifiers, and evaluation metrics).

Component	Configuration / Description
Feature Extraction (TF-IDF)	
Feature Selection	SelectKBest (Chi-square), k = 3000
Tokenizer	Preprocessing and normalization performed before TF-IDF; default whitespace-based tokenization used
N-grams	Unigrams only (single words)
Feature size	3000
Weighting scheme	Standard TF-IDF with L2 normalization
Classifiers	
Logistic Regression	Linear model, max iterations = 1000

Naive Bayes	Multinomial Naive Bayes
Support Vector Machine (SVM)	Linear kernel
Random Forest	Ensemble of decision trees
Evaluation Metrics	
Metrics	Accuracy, Precision, Recall, F1-score
Additional tools	Confusion matrices for each classifier; macro-average and weighted-average scores

4.4 Results Presentation

In this section, the performance of the four classifiers (Logistic Regression, Naive Bayes, Support Vector Machine (SVM), and Random Forest) trained on the TF-IDF features extracted from OCR-preprocessed Amharic texts was presented. Each classifier's performance is reported using accuracy, precision, recall, and F1-score, supported by confusion matrices and classification reports.

4.4.1 Logistic Regression

Logistic Regression was used as baseline classifier because of its well-established effectiveness in text classification tasks. To ensure convergence during training, the maximum number of iterations in this study was set at 1000.

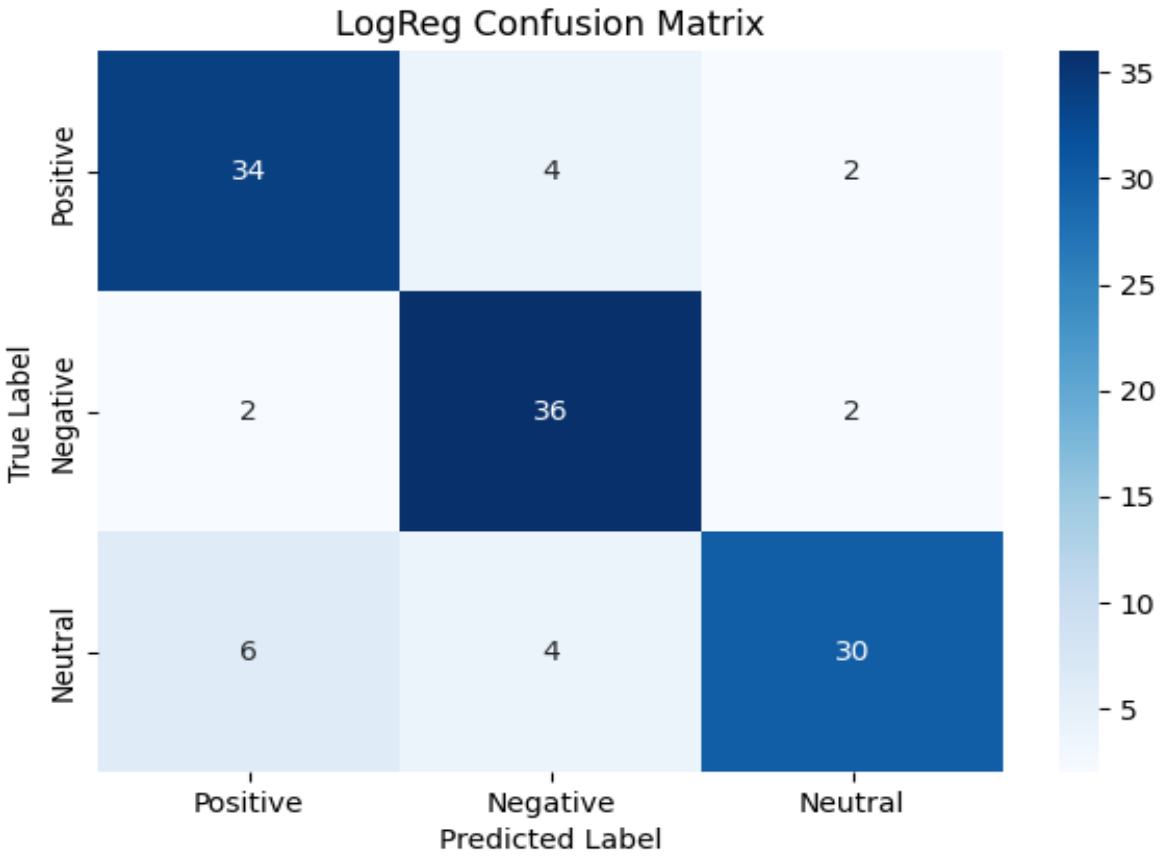

Figure 8: Confusion Matrix for Logistic Regression classifier.

Table 5: Classification report for Logistic Regression.

Sentiment	**Precision**	**Recall**	**F1-score**	**Support**
Positive	0.82	0.90	0.86	40
Negative	0.88	0.75	0.81	40
Neutral	0.81	0.85	0.83	40
Overall Accuracy	-	-	0.83	120
Macro Avg	0.84	0.83	0.83	-
Weighted Avg	0.84	0.83	0.83	-

Discussion:

With an overall accuracy of 83%, Logistic Regression demonstrated strong performance. The model performed best in detecting Positive sentiment (Recall = 0.90), showing its ability to capture supportive or optimistic expressions. Negative sentiment recall, however, was lower (0.75). This indicates that some cases that were Negative were misclassified as Neutral. This difficulty in separating Negative from Neutral sentiment has been reported in earlier Amharic sentiment studies [8,10]. Given these results, the next step was to examine whether a probabilistic model like Naive Bayes could better handle Amharic text features.

4.4.2 Naive Bayes

The Naive Bayes classifier was tested to evaluate a probabilistic approach under the TF-IDF representation.

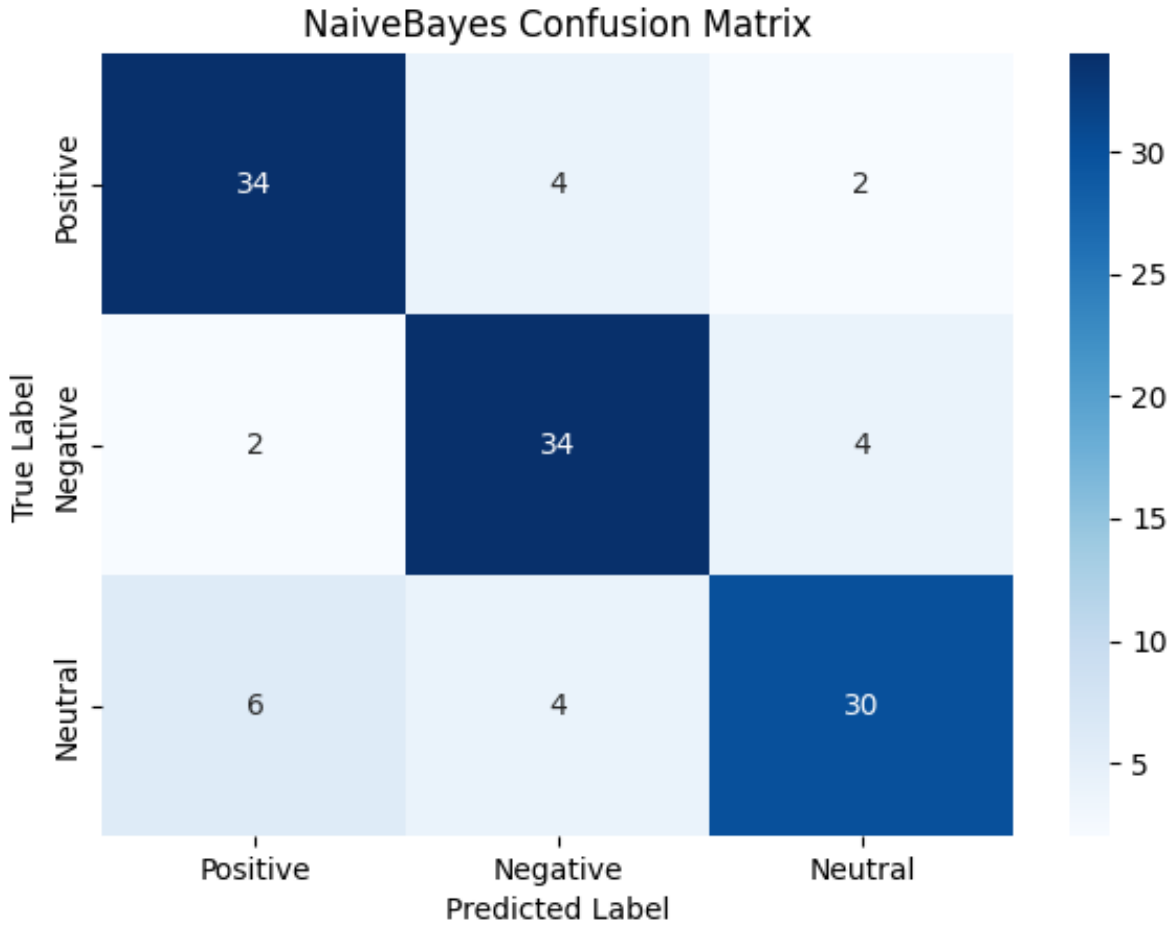

Figure 9: Confusion Matrix for Naive Bayes classifier.

Table 6: Classification report for Naive Bayes.

Sentiment	**Precision**	**Recall**	**F1-score**	**Support**
Positive	0.81	0.85	0.83	40
Negative	0.83	0.75	0.79	40
Neutral	0.81	0.85	0.83	40
Overall Accuracy	-	-	0.82	120
Macro Avg	0.82	0.82	0.82	-
Weighted Avg	0.82	0.82	0.82	-

Discussion:

Naive Bayes achieved 82% accuracy, slightly lower than Logistic Regression. Performance was balanced across all classes, though precision for the Negative class was slightly weaker (0.83) compared to the Positive. This shortcoming results from Naive Bayes' independence assumption, which fails to adequately capture the morphological dependencies in Amharic. Despite this, the model provided competitive results, supporting findings from earlier studies where Naive Bayes offered reasonable baseline performance for Amharic sentiment classification [4]. To further test robustness, a Support Vector Machine was then applied, given its strong reputation in text classification.

4.4.3 Support Vector Machine (SVM)

SVM was expected to excel due to its ability to handle sparse, high-dimensional TF-IDF vectors.

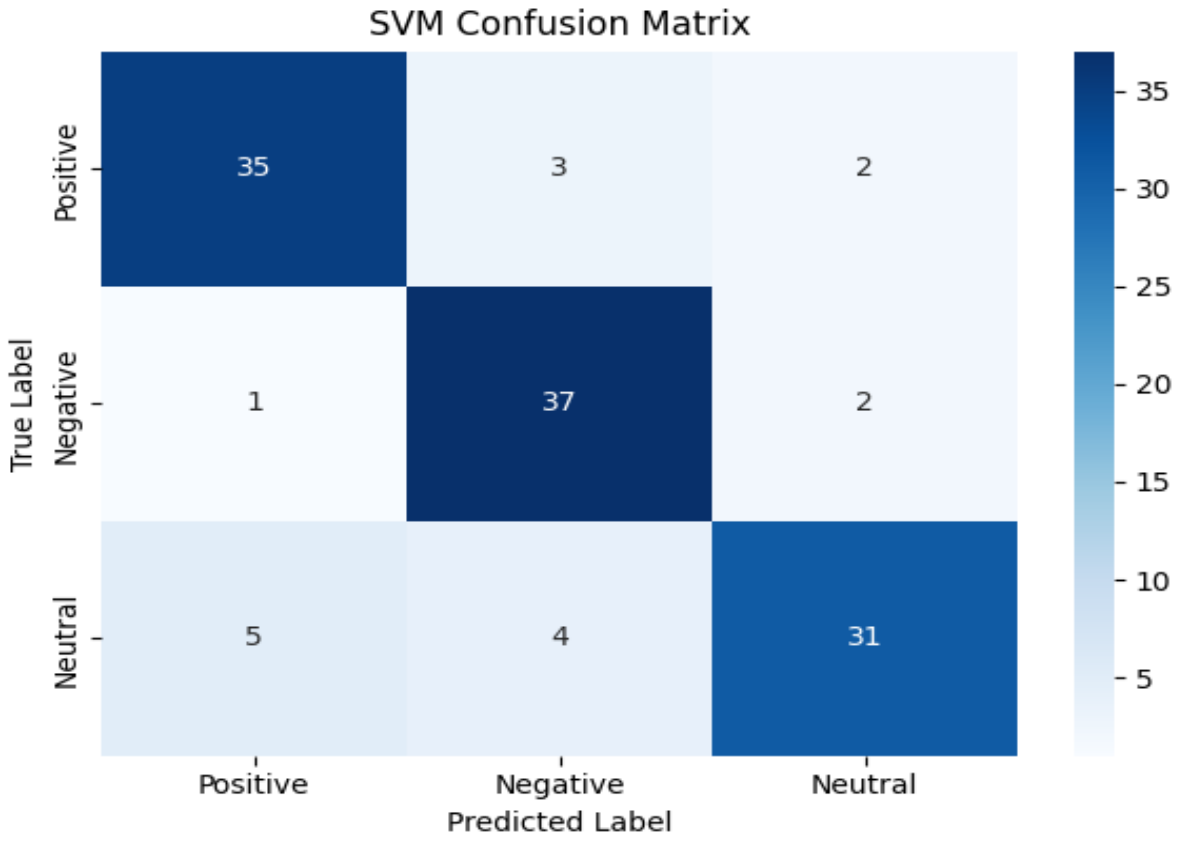

Figure 10: Confusion Matrix for SVM classifier.

Table 7: Classification report for SVM.

Sentiment	**Precision**	**Recall**	**F1-score**	**Support**
Positive	0.84	0.93	0.88	40
Negative	0.89	0.78	0.83	40
Neutral	0.85	0.88	0.86	40
Overall Accuracy	-	-	0.86	120
Macro Avg	0.86	0.86	0.86	-
Weighted Avg	0.86	0.86	0.86	-

Discussion:

SVM achieved the best performance, with an overall accuracy of 86% and balanced precision, recall and F1-scores across all classes. With a recall of 0.93, it was especially good at detecting Positive sentiment, while also producing good results for the Neutral and Negative classes. As previously observed in Amharic sentiment research [7,11], the improved performance validates SVM's effectiveness in sparse, high-dimensional spaces. Finally, an ensemble-based model, Random Forest, was evaluated to assess whether tree-based methods could rival linear classifiers.

4.4.4 Random Forest

Random Forest was included to evaluate the performance of ensemble tree-based methods on Amharic sentiment data.

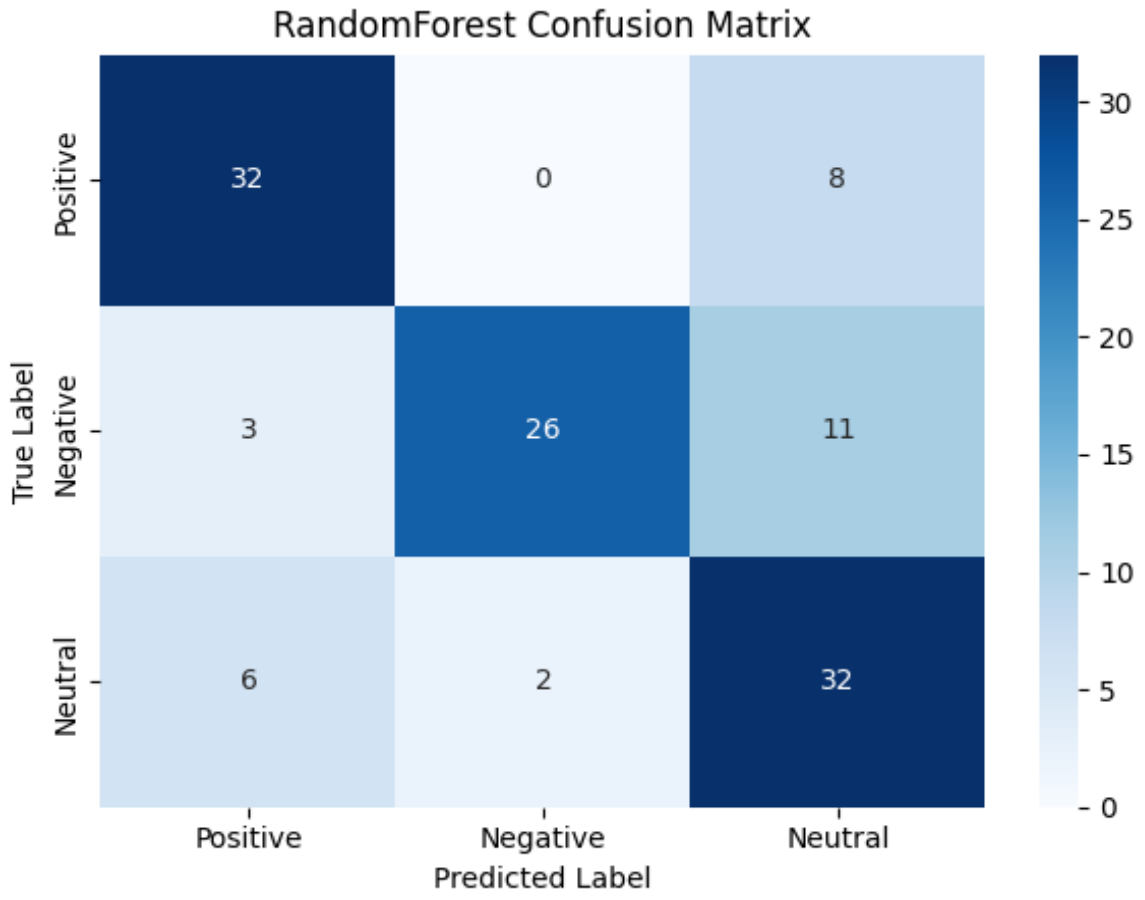

Figure 11: Confusion Matrix for Random Forest classifier.

Table 8: Classification report for Random Forest.

Sentiment	**Precision**	**Recall**	**F1-score**	**Support**
Positive	0.93	0.65	0.76	40
Negative	0.63	0.80	0.70	40
Neutral	0.78	0.80	0.79	40
Overall Accuracy	-	-	0.75	120
Macro Avg	0.78	0.75	0.75	-
Weighted Avg	0.78	0.75	0.75	-

Discussion:

Random Forest performed the lowest, with an accuracy of 75%. Its weak recall (0.65) and high precision (0.93) for positive sentiment suggested that positive texts were frequently misclassified. Similarly, the Negative class suffered from low precision (0.63). These findings imply that tree-based ensemble techniques are not suitable for sparse TF-IDF features, as they tend to overfit and struggle to generalize. Similar weaknesses of Random Forest have been noted in prior Amharic text classification studies [10].

4.4.5 Comparative Performance of Classifiers

To enable a holistic view of model performance, the results of all four classifiers are summarized in Table 9.

Table 9: Comparative performance of classifiers on Amharic sentiment analysis (test set, 120 samples).

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.83	0.84	0.83	0.83
Naive Bayes	0.82	0.82	0.82	0.82
SVM	0.86	0.86	0.86	0.86
Random Forest	0.75	0.78	0.75	0.75

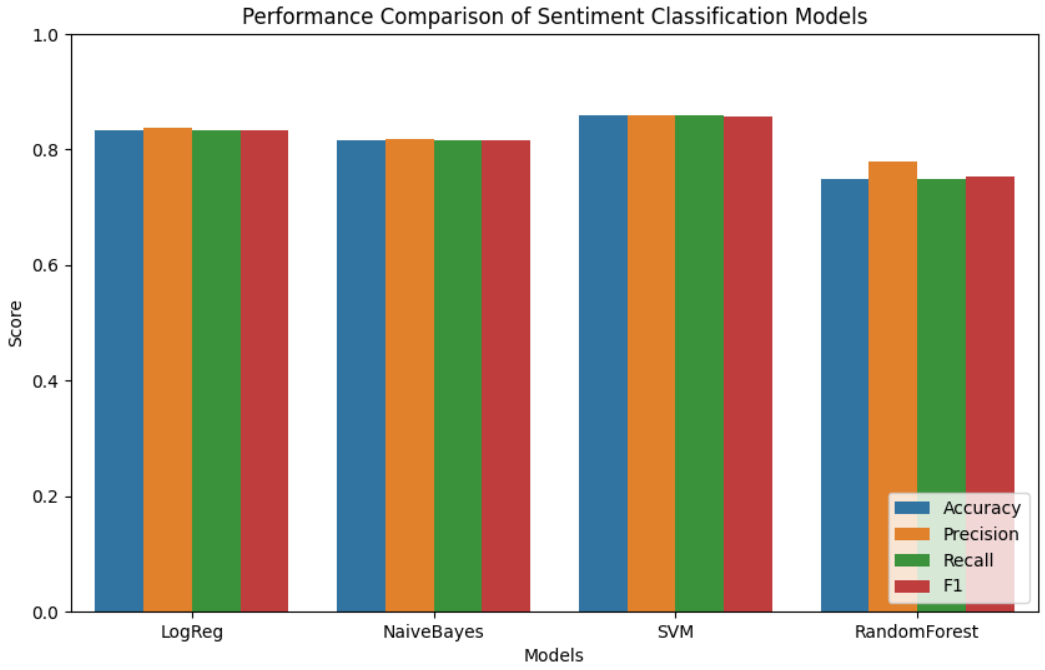

Figure 12: Performance comparison of classifiers based on Accuracy, Precision, Recall, and F1-score.

Comparative Analysis:

According to the comparative results, SVM outperformed all other classifiers by consistently achieving the highest scores across all metrics. Logistic Regression followed closely, confirming the strength of linear models for Amharic sentiment analysis. Despite producing somewhat weaker results, Naive Bayes was still competitive, making it a lightweight alternative. Random Forest lagged behind, demonstrating the limitation of using tree-based ensemble techniques for sparse text characteristics.

A noteworthy pattern observed in all models was the increased challenge of accurately detecting negative sentiment, which was frequently mistaken for neutral sentiment. This reflects the linguistic complexity of Amharic, where negative expressions can be subtle and context-dependent. Similar challenges have been documented in previous studies [1,7].

To summarize, **SVM demonstrated the most reliable performance**, while all models produced encouraging results. Therefore, SVM is the most suitable classifier for Amharic sentiment analysis in this study.

4.5 Discussion of Findings

The four classifiers' results offer a number of significant insights on the feasibility and challenges of performing sentiment analysis on Amharic text extracted from social media images. In this section we will discuss the findings in light of the linguistic characteristics of Amharic, the role of OCR preprocessing, and the comparative performance of machine learning models.

4.5.1 Effectiveness of OCR and NLP Pipeline

This study's main contribution is that it integrates OCR and NLP. Unlike previous Amharic sentiment works that relied solely on pre-collected textual data [3,4,5,9,10,12], this research demonstrates that sentiment can be automatically extracted from image-based social media content.

OCR preprocessing significantly improved recognition accuracy, especially through grayscale conversion, thresholding, and noise removal. Nonetheless, images with noisy backgrounds, decorative fonts, or overlapping watermarks still introduced artifacts, leading to occasional junk tokens (e.g., distorted Amharic words, as noted in Section 4.2). Because such artifacts appeared infrequently, their effect was minimized by TF-IDF weighting, which down-weights rare terms. This outcome highlights the robustness of the OCR + preprocessing + TF-IDF pipeline, though future work could benefit from OCR-specific noise-handling techniques or contextual embeddings.

4.5.2 Model Performance Trends

The comparative results revealed that SVM consistently outperformed the other classifiers, with Logistic Regression ranking in second. This pattern is consistent with earlier Amharic sentiment studies [1,3], which emphasized the effectiveness of linear models in sparse, high-dimensional feature spaces.

- **SVM** performed extremely well since it could maximize the margins between classes, crucial when subtle linguistic variations in Amharic create overlapping distributions.
- **Logistic Regression** also performed competitively, demonstrating that basic linear models are still effective baselines for Amharic sentiment classification.
- **Naive Bayes** produced outcomes that were somewhat weaker but balanced, constrained by its independence assumption, which limits its ability to identify the morphological dependencies of Amharic.
- **Random Forest** underperformed, proving that for sparse TF-IDF features, tree-based ensembles perform lower. The modest dataset further limited RF's ability to generalize.

4.5.3. Effect of Feature Selection on Model Performance

To evaluate the impact of feature selection, the SelectKBest method with the Chi-square (χ^2) test was applied after TF-IDF vectorization. Experiments were conducted using different numbers of selected features (k = 1000, 2000, and 3000). The results indicated that reducing the feature size to 1000 and 2000 led to a noticeable decrease in classification accuracy across the models. However, selecting 3000 features maintained performance comparable to the baseline model without feature selection. This suggests that aggressive feature reduction may remove important lexical information necessary for accurate sentiment classification in Amharic text.

4.5.4 Sentiment-Specific Observations

The accurate detection of negative sentiment was a persistent issue with all algorithms. Frequently, the Negative and Neutral classes were misclassified, and the Negative class's recall was consistently lower than that of the Positive and Neutral classes. The following factors may explain this:

1. Negative sentiment in Amharic often relies on context or idiomatic expressions not captured by word-level TF-IDF features.
2. Certain terms occur in both neutral reporting and negative commentary, blurring class boundaries.
3. OCR errors disproportionately affected negative posts, which often used stylistic fonts or graphic backgrounds.

By contrast, Positive sentiment was the easiest to detect. For example, SVM achieved a recall of 0.93 in this class. This success may be attributed to explicit positive markers such as "ደስታ" (happiness), "መልካም" (good), or "አመሰግናለሁ" (thankful), which appear consistently and with less ambiguity.

4.5.5 Influence of Dataset Balance

The 600 clean samples (200 in each class) that made up the final balanced dataset were essential in avoiding bias toward a dominant class. Unlike earlier Amharic sentiment datasets that were skewed toward Neutral sentiment [4,9], this balanced distribution ensured fair learning across all classes. This explains why both Logistic Regression and Naive Bayes achieved relatively uniform performance in Positive and Neutral sentiment detection.

However, the modest dataset size restricted the ability to fully evaluate more complex approaches. A larger dataset, however sufficient for linear models, would enable testing of deep learning methods capable of modeling Amharic's rich morphology and context.

4.5.6 Implications for Amharic NLP

The findings of this study contribute to Amharic NLP in two key ways:

1. **Integration of OCR with sentiment analysis:** The study demonstrates a viable pipeline for the analysis of multimodal Amharic social media data by combining sentiment classification with text extraction from images. This is especially beneficial, considering how frequently sentiment-rich content is shared as images.
2. **Confirmation of linear model effectiveness:** The superior performance of SVM and Logistic Regression reinforces that linear classifiers with TF-IDF remain strong baselines for resource-scarce languages like Amharic. Future researchers can use this understanding to adopt efficient methods when computational resources are limited.

4.5.7 Limitations Observed

Despite good outcomes, several constraints need to be noted:

- **OCR sensitivity to image quality**: Accuracy was limited by OCR performance, with stylized fonts and complex backgrounds causing recognition errors.
- **Morphological richness of Amharic**: Generalization is limited by TF-IDF's inability to adequately capture derivations, inflections, and context-dependent meaning.
- **Dataset size**: The relatively small dataset restricted the exploration of deep learning models, which might otherwise capture richer semantics.

Addressing these limitations will require larger annotated datasets, morphological analyzers, and contextual embeddings like multilingual BERT or AfroLM in future research.

4.6 Summary

The results of the Amharic sentiment analysis system applied to images from social media were presented in this chapter, along with an extensive discussion of the findings. The study addressed the challenge of extracting textual information from Amharic image-based content and classifying it into three sentiment categories, Positive, Negative, and Neutral, through a pipeline that integrated OCR, text preprocessing, TF-IDF feature extraction, and machine learning classifiers.

A total of 1,253 candidate images were initially collected from different social media platforms. Then, 600 clean and representative samples were selected after prescreening. With equal representation, images were manually classified into three sentiment classes, ensuring balanced evaluation. The images were subjected to OCR using Tesseract, supported by OpenCV preprocessing operations (grayscale conversion, binarization, and noise reduction). The extracted text was then normalized by removing punctuation, digits, and stopwords, as well as unifying script variations. Finally, TF-IDF feature representation highlighted sentiment-relevant terms while reducing the influence of common or noisy tokens. Feature selection using SelectKBest method with the Chi-square (χ^2) test was applied after TF-IDF vectorization.

Four classifiers were implemented: Logistic Regression, Naive Bayes, Support Vector Machine (SVM), and Random Forest, and evaluated using accuracy, precision, recall, F1-score, and confusion matrices. The results demonstrated that:

- **SVM** had the best performance, achieving 86% accuracy, with balanced precision, recall, and F1 across all classes. Its ability to detect positive sentiment was especially excellent with 0.93 Recall.
- **Logistic Regression** followed closely, achieving 83% accuracy and robust performance across classes, though with some difficulty in distinguishing Negative sentiment.
- **Naive Bayes** achieved 82% accuracy with consistent but slightly weaker results, reflecting its simplifying independence assumptions.
- **Random Forest** performed the weakest at 75% accuracy, exhibiting low recall despite occasional high precision and having trouble with sparse TF-IDF features.

The comparative analysis demonstrated how linear classifiers are the most dominant in Amharic sentiment analysis, reaffirming findings from previous studies [1,3,4]. Simultaneously, persistent challenge of differentiating Negative from Neutral sentiment highlighted ongoing linguistic and contextual complexities in Amharic. OCR errors contributed to noise, but were largely reduced through preprocessing and TF-IDF weighting.

Generally, the findings of this chapter demonstrate the feasibility of integrating OCR and NLP techniques for Amharic sentiment classification, while also pointing out limitations related to OCR sensitivity, dataset size, and the language's morphological richness.

Chapter Five

5. Conclusion and Recommendations

5.1 Conclusion

This study aimed to develop an integrated system that can extract Amharic text from social media images and perform sentiment classification into Positive, Negative, and Neutral categories. The system effectively showed that a reliable pipeline for Amharic sentiment analysis in image-based contexts can be formed by combining optical character recognition (OCR), text preprocessing, TF-IDF feature extraction, and machine learning classifiers. The study addressed a critical gap in Amharic natural language processing research, where image-embedded texts have been largely overlooked.

With accuracy of 86% and a recall of 0.93 for positive sentiment, the experiment's findings demonstrated that Support Vector Machine (SVM) achieved the best overall performance, followed closely by Logistic Regression (83%) and Naive Bayes (82%). Scoring 75%, Random Forest was behind, underscoring the limitations of tree-based methods when applied to sparse TF-IDF features. The findings reinforce the effectiveness of linear models for morphologically complex languages such as Amharic.

Image preprocessing, particularly grayscale conversion, binarization, and noise removal, proved essential in improving OCR recognition quality, However, challenges with styled fonts and cluttered backgrounds persisted. Class bias was reduced and fair evaluation was guaranteed by the balanced dataset of 600 images (200 per class). Despite the modest dataset size, the study validated the feasibility of applying traditional machine learning methods to Amharic image-based sentiment classification under resource constraints.

Below is a synthesis of how the research questions were addressed:

5.1.1 Research Question 1: How effectively can OCR extract Amharic text from social media images?

using Tesseract OCR engine, supported by OpenCV preprocessing, was effective in extracting Amharic text from diverse image formats. While noise and font variability occasionally caused recognition errors, preprocessing steps significantly improved accuracy, demonstrating the viability of OCR for Amharic social media images.

5.1.2 Research Question 2: What NLP techniques are suitable for analyzing noisy Amharic text from images?

To ensure consistent inputs, stopword removal, script unification, normalization, and punctuation and digit removal were used. TF-IDF feature extraction minimized noise and collected sentiment-relevant phrases, making it appropriate for contexts with limited resources and small datasets.

5.1.3 Research Question 3: How does system performance vary across classifiers using TF-IDF features?

Using the same dataset, four models were compared. SVM achieved the highest overall performance, Logistic Regression and Naive Bayes performed competitively, while Random Forest underperformed. These results indicate that, under the current constraints, linear classifiers remain the most effective choice for Amharic sentiment analysis.

5.1.4 Research Question 4: What metrics best evaluate sentiment classification performance?

F1-score, recall, accuracy, precision, and confusion matrices are the evaluation measures that are employed. Among these, the F1-score was most informative, balancing the trade-off between recall and precision. The evaluation confirmed the system’s reliability in detecting Positive and Neutral sentiments while also revealing persistent difficulties in distinguishing Negative sentiment.

To sum up, this study demonstrates that it is feasible to combine OCR and NLP techniques for Amharic sentiment analysis in social media images, gives a solid baseline for linear classifiers

with TF-IDF, and lays the foundation for advancing Amharic NLP in image-rich online environments.

5.2 Recommendations

Although this research achieved its intended target, several areas remain open for further exploration:

- **Expansion of Dataset:** Future work should focus on collecting larger, more diverse datasets that capture dialectal variations, informal spellings, and wider topical coverage. A larger dataset would also enable the effective application of deep learning approaches.
- **Advanced OCR Integration:** There were issues with the Tesseract engine's ability to handle fancy fonts and noisy backgrounds. To increase extraction accuracy, future studies should investigate transformer-based scene text recognizers, ensemble OCR frameworks, or more advanced OCR systems.
- **Morphological Processing Tools:** Incorporating strong morphological analyzers or embedding models (such as multilingual BERT or AfroLM) could improve sentiment representation due to the complexity of Amharic morphology. Character n-grams should also be explored to capture morphological variations more effectively.
- **Prefix/Suffix Handling:** Future research could revisit prefix and suffix stripping using language-specific morphological methods to better capture root words without losing semantics, even though this study didn't employ it because it can distort the meaning of some words.
- **Improved Negative Sentiment Detection:** One of the biggest challenges that still remains is misclassification of negative and neutral sentiments. For this, domain-specific lexicons, sarcasm-detection mechanisms, and context-aware embeddings are in order.
- **Integration of a Language Identification (LID) Module:** Future iterations of this research could implement a preprocessing layer using **character N-gram frequency** or **unique character detection** to automatically route text to the correct language-specific model.
- **Deployment Considerations:** By optimizing the system, it may be deployed on lightweight platforms such as web based application or mobile applications, increasing the

accessibility of sentiment monitoring for civil society organizations, researchers, and policymakers.

Overall, this thesis contributes a practical and cost-effective baseline for Amharic sentiment analysis from image-based content and highlights a roadmap for future research that bridges OCR, NLP, and sentiment analysis in under-resourced languages.

References

[1] Alemayehu, Fikirte & Meshesha, Million & Abate, Jemal. (2023). Amharic Political Sentiment Analysis Using Deep Learning Approaches. Available at: https://doi.org/10.21203/rs.3.rs-3060010/v1

[2] Gebremichael Tesfagergish, Senait & Damaševičius, Robertas & Kapočiūtė-Dzikienė, Jurgita. (2023). Deep learning-based sentiment classification in Amharic using multi-lingual datasets. Computer Science and Information Systems. 20. 42-42. Available at: https://doi.org/10.2298/CSIS230115042T

[3] Eyob Tesfu, Deep Learning Based Emotion Detection Model for Amharic Text, Master's Thesis, Addis Ababa University, 2021. Available at: https://etd.aau.edu.et/items/ee6a600c-0407-4b65-bb1f-3d87c5ae4555 (Accessed: 10 May 2025).

[4] Seid Muhie Yimam, Hizkiel Mitiku Alemayehu, Abinew Ayele, and Chris Biemann. 2020. Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models. In Proceedings of the 28th International Conference on Computational Linguistics, pages 1048–1060, Barcelona, Spain (Online). International Committee on Computational Linguistics. Available at: https://doi.org/10.18653/v1/2020.coling-main.91

[5] Ebabu, Yitayew & Chalie, Minalu. (2024). Sentiment Analysis for Amharic-English Code-Mixed Sociopolitical Posts Using Deep Learning. Available at: https://doi.org/10.21203/rs.3.rs-4742023/v1

[6] Belete, Mequanent & Kassa, Girma. (2025). Identification of Hateful Amharic Language Memes on Facebook using Deep Learning Algorithms. Systems and Soft Computing. 7. 200258. Available at: https://doi.org/10.1016/j.sasc.2025.200258

[7] Eyasu Tekle, Sentiment Analysis on Amharic Language-Based COVID-19 Discourse from Facebook Social Media Comments, Master's Thesis, St. Mary's University, 2022. Available at: http://repository.smuc.edu.et/handle/123456789/7074

[8] Philemon Wondwossen, and Wondwossen Mulugeta. "A machine learning approach to multi-scale sentiment analysis of amharic online posts." HiLCoE Journal of Computer

Science and Technology 2.2 (2014): 8. Available at: Available at: https://www.academia.edu/25947053

[9] GEBRIEL GIZATE MOLLA, Subjectivity and Sentiment Analysis of Amharic Comments on Social Media: The Case of Ethiopia Political Discourse, Master's Thesis, St. Mary's University, 2020. Available at: http://repository.smuc.edu.et/handle/123456789/6417

[10] Alemneh, Girma Neshir & Rauber, Andreas & Atnafu, Solomon. (2021). Meta-Learner for Amharic Sentiment Classification. Applied Sciences. 11. 8489. Available at: https://doi.org/10.3390/app11188489

[11] B. Gedif, A. Alemu, Y. Assefa and S. Nibret, "Design Amharic Text Sentiment Analysis Model Using Machine Learning Techniques. In Case of Restaurant Reviews," 2023 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), Bahir Dar, Ethiopia, 2023, pp. 150-154. Available at: https://doi.org/10.1109/ICT4DA59526.2023.10302239

[12] Hiwot Wonago Kululo, Information filtering of social media Amharic texts Based on Sentiment Analysis, Master's Thesis, Addis Ababa University, 2020. Available at: https://etd.aau.edu.et/items/4318d1f3-3b7a-4ec8-9e27-063cb103deea (Accessed: 10 May 2025).

[13] Andargachew Mekonnen Gezmu, Binyam Ephrem Seyoum, Michael Gasser, and Andreas Nürnberger. 2018. Contemporary Amharic Corpus: Automatically Morpho-Syntactically Tagged Amharic Corpus. In Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing, pages 65–70, Santa Fe, New Mexico, USA. Association for Computational Linguistics. Available at: https://aclanthology.org/W18-3809/

[14] Dikubab, W., Liang, D., Liao, M., & Bai, X. (2022). Comprehensive benchmark datasets for Amharic scene text detection and recognition. Science China Information Sciences, 65(6), 160106. Available at: https://doi.org/10.1007/s11432-021-3447-9

[15] Samira Gholizadeh (2022). Top Popular Python Libraries in Research. J Robot Auto Res 3(2), 142-145. Available at: https://doi.org/10.33140/JRAR.03.02.02

[16] Malik, U. (2022). Image processing in Open CV. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 10(VI). Available at: https://doi.org/10.22214/ijraset.2022.44527

[17] Smith, R. An overview of the Tesseract OCR engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Parana, Brazil, 23–26 September 2007; Volume 2, pp. 629–633. Available at: https://doi.org/10.1109/ICDAR.2007.4376991

[18] Patel, C.; Patel, A.; Patel, D. Optical character recognition by open source OCR tool tesseract: A case study. Int. J. Comput. Appl. 2012, 55, 50–56. Available at: https://doi.org/10.5120/8794-2784

[19] Sporici, D., Cușnir, E., & Boiangiu, C.-A. (2020). Improving the Accuracy of Tesseract 4.0 OCR Engine Using Convolution-Based Preprocessing. *Symmetry*, *12*(5), 715. Available at: https://doi.org/10.3390/sym12050715

[20] McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 56–61. Available at: https://doi.org/10.25080/Majora-92bf1922-00a

[21] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Müller, A., Grisel, O., … Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *arXiv.* Available at: https://doi.org/10.48550/arXiv.1309.0238

[22] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," in Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, May-June 2007. Available at: https://doi.org/10.1109/MCSE.2007.55

[23] Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021. Available at: https://doi.org/10.21105/joss.03021

[24] Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. Information Processing & Management, 39(1), 45-65. Available at: https://doi.org/10.1016/S0306-4573(02)00021-3

[25] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. Information, 10(4), 150. Available at: https://doi.org/10.3390/info10040150

[26] Dogra, V., Kumar, A., Sachdeva, N., & Sharma, A. (2022). A complete process of text classification system using machine learning and deep learning algorithms. Computational Intelligence and Neuroscience, 2022, 1–12. Available at: https://doi.org/10.1155/2022/1883698

[27] Cortes, C., Vapnik, V. Support-vector networks. Mach Learn 20, 273–297 (1995). Available at: https://doi.org/10.1007/BF00994018

[28] Luo, L., & Li, L. (2014). Defining and evaluating classification algorithm for high-dimensional data based on latent topics. PLOS ONE, 9(7), e102019. Available at: https://doi.org/10.1371/journal.pone.0102019

[29] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). Available at: https://doi.org/10.1023/A:1010933404324

[30] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. Information, 10(4), 150. Available at: https://doi.org/10.3390/info10040150

Appendix

Appendix A: Source code

Following are some parts of the source code that are part of the system development.

1. Model Training; the following code shows training of each model

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
# === Load Training Data from CSV ===
def load_training_data(csv_path):
    df = pd.read_csv(csv_path)
    texts = df['text'].astype(str).tolist()
    labels = df['label'].astype(str).tolist()
    return texts, labels

# === Train Sentiment Models ===
def train_model(train_texts, labels, model_type="logreg"):
    vectorizer = TfidfVectorizer()
    X_train = vectorizer.fit_transform(train_texts)
    if model_type == "logreg":
        model = LogisticRegression(max_iter=1000)
    elif model_type == "naivebayes":
        model = MultinomialNB()
    elif model_type == "svm":
        model = LinearSVC()
    elif model_type == "rf":
        model = RandomForestClassifier(n_estimators=200, random_state=42)
    model.fit(X_train, labels)
    return vectorizer, model
```

2. Image Preprocessing: the following code shows image preprocessing using OpenCV

```
import cv2
# === Preprocess Image ===
def preprocess_image(image_path):
```

```
    image = cv2.imread(image_path)
    if image is None:
        return None
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY +
      cv2.THRESH_OTSU)
    return thresh
```

3. Amharic Text Extraction (OCR) using Tesseract OCR

```
import pytesseract
# === Extract Amharic Text, Removing English ===
def extract_amharic_text(image):
    config = r'--oem 3 --psm 6 -l amh+eng'
    raw_text = pytesseract.image_to_string(image, config=config).strip()
    words = raw_text.split()
    def is_english(word):
        return all('a' <= c.lower() <= 'z' for c in word if c.isalpha())
    amharic_words = [word for word in words if not is_english(word)]
    return ' '.join(amharic_words)
```

4. Text Preprocessing

```
import re
import string
# === Define Amharic Stopwords (expand as needed) ===
amharic_stopwords = set([
    "እኔ", "አንተ", "አንቺ", "እሱ", "እሷ", "እኛ", "እናንተ", "እነሱ",
    "ይህ", "ያ", "ዛ", "ነው", "ናቸው", "ነኝ", "ነህ", "ነሽ", "ነን", "ናችሁ", "ነዋ",
    "እንደ", "በ", "ከ", "ለ", "በላይ", "በታች", "በስተ", "ወደ", "እስከ",
    "እየ", "እንዲ", "እንጂ", "ያለ", "የለ", "ነበር", "ነበሩ", "ነበረ",
    "አንድ" , "ሌላ", "ማን", "ምን" , "ጊዜ",
    "እባክህ", "እባክሽ", "እባካችሁ", "አዎን", "አይደለም", "አዎ", "በኩል"
])

# === Text Preprocessing Function with Conservative Normalization ===
def preprocess_text(text):
    normalization_map = {
        'ሠ': 'ሰ', 'ሡ': 'ሱ', 'ሢ': 'ሲ', 'ሣ': 'ሳ', 'ሤ': 'ሴ', 'ሥ': 'ስ', 'ሦ':
'ሶ',
        'ጸ': 'ፀ', 'ጹ': 'ፁ', 'ጺ': 'ፂ', 'ጻ': 'ፃ', 'ጼ': 'ፄ', 'ጽ': 'ፅ', 'ጾ':
'ፆ',
```

```
        'ሐ': 'ሀ', 'ሑ': 'ሁ', 'ሒ': 'ሂ', 'ሓ': 'ሃ', 'ሔ': 'ሄ', 'ሕ': 'ህ', 'ሖ': 'ሆ',
        'ኀ': 'ሀ', 'ኁ': 'ሁ', 'ኂ': 'ሂ', 'ኃ': 'ሃ', 'ኄ': 'ሄ', 'ኅ': 'ህ', 'ኆ': 'ሆ'
    }
    for old_char, new_char in normalization_map.items():
        text = text.replace(old_char, new_char)

    amharic_punctuations = '።፡፣፤፥፦፧፨"“”‘’«»‹›'
    geez_numerals = '፩፪፫፬፭፮፯፰፱፲፳፴፵፶፷፸፹፺፻፼'
    all_punctuations = string.punctuation + amharic_punctuations +
geez_numerals
    text = text.translate(str.maketrans('', '', all_punctuations))
    text = re.sub(r'\d+', '', text)
    text = text.replace('\n', ' ').replace('\r', ' ')
    text = re.sub(r'\s+', ' ', text).strip()
    words = text.split()
    words = [word for word in words if len(word) > 1]
    filtered_words = [word for word in words if word not in
amharic_stopwords]
    return ' '.join(filtered_words)
```

5. Sentiment Prediction

```
# === Predict Sentiment ===
def predict_sentiment(text, vectorizer, model):
    X_test = vectorizer.transform([text])
    return model.predict(X_test)[0]
```

6. Model Evaluation: following code shows evaluation and comparison of models using evaluation metrics

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import (
    classification_report, accuracy_score, precision_score,
    recall_score, f1_score, confusion_matrix
)
```

```
# === Extended Evaluation (compare multiple models + visualize) ===
def evaluate_models(test_csv, results_dict):
    test_df = pd.read_csv(test_csv)
    y_true = test_df['label'].tolist()
    labels = ["Positive", "Negative", "Neutral"]

    summary = []
    confusion_matrices = {}

    for model_name, results_csv in results_dict.items():
        results_df = pd.read_csv(results_csv)
        y_pred = results_df['Predicted_Sentiment'].tolist()

        acc = accuracy_score(y_true, y_pred)
        prec = precision_score(y_true, y_pred, average='macro',
zero_division=0)
        rec = recall_score(y_true, y_pred, average='macro', zero_division=0)
        f1 = f1_score(y_true, y_pred, average='macro', zero_division=0)

        summary.append({"Model": model_name, "Accuracy": acc, "Precision":
prec, "Recall": rec, "F1": f1})

        # Store confusion matrix
        cm = confusion_matrix(y_true, y_pred, labels=labels)
        confusion_matrices[model_name] = cm

        # Print report
        print(f"\n=== {model_name} Report ===")
        print(classification_report(y_true, y_pred, target_names=labels))

    # Comparison table
    summary_df = pd.DataFrame(summary)
    print("\n Model Comparison Summary:")
    print(summary_df)

    # === Confusion Matrices (save as PNGs) ===
    fig, axes = plt.subplots(2, 2, figsize=(12, 10))
    axes = axes.flatten()
    for i, (model_name, cm) in enumerate(confusion_matrices.items()):
        sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
xticklabels=labels, yticklabels=labels, ax=axes[i])
        axes[i].set_title(f"{model_name} Confusion Matrix")
        axes[i].set_xlabel("Predicted Label")
        axes[i].set_ylabel("True Label")
```

```
    plt.suptitle("Confusion Matrices of Sentiment Classification Models",
fontsize=14, weight="bold")
    plt.tight_layout(rect=[0, 0, 1, 0.96])
    plt.savefig("plots/confusion_matrices.png")
    plt.close()

    # === Bar Chart (Accuracy, Precision, Recall, F1) ===
    summary_melted = summary_df.melt(id_vars="Model", var_name="Metric",
value_name="Score")
    plt.figure(figsize=(10, 6))
    sns.barplot(x="Model", y="Score", hue="Metric", data=summary_melted)
    plt.title("Performance Comparison of Sentiment Classification Models")
    plt.ylim(0, 1)
    plt.ylabel("Score")
    plt.xlabel("Models")
    plt.legend(loc="lower right")
    plt.savefig("plots/model_comparison.png")
    plt.close()

    return summary_df
```

Appendix B: A Machine Learning Framework for Amharic Sentiment Analysis in Social Media Images Using OCR and NLP Techniques

Algorithm Steps

I. Step 1: Dataset Collection and Annotation

- Source: Amharic social media images containing user-generated text were collected manually.
- Annotation: Extracted texts were manually labeled into three sentiment categories: Positive, Negative, and Neutral.
- Dataset size: A total of 600 images were prepared, with a balanced distribution of 200 samples per sentiment class.
- Split: The dataset was divided into training (80%) and testing (20%) sets.

II. Step 2: Image Preprocessing and OCR

- Image Enhancement: Preprocessing was performed using OpenCV (grayscale conversion, binarization, and noise removal).
- Text Extraction: Amharic texts were extracted from images using Tesseract OCR.
- Output: Extracted raw Amharic text was stored in a structured CSV file with corresponding sentiment labels.

III. Step 3: Text Preprocessing

- Cleaning: Removal of punctuation, digits, and Geʿez numerals.
- Normalization: Character unification (e.g., $ሠ \rightarrow ሰ$, $ጸ \rightarrow ፀ$) to reduce script variations.
- Stopword Removal: Elimination of frequently occurring but sentiment-irrelevant Amharic words.
- Consistency: Both training and test datasets underwent identical preprocessing steps.

IV. Step 4: Feature Extraction

- Technique: Term Frequency–Inverse Document Frequency (TF-IDF) was used to convert preprocessed text into numerical feature vectors.
- Representation: Each document was transformed into a sparse matrix reflecting word importance across the dataset.

V. Step 5: Model Training and Evaluation

- Classifiers Implemented: Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), and Random Forest (RF).

- Evaluation Metrics: Accuracy, Precision, Recall, F1-score, and Confusion Matrices were used for model comparison.

Hyperparameter Configuration (Selected Model – SVM)

- Kernel: Linear
- Regularization Parameter (C): 1.0
- Feature Representation: TF-IDF (unigram features)
- Evaluation: 80/20 train-test split, stratified by sentiment labels