Addis Ababa University

College of Natural and Computational Science

School of Information Science

Amharic Language Criminals Keyword Spotting Using Deep Learning Model

By Abel Ademe (GSE/8065/11)

A Thesis Submitted to Addis Ababa University, School of Graduate Studies in Partial Fulfillment for the Degree of Master of Science in Information Science.

Nov 2023

Addis Ababa, Ethiopia

Addis Ababa University

College of Natural and Computational Science

School of Information Science

Amharic Language Keyword Spotting Using Deep Learning Model

By Abel Ademe (GSE/8065/11)

Signed by the Examining Committee:

Advisor:	Michael Melese (Ph.D.)	_____	_____
	Name	Signature	Date
Internal-Examiner:	Million Meshesha (Ph.D.)	_____	_____
	Name	Signature	Date
Internal-Examiner:	Solomon Teferra (Ph.D.)	_____	_____
	Name	Signature	Date

Declaration

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

Declared by: Abel Ademe _____ _____

Name Signature Date

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

Confirmed by advisors:

Main advisor: Michael Melese (Ph.D.) _____ _____

Name Signature Date

Acknowledgment

Primarily, I would like to pass my gratitude to the Almighty of God for giving me the strength and patience in completing this thesis; the journey would be difficult without his support. Next, I am grateful to my advisor, Dr. Michael Melese (Ass. Prof) for his continuous support and guidance throughout the duration of this thesis. His unreserved support starts to end of the study was creditable.

There are also a lot of individuals who helped me along the way in my journey of completing this thesis work. Some of the individuals are those who provide me their voice record for several times without losing their patience and complain during the sample speech collection time.

Additionally, I would also like to pass my thanks to my wife for her understanding the situation I am in and her support in handling all the family workloads. Finally, my gratitude should also go to my friends, especially Adane and Ephrem, my parents, and my siblings. I am very grateful for your patience and understanding.

Abstract

The task of automatic speech recognition (ASR) presents a significant challenge for many low-resource languages. Amharic is one of the Afro-Asiatic languages spoken in Ethiopia, and it is one of the languages that require special attention for speech recognition research. Amharic has distinct phonetics and supra-segmental features that make it distinct from other languages. Therefore, developing an accurate and efficient speech recognition system for Amharic requires specialized techniques.

In this thesis, we propose an Amharic language keyword spotting based on the integration of the Gramian Angular Field (GAF) representation with Convolutional Neural Network-Long Short-Term Memory Network (CNN-LSTM) architecture. Amharic Speech Commands (ASC) dataset is initially prepared. This ASC is prepared from a comprised of 30 individuals. A diverse set of 40 criminal keywords are selected and each selected criminal word is recorded 10 times from a single person. Ultimately a comprehensive dataset totaling of 12,000 audio files is prepared. The GAF transformation converts raw Amharic audio signals into visual representations, capturing both temporal and spatial patterns inherent in the data. The CNN-LSTM architecture combines the power of convolutional neural networks in learning spatial features from GAF images with the ability of LSTM networks to capture temporal dependencies in sequential data. To evaluate the effectiveness of the proposed model, we compiled a carefully annotated Amharic keyword dataset and conducted a systematic search for optimal hyperparameters. The model demonstrated 96.73%, and 89.98%, training and testing accuracy respectively.

In conclusion, the combination of GAF-based representation with the CNN-LSTM model showcases the potential of deep learning techniques in tackling the unique challenges of Amharic language keyword spotting. However, this study does not leverage transfer learning and pre-training approaches on larger multilingual datasets. Future researchers will actively contribute to the advancement of speech technology, specifically focusing on keyword spotting for diverse and large linguistic communities.

Keywords: Amharic Language, Keyword Spotting, Gramian Angular Field (GAF), Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM)

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ASC	Amharic Speech Command Dataset
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep- Neural Network
DSCNN	Depth-wise Convolutional Neural Network
DSP	Digital Signal Processing
FC	Fully Connected
FE	Feature Extraction
FFT	Fast Fourier Transform
GAF	Gramian Angular Field
GADF	Gramian Angular Difference Field
GASF	Gramian Angular Summation Field
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HMMs	Hidden Markov Models
KWS	Keyword Spotting
LSTMs	Long-Short Term Memory
MFCCs	Mel Frequency Cepstral Coefficients
MOPs	Millions Of Operations Per Second
NLP	Natural Language Processing

NLQ	Nonlinear Quantization
OOV	Out Of Vocabulary
PG	Power Gating
ReLu	Rectifier Linear Unit
RNN	Recurrent Neural Network
SD	Sound Detection
SIMD	Single Instruction Multiple Data
SNR	Signal To Noise Ratio
SV	Speaker Verification
TCN	Temporal Convolutional Network
TD	Temporal Dynamics
TPR	True Positive Rate

Chapter One

Introduction

1.1 Background

Nowadays, Artificial Intelligence (AI) is the driving force guiding and shaping our modern society. Technological changes can be seen in a wide variety of areas which include healthcare, education, entertainment, enterprise, and so on. Some examples are computerized medical diagnosis, autonomous robotics for industrial environments, advice systems for business, prediction of forest wildfires, interactive intelligence for computer video games, etc. The list of applications is countless, and the restrictions determined with the aid of technology are continuously moving [1][2].

The rise of AI can be seen in the prevalence of virtual assistants that have made voice a common medium of interaction among users and their devices. Industrial examples of this kind of human-computer interface may be determined in applications like Siri, Cortana, or Amazon Echo [2]. In addition to this area, AI also plays a vital role in advanced security systems, which extend beyond honoring complete speech inputs. In specific operations, there's a need to find and separate keywords said by an individual, such as criminal word identification, benefiting law enforcement and security efforts significantly. This type of task of detecting a specific word or phrase from audio signals is in general called Keyword Spotting (KWS).

Throughout the generations, colorful methods have been researched for KWS. One of the foremost techniques involves the utilization of large vocabulary continuous speech recognition (LVCSR) systems [2-9]. These systems are utilized to decode the audio signal and search for the desired keyword within the generated lattices. An advantage of this approach is its rigidity in handling dynamic or undetermined keywords. However, a notable drawback of LVCSR- based KWS systems lies in their computational complexity, rendering them unsuitable for modern KWS applications designed for compact digital devices like smartphones, smart speakers, and wearables.

An enduringly attractive and lighter alternative to LVCSR is the keyword/filler hidden Markov model (HMM) method, which was introduced approximately three decades ago [12-14]. These days, deep neural network-based techniques have replaced LVCSR and HMMs due to the regular

advanced overall performance. In summary, it can be affirmed that deep-learning-based KWS is currently a highly active and hot research area [2].

Given the latest progress in deep learning methodologies, there have been significant improvements in the accuracy of keyword spotting systems. Convolutional Neural Networks (CNNs) have been shown to work well on image classification tasks [9], and Recurrent Neural Networks (RNNs) have been shown to work well on sequential data. The combination of these two networks, called Convolutional Recurrent Neural Networks (CRNNs), has been shown to achieve state-of-the-art performance in various speech and language tasks such as speech recognition and keyword spotting [9], [19]. Additionally, innovative representations such as the Gramian Angular Field (GAF) transformation have emerged as powerful techniques for converting time series data, such as audio waveforms, into visual representations that can be effective through a deep learning technique [32].

Despite the remarkable progress in KWS, most research efforts have focused on widely spoken languages, such as English, leaving under-resourced languages, including Amharic, with limited attention [2]. Amharic serves as the official working language for the Ethiopian government, a nation located in East Africa, boasting a population exceeding one hundred seven million [8]. This language is part of the Ethio-Semitic language group, falling within the Semitic branch of the Afro-Asiatic language family [15-17]. Amharic holds the distinction of having the world's second-largest number of speakers among Semitic languages, surpassed only by Arabic, and it stands as the most widely spoken Semitic language within Ethiopia [15].

People and companies often use audio files to fulfill their information needs [16]. Locating a specific spoken phrase within an audio data poses difficulty, as users may be aware that a specific word was spoken but might not know precisely where in the audio file it was uttered. For example, when searching for the spoken keyword 'ኢትዮጵያ' (Ethiopia) within an audio file of duration 'n', users might need to listen to the entire audio recording or make educated guesses about the word's location within the file [24]. Consequently, automatically identifying a particular spoken word or phrase within a given audio file presents a challenge for languages spoken in Ethiopia, particularly Amharic, which possesses distinct linguistic characteristics. Particularly, the presence of glottal, palatal, and labialized consonants sets the Amharic language apart from others [17].

Moreover, Amharic language poses unique challenges for KWS due to its complexity of the morphology and richness characteristics. As a result, it is not feasible to directly use methods applied to other languages. Thus, addressing the need for an accurate and contextually aware KWS system for Amharic language is essential to unlock its potential in various applications, promoting linguistic diversity and inclusivity in speech technology.

1.2 Statement of the Problem

Significant quantities of speech data are continuously accumulating in various languages due to the advancement of machine processing capabilities and increased storage capacity [2]. At this time, KWS is the best promising technology for natural language processing (NLP) in voice recognition. KWS is the recognition of spoken keywords in audio streams and has become a rapidly emerging technology [33]. Along with this, different studies try to develop KWS model for the widely spoken language in developed countries using different methodology [2].

In this thesis, the research problem focuses on the detection of Amharic criminal words within speech data. Amharic, being the official language of Ethiopia, plays a crucial role in communication and discourse among the Ethiopian population. However, identifying and detecting criminal words or phrases within spoken Amharic poses a significant challenge [34]. Criminal words can include expressions related to illegal activities, hate speech, violence, or any language that violates legal and social norms [17].

Existing KWS techniques primarily focus on languages such as English, and there is a significant gap in research related to detecting criminal words specifically in Amharic. Traditional techniques that rely on keyword dictionaries or rule-based approaches are often inadequate to handle the complexity and nuances of the Amharic language, resulting in low detection accuracy and high false-positive rates.

Hence, the aim of this research is to design an innovative approach that leverages advanced deep learning architectures, to effectively detect the presence of criminal words in Amharic speech. To this end, the following research questions are investigated and answered by this study.

1. Which hyperparameters are best for the spotting of Amharic Keywords?
2. What is the effectiveness of the developed model in detecting the criminal Amharic keywords?

1.3 Objective of the Thesis

1.3.1 General Objective

The general objective of the study is to design and develop Amharic Language Keyword Spotting based on GAF based CNN-LSTM Model for selected criminal Amharic words.

1.3.2 Specific Objectives

To attain the general objective of the study, the following specific objectives of this thesis are listed below:

- To perform a literature review to understand the state-of-the-art in speech recognition, KWS and Amharic language.
- To collect and prepare Amharic criminal keyword audio dataset.
- To apply appropriate preprocessing and feature extraction techniques.
- To design the general architecture of Amharic KWS
- To develop a deep learning model that combines the GAF representation and CNN-LSTM for keyword spotting.
- To evaluate the developed CNN-LSTM model using the unseen GAF-transformed Amharic audio dataset.

1.4 Methodology of the Study

Methodology is a means of achieving the goals of research problems. Literature on the KWS system, dataset preparation, data preprocessing, deep learning approaches for solving the challenge of KWS can be found in books, journals, articles, conference proceedings, and the Internet. In addition, prior and current research efforts on several KWS system designing methods and these various techniques were evaluated in order to gain a better understanding of the best performing algorithms and methodologies in terms of empirical models and machine learning models. Since this study is meant to be a continuation of prior studies, it will need to study Amharic language KWS system through GAF audio speech image representation and combination of CNN and LSTM. Therefore, in order to undertake and achieve the objective of this research the following methods and techniques are used.

1.4.1 Research Design

This research adopts an experimental research approach as its foundational methodology in order to accomplish the overarching objectives of this thesis. Experimental research, which falls within the domain of scientific research strategies, is characterized by its rigorous investigation into the relationships between two variables: the dependent and independent variables. The fundamental aim of experimental research is to discern and elucidate connections between a distinct attribute of a subject and the variable that is the subject of investigation [37]. Consequently, within the context of this thesis, the research undertaking is focused on the assessment of the presence or absence of keywords associated with criminal activities. Through systematic experimentation, this study endeavors to determine the availability of criminal words.

The research design for KWS using the GAF based CNN-LSTM architecture involves a systematic approach for developing and evaluating the proposed model. To accomplish this task, the study uses a procedural framework, as depicted in Figure 1.1, consisting of three primary steps. In the initial step, the research delves into the domain of the problem at hand, enhancing comprehension through an extensive review of diverse literature sources. Subsequently, the study formulates both the general and specific objectives of the thesis. The rest of the research design is generalized into three main phases.

Here is a detailed outline of the research design.

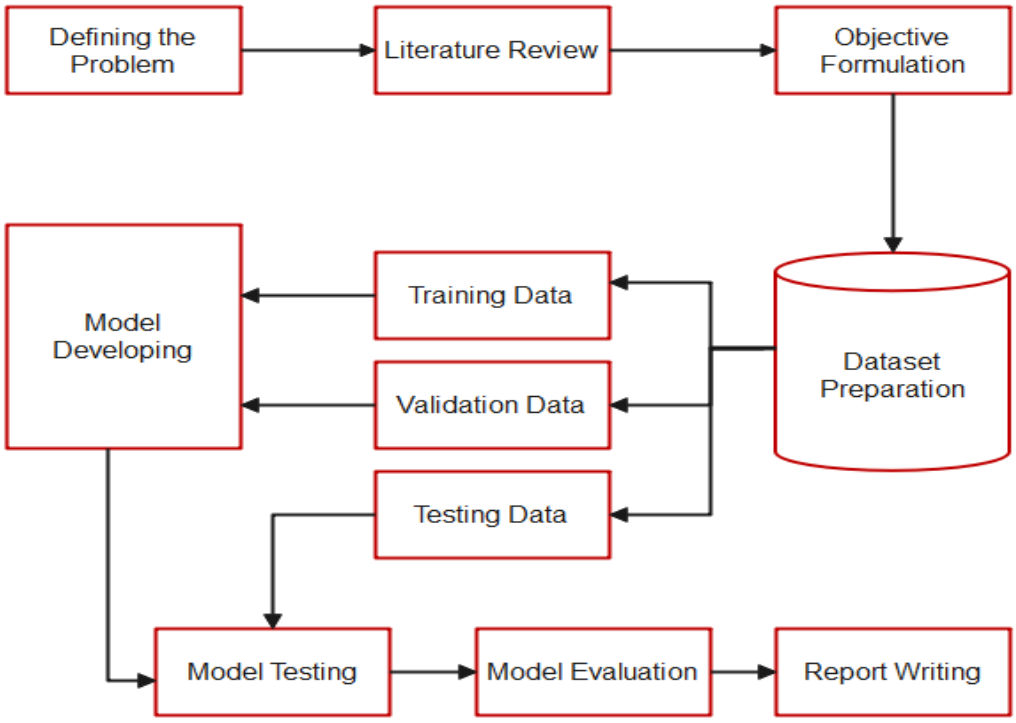


Figure 1.1 Steps followed in conducting this Research.

Phase 1: Data Preparation and Model Architecture Design

In this initial phase, the focus is on preparing the necessary data and apply the developed of the CNN-LSTM model:

Data Collection and Preprocessing:

- ✓ Gather a diverse dataset of Amharic audio recordings containing both keywords and background noise.
- ✓ Segment, resample, and clean the audio data to ensure consistency and quality.

GAF Image Generation:

- ✓ Transform the preprocessed audio data into GAF images.
- ✓ Experiment with image dimensions and representations to find the most suitable format.

Develop CNN-LSTM Model:

- ✓ Develop a CNN architecture tailored for processing GAF images.

- ✓ Integrate LSTM layers to capture temporal dependencies within the GAF-based features.

Phase 2: Model Training and Optimization

In this phase, the focus shifts to training and optimizing the CNN-LSTM model using the prepared data:

Dataset Splitting:

- ✓ Partition the prepared dataset into training, validation, and testing subsets, ensuring that recordings from individual contributors are exclusively present in one subset.

Model Training and Hyperparameter Tuning:

- ✓ Train the CNN-LSTM model using the training subset and monitor its performance on the validation subset.
- ✓ Experiment with various hyperparameters (learning rate, batch size, number of layers) to fine-tune the model.

Validation and Overfitting Analysis:

- ✓ Evaluate the model's performance on the validation and testing dataset, using a commonly used performance evaluation parameter.
- ✓ Adjust hyperparameters to prevent overfitting while achieving optimal validation results.

Phase 3: Evaluation and Interpretation of Results

The final phase involves evaluating the trained model's performance, interpreting, and concluding the findings:

Model Testing and Generalization:

- ✓ Test the optimized CNN-LSTM model on the hold-out testing subset to assess testing performance.

Interpretation and Discussion of Results:

- ✓ Interpret the model's performance in the context of Amharic language processing and keyword spotting tasks.
- ✓ Discuss the impact of GAF representations and the CNN-LSTM architecture on keyword detection accuracy and efficiency.

Conclusion and Future Work:

- ✓ Summarize the research findings and contributions of the developed KWS model using GAF based on CNN-LSTM.
- ✓ Provide insights into potential areas for future research and improvements in the proposed architecture.

These three phases encapsulate the systematic approach to developing, optimizing, and evaluating the KWS model using GAF representations and a CNN-LSTM architecture, with the ultimate goal of advancing keyword spotting techniques for the Amharic language.

1.4.2 Dataset Collection

By having access to an appropriate dataset, various models can be trained to identify a predefined set of application-specific keywords. For the construction of the Amharic Speech Command (ASC) dataset, a dataset having 12,000 one-second-long audios of 40 criminal keyword recorded from 30 different person, the primary task is first to identify and select the Amharic criminal keywords by reviewing kinds of literature and advising legal experts. The prepared dataset is comprised of a diverse set of 40 criminal keywords. This prepared audio data is collected through recording. Each audio file within the dataset has been uniformly structured to possess a duration of one second, sampled at a rate of 16kHz. To compile this dataset, we engaged the participation of 30 individuals, each of whom contributed 10 distinct utterances for every keyword. Consequently, this results in a total of 300 audio files for each individual keyword, ultimately with a dataset totaling of $(30 \times 10 \times 40 = 12{,}000)$ audio files. The collective size of this dataset is 1.64 GB, facilitating a foundation for the subsequent phases of our research and experimentation.

1.4.3 Tools and Techniques

The tools used in this research are Python 3.6 using Pycharm editor, OpenCV library for image processing cases, NumPy and Pandas libraries [19] are used for data preprocessing and feature extraction purposes. We use the TensorFlow and Keras libraries [20] to implement deep learning models.

1.4.4 Evaluation methods

When evaluating the proposed deep learning model for keyword spotting in the Amharic language, several evaluation methods can be applied. Among them, the two primary metrics play a pivotal

role: accuracy and loss. Training accuracy provides insight into how effectively the model is learning the features of the training dataset, measuring the proportion of correctly classified instances during the training process. Meanwhile, testing accuracy offers an assessment of the model's ability to generalize to unseen data by measuring its performance on a separate testing dataset. These metrics collectively assess the model's overall classification accuracy and its capacity to correctly identify keywords in Amharic utterances.

In addition to accuracy, the evaluation also depends on loss metrics, including training loss and testing loss. Training loss quantifies the error incurred by the model during the training phase, reflecting how well it converges towards optimal parameter values. Lower training loss indicates a more effective learning process. On the other hand, testing loss measures the discrepancy between the model's predictions and the actual values in the testing dataset. Monitoring testing loss aids in discerning potential overfitting or underfitting issues, thus ensuring the model's robustness and generalization capability beyond the training data. By examining both accuracy and loss metrics, the performance of the developed deep learning model performance in keyword will be evaluated, particularly in the context of the Amharic language.

1.5 Scope and Limitation of the Study

The scope of this thesis focuses on developing a new method for detecting keywords in speech data using CNN-LSTM architecture that incorporates GAF-based features. The study focuses on reviewing the literatures, preparing ASC dataset, preprocessing the prepared audio dataset, generating GAF images GAF based feature extraction, developing the deep learning, evaluate the proposed model and documenting the findings, aiming to enhance the performance of KWS based on the developed deep learning model. However, it is important to acknowledge certain limitations. The study constrained by the availability and representativeness of training data, and variations in spoken language characteristics across different dialects or accents which can have impact on the generalizability of the model. The study uses only the prepared dataset which is the dataset having 12,000 audio files. Moreover, the prepared dataset is only a collection of 40 criminal keywords.

1.6 Significance of the Thesis

The intended contributions of this thesis can be summarized as follows:

- Development of a new approach to keyword spotting that integrates the power of convolutional and recurrent neural networks with GAF-based features, showcasing the ability to capture the temporal and spectral characteristics of the signal.
- Development of ASC dataset that consists of selected Amharic keyword. Preparing such datasets offers significant benefits to the machine learning and deep learning community.
- Thorough experimental assessment of the proposed method on the prepared ASC dataset, showcasing its efficacy in detecting keywords of interest at high accuracy levels, thus demonstrating the developed approach that can lead to improved performance in keyword spotting tasks.
- Identification of key parameters that affect the effectiveness of the proposed approach's performance and provide insights into the extent of their impact.
- Finally, identification of possible directions for future research that could build on the proposed method and improve the current state-of-the-art.

In summary, the thesis contributes an effective approach to Amharic keyword spotting and potential applications in several areas, including non-invasive home automation, speaker identification, voice queries, etc.

1.7 Organization of the Thesis

In this subsection, the outline of each chapter of this thesis is given. There are six chapters in total, starting from an introduction of the thesis and ending with conclusion and recommendation. The first chapter presents the introduction of the research, statement of the problems, objectives, and scope of the research. It also provides the main contribution and outlines of the thesis. The rest of the thesis is organized as follows:

Chapter two presents some related technical background knowledge of the thesis, including the fundamental concept of KWS, designing and implementation approach of KWS, Deep learning model, introduction about CNN and LSTM and Amharic language description. It also provides a literature review. Chapter three presents the research design of this thesis, description of dataset preparation and explains the evaluation metrics of the proposed model. Chapter four explores

proposed model called KWS based on GAF based CNN-LSTM Model. It clearly mentions how the proposed model is developed. It further explains the data preprocessing of the prepared dataset. Chapter five presents the simulation result of the proposed model including discussion with comparisons of some other related schemes. The last part, which is chapter six, provides the conclusion of the proposed thesis. Future studies are also mentioned in the last chapter.

Chapter Two

Literature Review

2.1 Overview

This chapter presents a comprehensive literature review of existing research and scholarly works related to the field of KWS. The review begins by establishing the foundational concepts of KWS, highlighting key applications, designing methodologies, and challenges encountered in previous studies. It then systematically explores the existing literature on CNNs, and LSTMs, providing insights into their individual contributions to the field. Furthermore, the literature review critically examines studies that intersect KWS with CNN, and LSTM architectures, assessing their methodologies, findings, and limitations. By synthesizing the collective knowledge from these diverse sources, this section aims to contextualize the current study within the broader landscape of KWS research while identifying gaps and opportunities for further research.

2.2 Keyword Spotting and its Applications

Speech Recognition falls under the subfield of Natural Language Processing, concentrating on the comprehension of spoken language [3]. This involves translating auditory input into corresponding words within a specific language's vocabulary. KWS is often integrated as a specialized application within broader speech recognition systems, serving in scenarios where the detection of particular trigger words or phrases is essential, such as criminal words detection. Keyword spotting, originally introduced in the context of speech processing, revolves around the detection of specific keywords within spoken utterances.

Keyword spotting is a pivotal component of speech recognition systems, enabling efficient and accurate detection of specific keywords or phrases within a continuous audio stream. The primary objective of KWS is to identify and localize predefined target words, commands, or prompts, allowing for seamless interaction with voice-activated applications and voice assistants [3]. Beyond traditional speech recognition tasks, KWS finds extensive applications in various domains, enhancing user experience and enabling novel functionalities.

2.2.1 Voice-Activated Systems and Voice Assistants

One of the most prominent applications of Keyword Spotting (KWS) is in voice-activated systems and voice assistants, revolutionizing the way users interact with various devices and applications [2]. Voice assistants, such as Siri, Google Assistant, or Amazon Alexa, have become an integral part of daily life, assisting users in tasks ranging from setting reminders and managing schedules to providing real-time weather updates and answering general knowledge queries. Users can effortlessly trigger these voice assistants by uttering a wake-up keyword or command, such as "Hey Siri," "OK Google," or "Alexa." Upon recognition of the wake-up phrase by the KWS module, the voice assistant becomes actively attentive, awaiting the user's subsequent request [31].

Accurate and efficient KWS is of paramount importance in this context, as any delays or misinterpretations may lead to frustrating user experiences and hinder seamless interactions. The ability of KWS to promptly and accurately identify the wake-up keywords enables voice assistants to respond swiftly and provide relevant information or perform desired actions, enhancing user satisfaction and overall usability. This application of KWS exemplifies the fusion of speech recognition and natural language understanding, making voice assistants an indispensable part of modern technology and communication.

2.2.2 Voice Search and Voice Command Applications

Moreover, KWS plays a significant role in enabling voice search functionalities across a wide range of applications, transforming the way users access information and interact with technology search [2]. In voice search applications, users can initiate searches by simply speaking a specific keyword or phrase, and the system promptly processes the spoken query, retrieving relevant information or search results from vast databases or online resources. This seamless and intuitive approach to search empowers users to access information hands-free, making it particularly valuable in scenarios where manual input might be inconvenient or unsafe, such as while driving or performing physical tasks [2].

Similarly, in voice command applications, KWS empowers users to interact with smart devices, home automation systems, and Internet of Things (IoT) devices effortlessly. Users can control a myriad of functions, such as adjusting smart thermostats, turning on or off lights, setting reminders, or even ordering products online, solely by issuing voice commands. The integration of KWS into these applications significantly enhances user experiences, making technology more accessible

and user-friendly while reducing the reliance on traditional manual input methods. As voice interaction becomes increasingly pervasive, KWS stands at the forefront, unlocking new levels of convenience and efficiency in how users engage with technology in their daily lives [33].

2.2.3 Transcription Services and Content Indexing

Furthermore, KWS finds valuable applications in transcription services and content indexing, revolutionizing how audio data is organized, searched, and retrieved. In transcription services, KWS algorithms are employed to automatically identify and segment specific topics or keywords within extensive audio datasets, including lectures, interviews, podcasts, and conference recordings. By accurately detecting and indexing keywords or phrases, the transcription process becomes more efficient, enabling users to pinpoint specific content of interest without the need to listen to the entire recording [33]. This enhanced indexing capability streamlines the retrieval of relevant information, saving valuable time and effort for researchers, journalists, and anyone working with large audio datasets. Additionally, content indexing using KWS technology significantly improves the accessibility and navigation of spoken documents, making it easier for users to locate and review critical sections of the audio, fostering a more comprehensive understanding of the material.

2.2.4 Automated Detection of Criminal Words

KWS finds crucial applications in criminal word identification, benefiting law enforcement and security efforts significantly. By automatically detecting specific keywords or phrases associated with criminal activities, KWS serves as an essential tool for automated threat detection and monitoring, enabling swift responses to potential security breaches or unlawful actions. Moreover, in criminal investigations, KWS aids in forensic speech analysis and evidence collection, efficiently extracting relevant information from recorded audio evidence. It also plays a pivotal role in identifying criminal networks, suspects, and extremist elements by analyzing audio data for incriminating keywords, assisting in intelligence gathering and tracking down criminal elements. Additionally, KWS contributes to monitoring illicit activities, detecting threats and violent extremism on social media platforms, and serves as an early warning system for public safety in critical areas. These applications highlight the significance of KWS in empowering law enforcement agencies with timely and accurate insights for criminal word identification and enhancing overall public safety [34][35].

2.3 KWS System Designing Methodologies

The primary aim of a KWS system is to identify presence of predefined keywords within an input spoken utterance [3]. To achieve this, the utterance can be initially converted into text, segmented into phonemes, or directly processed. This leads to three primary design approaches for KWS systems: LVSCR-based KWS, Acoustic based KWS, and Phonetic Search based KWS. The next subsections involve a discussion and comparison of these methods.

2.3.1 LVCSR based KWS System

This approach involves the conversion of the input speech utterance into text, followed by the utilization of a text search algorithm to identify the keywords present in the utterance [3], [18], [27]. Consequently, this initial approach is referred to as "LVCSR-based KWS". LVCSR-based KWS is a two-stage process. In the first stage, a large vocabulary speech recognizer transcribes extensive audio files into phoneme or phrase lattices. In the second stage, a lattice-based search is conducted to identify the target set of keywords. The first stage of LVCSR-based KWS is conducted offline, while the second stage operates in real-time. However, this method exhibits three significant drawbacks. Firstly, it necessitates a substantial number of labeled records for training LVCSR-based KWS systems. Secondly, the computational demands associated with large vocabulary decoding are excessively high. Lastly, it tends to underperform when dealing with Out of Vocabulary (OOV) words.

2.3.2 Acoustic based KWS System

In contrast to LVCSR, the second approach, the acoustic KWS system, does not attempt to estimate the entire text of the audio. Instead, it employs the same search algorithm. Rather than having a single extensive acoustic model trained using representative data, this approach relies on performing speech recognition using a smaller subset of specific words alongside a general non-keyword model. By having distinct models for keywords and for other commonly termed 'garbage' words, the acoustic KWS system can conduct its search in a single step [10-12]. These systems utilize techniques like HMMs, DNNs, or CNNs to map audio features to phonemes or sub-word units. By modeling the acoustic characteristics of keywords, the system can efficiently detect them within the audio stream.

In reality, Google [13] uses the same approach to train the trigger word "Okey Google" using DNN. Instead of utilizing multiple models, they train their neural network by including numerous instances of the trigger word along with another training set categorized as “filler” [19].

2.3.3 Phonetics based KWS System

Phonetic Search KWS involves the identification of keywords based on their phonetic transcriptions, which is accomplished through a two-stage process. Phoneme decoding is performed in the initial stage, converting the audio input into an array of phonemes, as opposed to LVCSR, which produces a list of words [2]. The system utilizes the generated phoneme dictionary to compare the user's search keyword to the correct phonetic representation in the last stage. It's advantageous to note that this approach necessitates a keyword list with their corresponding phonetic descriptions, along with a comprehensive phoneme database. This approach combines the benefits of LVCSR-based KWS and acoustic KWS [13], [15].

2.4 Keyword Spotting (KWS) Approaches

Irrespective of the design methodologies applied to address the KWS problem, it is essential to identify an appropriate implementation approach for training KWS systems. To achieve this, diverse approaches have been developed in recent years. Some of these approaches are described in the next subsections.

2.4.1 Template Matching for Keyword Spotting

In the realm of KWS, the Template matching based approaches represent a fundamental and widely used technique [24-19]. This method involves the comparison of predefined templates or patterns with the input data to identify instances of specific keywords or phrases. Through a process of cross-correlation or similarity scoring, these approaches enable the system to recognize target keywords amidst varying backgrounds and noise levels, making them valuable in tasks such as text recognition.

The template matching algorithm, incorporating the Dynamic Time Warping (DTW) technique, is widely employed in Keyword Spotting (KWS) [21]. Unlike various other techniques, it does not necessitate the training of models, and the addition of new keywords does not require retraining. In the basic procedure, keywords are stored as templates in a database and are compared to the unknown spoken utterance for similarity. Whether a keyword is found or not is determined based

on the similarity distortion between the two sequential data. Due to recent advancements in computing power, the DTW-based approach is commonly utilized in KWS.

Template Matching KWS is computationally efficient and is effective for specific keywords with relatively consistent acoustic characteristics. However, it may encounter challenges with variations in pronunciation or noisy environments since it relies on exact matches.

2.4.2 HMM-Based Keyword Spotting

In the realm of KWS, HMM-based approaches represent a significant paradigm. These approaches leverage HMMs to detect and recognize specific keywords or phrases within audio and speech data. HMMs excel at capturing temporal dependencies and variations in acoustic signals, making them particularly well-suited for tasks like speech recognition and keyword spotting. By modeling the statistical properties of both keywords and non-keyword segments, HMM-Based Approaches have played a crucial role in enhancing the accuracy and robustness of keyword spotting systems, enabling applications in fields ranging from voice assistants to transcription services.

HMM is typically applied for speech recognition and has found use in KWS as well. In this technique, HMM model is constructed for both keyword and test utterances. The model designated for elements other than the keyword is termed the garbage model or filler model [12]. The probability is computed for every speech utterance to determine its proximity to the keyword. The majority of previous work on keyword spotting has employed HMM models [3]. However, HMM-based keyword spotting encounters challenges, such as the need for a substantial amount of training data, time-consuming processes, and a requirement for linguistic expertise. Additionally, incorporating new keywords necessitates retraining. Despite attempts to address these issues in HMM-based KWS, the solution often involves further training, posing a challenge for the audio indexing of new phrases. Consequently, in recent years, DTW-based KWS has gained traction as an alternative [15-17].

2.4.3 ANN Based Keyword Spotting

Artificial Neural Network (ANN) Based Approaches constitute a significant branch within the domain of keyword spotting. These methods harness the power of neural networks, inspired by the structure of the human brain, to recognize and extract keywords from various data sources, such as audio, text, or images. ANNs offer the advantage of learning complex patterns and representations from large datasets, making them versatile for tasks like speech recognition and

natural language processing. Whether through CNNs for image-based keyword spotting or RNNs for sequential data like speech, these ANN-based approaches have contributed to substantial advancements in keyword spotting accuracy, enabling applications in fields like content recommendation, voice search, and automated transcription.

Recent years have witnessed significant advancements in Keyword Spotting (KWS) thanks to ANNs. The surge in deep learning techniques and the abundance of available data have revolutionized the landscape of KWS, primarily owing to the superior performance of deep learning-based methods. These advanced KWS systems typically incorporate DNNs alongside compression techniques or multi-modal training strategies [23]. In the DNN approach, test audio undergoes preprocessing to eliminate noise, followed by feature extraction employing the cepstral technique. The ANN is then trained using these cepstral values to establish a set of optimal weights. During the testing phase, these weights are employed to analyze audio files.

The training process involves feeding labeled audio samples to the network, adjusting its weights to minimize the detection errors. Once trained, the system can process incoming audio data and output the detected keywords. ANN-based KWS systems have shown great success due to their ability to learn complex patterns and generalize well to different speakers and environments. A neural network is constructed through a complex network of interconnected processing units, commonly referred to as neurons, which carry out elementary mathematical computations. These neural networks are distinguished by their unique configurations, weight vectors, and activation functions applied within the hidden and output layers, as elaborated in subsequent sections [18], [24-29]. Further elaboration on this topic is provided in the following section for a more comprehensive understanding.

2.5 Deep KWS System

Nowadays Google's presents KWS system using a Deep Neural Network (DNN), trained to predict sub-keyword goals [15]. The DNN has demonstrated superior performance compared to a typically used approach for keyword spotting, which involves a Keyword/Filler Hidden Markov Model system. Moreover, the DNN holds an advantage for on-device execution, as the model size can be easily modified by adjusting the number of parameters within the network.

The present KWS system at Google [15] utilizes a DNN that has been trained to anticipate sub-keyword objectives. Extensive testing has demonstrated that the DNN consistently surpasses the performance of a traditional keyword/filler HMM system, which is commonly utilized for keyword detection. Furthermore, the DNN's adaptability for on-device usage is noteworthy, as it can be easily tailored to varying model sizes by adjusting the number of parameters within the network.

In recent times, the deep Keyword Spotting (KWS) paradigm has garnered considerable attention, primarily due to the following three key factors [20], [22]:

- It eliminates the need for a complex sequential search algorithm, such as Viterbi decoding. Instead, a considerably simpler posterior handling approach is sufficient.
- The flexibility of adapting the DNN responsible for generating the posteriors (acoustic model) is easily achievable [26], making it possible to tailor it to match computational resource constraints.
- It consistently delivers significant improvements over the keyword/filler HMM method in scenarios with limited resources, characterized by low memory and low computational complexity, both in clean and noisy environments.

These three compelling factors render the deep Keyword Spotting (KWS) paradigm highly appealing for implementation across a range of consumer electronic devices characterized by limited resources. These devices include earphones and headphones [27], smartphones, smart speakers, and similar products. For this reason, a lot of studies on deep KWS has been carried out when we considering 2014 till nowadays, e.g., [3], [15], [22], [24], [25]. Furthermore, it's worth noting that we can anticipate deep KWS to continue being a prominent and evolving subject of interest in the future, despite the significant progress achieved thus far. Figure 2.1, as referenced from [25], illustrates the overall structure of a contemporary deep spoken keyword spotting system, comprising three main components:

1. The speech features extractor, responsible for converting the incoming signal into a condensed representation of speech.
2. The deep learning-based acoustic model, which produces posterior probabilities for various keyword and filler (non-keyword) categories based on the speech features.
3. The posterior handler, which manages the temporal sequence of posteriors to determine the potential presence of keywords within the input signal.

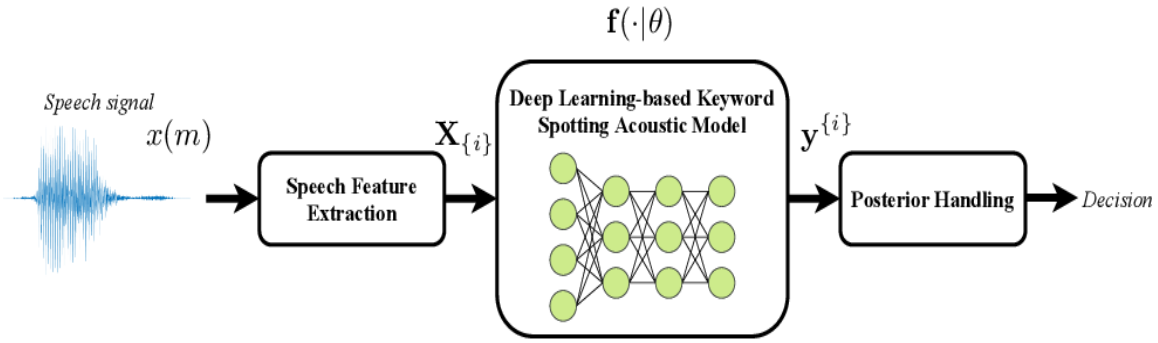


Figure 2.1 General pipeline of a modern deep spoken KWS system [3]

A more detailed depiction of the Deep KWS framework is provided in Figure 2.2 [25] below. This framework is composed of three fundamental components. (i) This module is responsible for conducting voice-activity detection and generating a feature vector for each frame. These feature vectors are then combined, incorporating contextual information from both the left and right, to form a larger vector. This augmented vector serves as input to the DNN. (ii) The DNN is trained to forecast posterior probabilities associated with each output label based on the concatenated feature vectors. These labels can correspond to complete words or sub-words relevant to the keywords. (iii) In the final step, a straightforward posterior processing module takes the label posteriors computed for each frame and combines them into a confidence score utilized for the detection process.

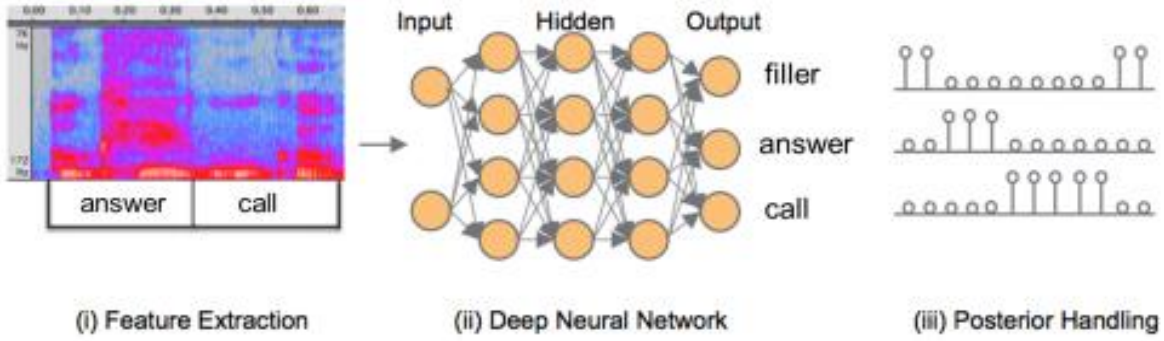


Figure 2.2 Framework of Deep KWS system, components from left to right.
(i) Feature Extraction (ii) Deep Neural Network (iii) Posterior Handling

In the example Figure 2.2, the audio consists of the keyword “answer call”. The DNN in this example only has three output labels: “answer”, “call”, and “filler” and it generates frame-level

posterior scores shown in (iii). The posterior handling module combines these scores to provide a final confidence score for that window.

2.6 Deep Learning Models

KWS based on deep learning has a significant advancement, leveraging the capabilities of neural network architectures to enhance the accuracy and efficiency of keyword detection in spoken language [2-19]. Deep learning models, such as Convolutional Neural Networks (CNNs) [24] and Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [30], have been applied to KWS tasks. These models excel at learning hierarchical representations of complex features, capturing both local and temporal dependencies in audio data.

2.6.1 Convolutional Neural Networks

CNNs have indeed brought about a revolution in several domains, including computer vision and audio processing [24]. Their capacity to autonomously acquire and extract pertinent features from input data has been a game-changer. CNNs excel in capturing localized patterns and spatial relationships within images, rendering them highly suitable for a range of tasks, such as image classification, object detection, and even speech recognition [2].

The key components of a CNN include, as indicated in Figure 2.3, convolutional layers, pooling layers, and fully connected layers. Convolutional layers consist of a set of learnable filters that are convolved over the input data, extracting local features through convolution operations. These filters are responsible for capturing different types of patterns at varying scales, such as edges, corners, and textures.

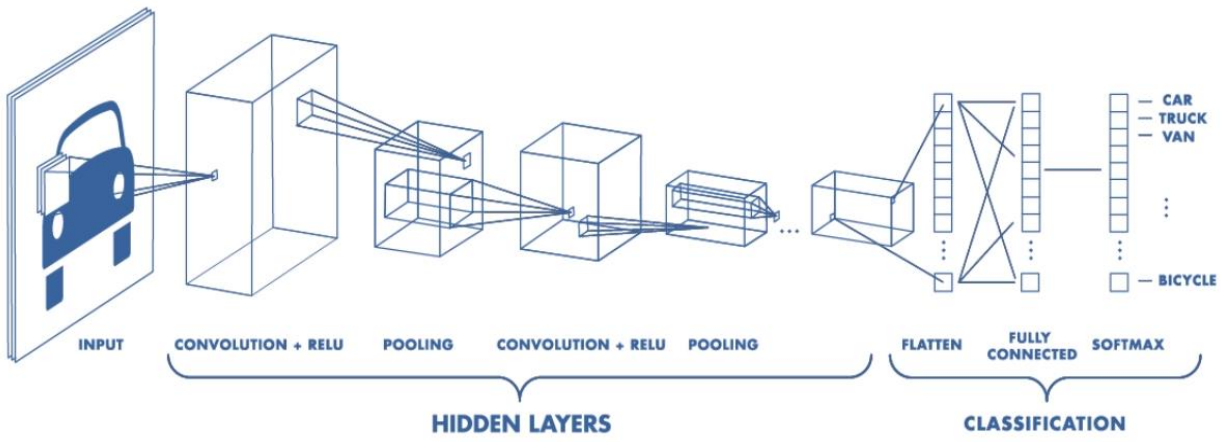


Figure 2.3 CNN Architecture

I. Convolutional Layer:

The CNN algorithm took its name from the convolution layer. In this layer series of mathematical operations i.e., matrix operation performed to extract a feature map from the input image [2].

Convolution Operation: The Convolution Layer, a pivotal element of CNNs, employs a convolution operation. This involves the use of small filters, or kernels, that traverse the input data. Through element-wise multiplications and summations, feature maps are generated. This operation is crucial for capturing spatial hierarchies and local patterns in the data, enabling the network to discern basic shapes and textures.

Activation Function: Following the convolution operation, an activation function is applied elementwise to the feature maps. A common choice is the Rectified Linear Unit (ReLU), introducing non-linearity. The activation function is crucial for introducing non-linearities into the network, allowing it to learn intricate patterns and relationships in the data [9].

Multiple Filters: Multiple filters are employed in this layer, each aimed at detecting different features. The outputs of these filters form feature maps, representing the presence of various features in the input.

II. Pooling Layer (Subsampling Layer)

The Pooling Layer is typically inserted after the convolution operation to reduce the spatial dimensions of the feature maps while preserving crucial information. Common pooling types include Max Pooling and Average Pooling. The pooling operation samples the feature maps by taking the maximum (Max Pooling) or average (Average Pooling) values within a local neighborhood (pooling window) of each feature map. This reduces computational complexity and makes the network more resistant to variations in input position.

Pooling Parameters such as the size of the pooling window and the stride (the step size by which the window moves) can be adjusted to control the amount of down sampling and the spatial resolution of the feature maps [2], [9].

III. Fully Connected (FC) Layer

Flattening: Flattening is the process of transforming the two-dimensional array or matrix obtained from the preceding convolutional and pooling layers into a one-dimensional array. It essentially unravels the multi-dimensional data into a flat vector. Flattening is necessary to feed the data into

the subsequent fully connected layers. It converts the spatial hierarchies and patterns learned by the convolutional layers into a format suitable for dense neural networks.

Dense Layers: Following flattening, one or more fully connected layers are added. The dense Layer, or fully connected layer, is a traditional neural network layer where each neuron is connected to every neuron in the preceding layer. It processes the flattened input, applying weights and biases to generate high-level features. Dense layers are instrumental in learning complex patterns and relationships within the data. They contribute to the overall understanding of the input data and play a vital role in making final decisions.

Output Layer: The final fully connected layer typically serves as the output layer. In tasks like image classification, this layer often has as many neurons as there are classes, with each neuron representing the likelihood of the input belonging to a particular class. A softmax activation function is frequently used to convert these raw scores into class probabilities [2], [9].

2.6.2 LSTM

LSTM is a type of RNN architecture that has gained significant popularity in various fields of deep learning, including natural language processing, speech recognition, and time series analysis [30]. LSTM networks are designed to address the vanishing gradient problem that occurs in traditional RNNs, making them well-suited for handling sequences of data with long-term dependencies. RNNs are designed with multiple time-steps, and each time-step comprises a single network layer, typically utilizing the Tanh activation function. In contrast, LSTM networks incorporate a memory cell for each time-step, consisting of four distinct network layers. Among these layers, there are Sigmoid layers, responsible for controlling the input, output, and forget operations, while the remaining layer utilizes the Tanh activation function. [24].

The input gate plays a critical role in determining what information should be retained in memory, while the output gate determines which information should be conveyed as output. Within LSTM memory cells, there exist two distinct states: the hidden state, which aligns with the output, and the memory state denoted as 'C,' responsible for storing memory inputs. Detailed equations illustrating the operations of these gates and states can be found in Figure 2.4, providing a comprehensive understanding of their functioning [24], [29], [30].

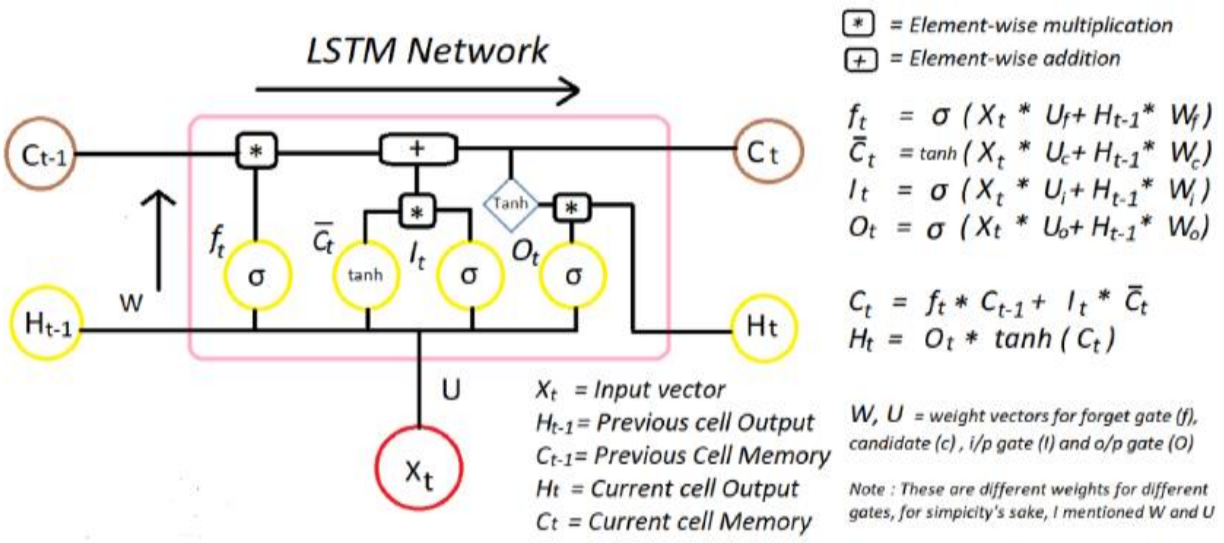


Figure 2.4 LSTM Cell Equations

The cornerstone of long-term memory within the context of LSTMs lies in the operation of the forget gate. In traditional RNNs, all incoming inputs are accepted [29]. Consequently, this approach may inadvertently replace previously accepted pertinent information with new, perhaps irrelevant data. LSTMs, on the other hand, prevent this issue through the selective action of the forget gate, which filters out irrelevant information from the memory state, thus enabling the retention of meaningful long-term memory. This is achieved by performing a multiplication operation between the prior memory state and the forget gate, denoted as 'f.' When 'f' equals zero, it signifies complete forgetfulness of the memory state, allowing only the current input ($C'_t * I_t$)) to contribute to the ongoing processing:

$$C_t = C_{t-1} * f_t + C'_t * I_t, \quad (2.1)$$

Memory cells additionally feature an input modulation gate, often regarded as a subsidiary component of the input gate. This gate serves to standardize the input information, thereby enhancing the rate at which convergence is achieved.

2.7 Related Works

The research in the field of Automatic Speech Recognition (ASR) is dedicated to addressing a multitude of technical requirements and obstacles. It has shown significant interest from both academic and industrial sectors in recent times. This section reviews existing literature and research efforts pertaining to KWS. KWS, a fundamental component of speech and audio

processing, focuses on the detection and recognition of specific keywords or phrases within an audio stream. Researchers across various domains have explored diverse methodologies and techniques to advance KWS systems' accuracy, efficiency, and adaptability to different applications and languages. The following paragraphs provide an overview of notable studies and review different studies to frame the scope of the thesis and the existing problem.

Conventional methods for KWS predominantly use the LVCSR model, as outlined in references [4-11]. In these conventional and traditional approaches, the KWS task is essentially streamlined to the search for specific keywords within spoken utterances that have already undergone recognition and subsequent indexing by the LVCSR model. Unfortunately, these traditional approaches are impractical for on-device KWS due to the stringent limitations imposed by memory and power constraints inherent in portable devices. Nevertheless, the evolving landscape of technology demands a better solution that enables the real-time detection of keywords directly from the ongoing audio stream, eliminating any delay in the process.

Another approach which aims to overcome the tackle of LVCSR, is the widely adopted technique in KWS known as the Keyword/Filler HMM technique, as referenced in [3], [12-18]. Despite its initial inception more than two decades ago, this approach continues to maintain its competitiveness within the field. In this generative methodology, individual HMM models are trained for each keyword of interest, alongside a separate filler model HMM, trained using non-keyword segments from the speech signal (referred to as fillers). During runtime, these systems necessitate Viterbi decoding, whose computational process requirements can vary significantly depending on the complexity of the HMM topology employed.

In recent times, the state-of-the-art in KWS has undergone a transformation, with deep learning-based approaches replacing traditional methods due to their remarkable performance gains [3]. These modern KWS systems predominantly use DNNs in combination with compression techniques or multi-style training methodologies. This fusion of techniques results in significant enhancements over the traditional HMM method. The current deep learning framework at KWS has undergone extensive study, with many varieties of neural networks taking the lead. CCNNs and RNNs, two architectures that both have improved contextual modeling capabilities, are included in the range of networks that make up these systems [19-29].

In reference [23], a novel sequence discriminative training framework is introduced, tailored for both fixed vocabulary and unrestricted acoustic KWS tasks. This framework systematically explores the application of sequence-level generative and discriminative models. Experimental results demonstrated that these proposed approaches consistently achieved valuable improvements in both fixed vocabulary and unrestricted KWS tasks, outperforming prior methods that relied on frame-level deep learning for acoustic KWS.

In reference [24], the authors introduce a discriminative KWS system that relies solely on RNNs. This system uses information from long time spans to predict keyword probabilities. In a challenging keyword spotting task conducted on a vast database of unconstrained speech, where a traditional HMM-based speech recognizer attains only a word accuracy of 65%, this system demonstrated remarkable performance, achieving a keyword spotting accuracy of 84.5%. In another significant development described in [25], the authors propose an attention-based end-to-end neural approach for small-footprint KWS. This approach aims to simplify the process of constructing a production-quality KWS system. The authors conducted an evaluation of different encoder architectures, including LSTM, GRU, and CRNN. Their experiments on real-world wake-up data revealed that the CNN-based approaches consistently outperformed the other architectures in terms of performance.

DNNs come with a potential drawback in that they don't account for the inherent structure and context of the input data. Audio inputs may exhibit strong time or frequency domain dependencies, and DNNs may not effectively capture these. In pursuit of exploiting these local connectivity patterns through weight sharing, CNNs have been explored for KWS. However, it's worth noting that CNNs may face challenges in modeling context across the entire frame without employing wide filters or extensive depth.

On the other hand, RNNs have also been studied for KWS, often using connectionist temporal classification (CTC) loss as opposed to the cross-entropy (CE) loss commonly used in DNN and CNN models [28]. However, achieving high accuracy at low false alarm rates remains a challenge, given the demanding requirements of such applications. Similar to DNNs, one potential limitation of RNNs lies in the focus on modeling input features without explicitly learning the structural relationships between successive time and frequency steps. More recently, [29] introduced the

CRNN architecture with the CTC loss. Despite the utilization of large model sizes, like RNNs, achieving high accuracy at low false alarm rates has proven to be a challenging endeavor.

Insights gained from these studies are invaluable in shaping the design and development of our Amharic language Keyword Spotting system and its potential impact on diverse applications. Hence, the primary motivation behind this thesis is to design and develop an efficient and robust KWS system specifically catering to the linguistic complexities of the Amharic language. By leveraging the powerful GAF representation and CNN-LSTM, we aim to create a KWS model that can accurately spot keywords in audio streams, demonstrating effectiveness in various linguistic contexts and environments.

Chapter Three

Data set preparation

3.1 Overview

In the previous chapter, the main concepts necessary to discuss the Keyword spotting system are explained. The overall signal processing chain and then, the speech features and the neural network-based classifier are studied.

In this chapter, the primary emphasis lies in providing the detailed description of the dataset preparation. A detailed description of the audio data criteria, the chosen keyword and the collection procedure utilized to prepare the Amharic audio dataset to create a KWS model using the chosen modeling techniques is presented. Additionally, this chapter includes the presentation of evaluation metrics that serve as crucial criteria for assessing the performance of the developed KWS model.

3.2 Data Description

3.2.1 Criteria and Properties

At the outset, we establish a set of guidelines to ensure that contributors record audio data with precision and adherence to consistent standards. Consequently, we have chosen the following specifications for the audio files [11].:

- A sampling rate of 16 kHz is used since it is typically used in most previous references.
- The utilization of a 16-bit representation for each sample in the signal to balance precision and file size.
- The use of one audio signal instead of stereo signals for simplicity and uniformity.
- A standardized audio duration of one second for each file [17].
- The saving of each audio file in the .wav file format, a format like what was used in the GSC dataset.

These specifications collectively contribute to the accurate and standardized collection of audio data in line with our research objectives.

3.2.2 Keyword Choices

There are 40 criminal keywords which are identified and selected through consulting with legal experts and domain-specific literature [11], [34-35]. Therefore, we have carefully selected a set of commonly used words specifically designed for the purpose of crime detection. These words can be used with telephone tapping and audio monitoring devices by security organizations to enhance their surveillance capabilities. Table 3.1 enumerates 40 chosen Amharic criminal terms alongside their corresponding English translations.

Table 3.1 The 40 Selected Amharic Criminal Keywords with their Translations.

Keyword	Translation	Keyword	Translation	Keyword	Translation
እሰራት	Incarcerate	ሰልለው	Spy on him	ዶላር	Dollar
አግተው	Kidnap	ሙስና	Corruption	እርምጃ ይወሰድበት	Take action
ድፈራት	Rape the girl	ዘልዝለው	Cut him up	ፈንክተው	Hit on the head
ጥለፋት	Abduct	አፈንዳው	Explode it	ጋንጃ	Weed
ረሽናቸው	Massacre them	አጋየው	Burn it down	ግጨው	Hit him
ጸጥ አድርገው	Finish him	ዘርረው	Put him out	ግደለው	Kill him
ፎርጂድ	Forged	አቃጥለዉ	Burn	ሽጉጥ	Gun
ሽብር	Terror	አፍነዉ	Cut his breath	ሺሻ	Shisha
ገጀራ	Machete	አስፈራራዉ	Threaten	ስረቀው	Steal
እጽ	Drug	አሶግዱት	Get rid of him	ተደበቅ	Hide your self
ጉቦ	Bribe	ጨፍጭፋቻዉ	Genocide	ተኩስበት	Shoot him
ዝረፈው	Robe him	ጩቤ	Knife	ውጋው	Stab him

ረፍርፈው	Lay them down	ደብድበዉ	Bit him		
ድፋው	Grave him up	ደብድቡት	Bit him		

3.2.3 Collection Procedure

Our process for collecting voice recordings took into account the amount of time we could reasonably expect each contributor to spend without causing annoyance or fatigue. We determined that ten repetitions of each keyword per contributor were acceptable. As a result, each contributor was asked to pronounce all 40 keywords consecutively, which we referred to as a "round". During a round, the contributor would view a set of 40 slides, one for each keyword. They would then consecutively pronounce each keyword as they moved from slide to slide. We found this approach to be highly effective due to its versatility in capturing a wide range of utterances and vocal pitches. One of the benefits of this technique is its ability to accommodate variations that may occur as contributors grow fatigued or experience changes in their vocal tone. Following each round of recording, the audio file was diligently saved. We iterated this process a total of 10 times with each contributor, and the cumulative time spent collecting data from each contributor across all 10 rounds averaged approximately half an hour.

The set of contributors was comprised of 30 individuals (19 males and 11 females) hailing from various regions of the country and speaking in slightly different accents. Their ages varied significantly, with some as young as 13 years old and others as old as 91. We obtained consent from each contributor in person before recording their voices. To ensure the highest quality recordings possible, we conducted all recordings in quiet environments or enclosed spaces where private conversations could not interfere. This is because the thesis focuses in noise free criminal keyword audio files for better detection. While there still may have been some naturally occurring background noise present, it was difficult to eliminate entirely without the use of a soundproof studio, which could be costly. For most of the recordings, we utilized an affordable external microphone, a headset with close-speaking, noise cancelling and mono-phone features. However, in certain instances, we also made use of the laptop's built-in microphone. It's worth noting that the data collection process took place during the months of February and March in the year 2023.

Each audio file resulting from our recordings contained 40 utterances of the selected keywords, with each being separated by the appropriate pause. To facilitate effective data analysis, it was necessary to partition the larger audio files into smaller segments, each lasting one second. These segments were carefully crafted to isolate a single keyword, which was typically spoken around the midpoint of the audio segment. This segmentation process enabled us to precisely extract and analyze individual keyword instances within the audio dataset. Because of the difficulty in obtaining clean, one-second-long audio recordings, we chose to manually clip the files ourselves using free and open-source software called Audacity. This process was time-consuming, with it taking about an hour to process the data collected from a single contributor. The final layout of the dataset is shown in Figure 3.1.

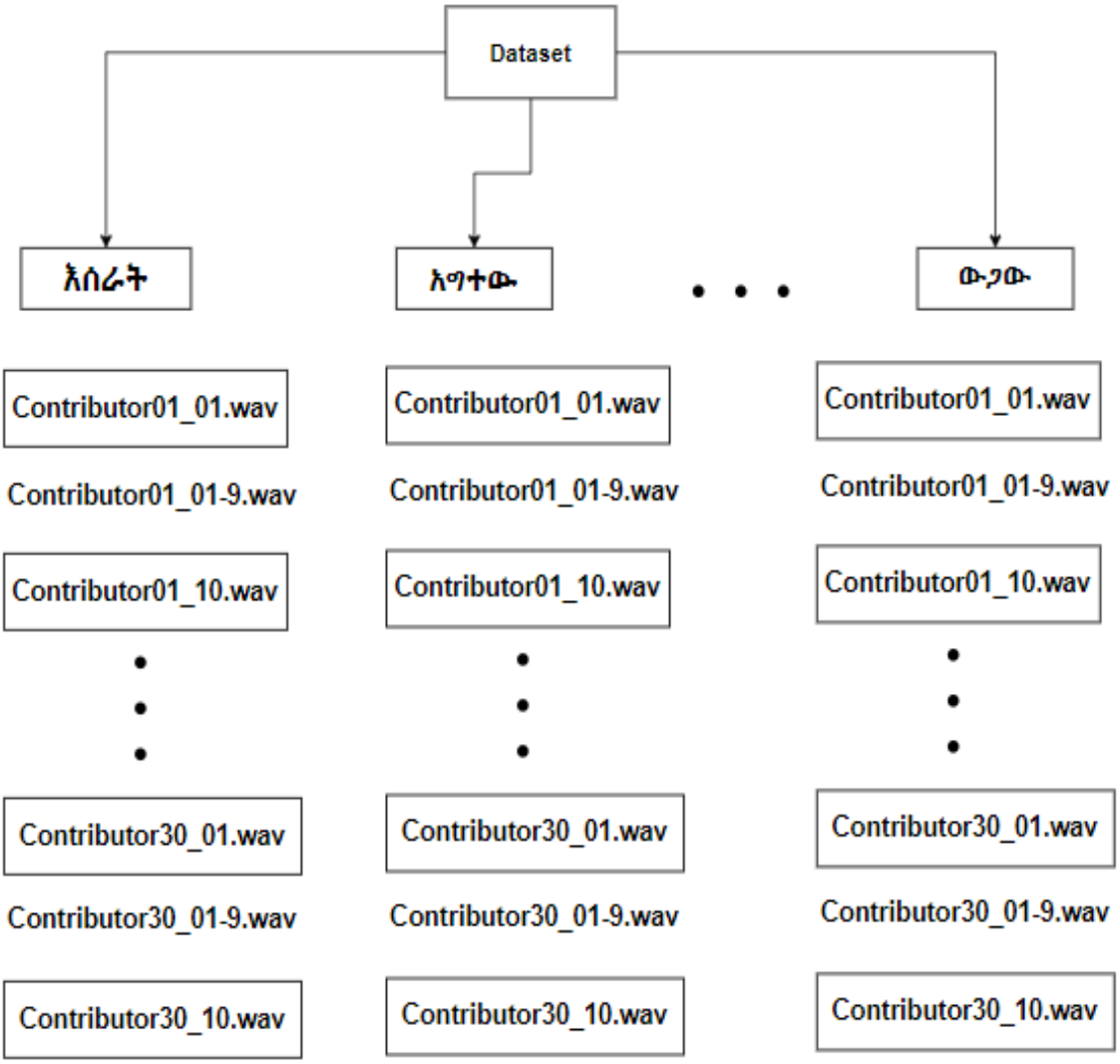


Figure 3.1 The Final Structure of the ASC Dataset

The generated dataset consists of 40 different folders, each one representing a distinct keyword and containing a total of 300 individual files. Each filename consists of sixteen characters - the twelve and threaten digits indicating the contributor number and the last two digits identify the round number. For instance, if we examine the file path "አቃጥለው/Contributor21_06.wav", we can see that it corresponds to the pronunciation of the keyword "አቃጥለው" by 21th contributor during the sixth round of recordings.

3.3 Evaluation Metrics

Like many other systems, evaluating the performance of a KWS system requires metrics that are typically based on the confusion matrix for classification. The confusion matrix, illustrated in Figure 3.2, provides a summary of a model's predictions on unseen data (test data). A true positive (TP) occurs when the model correctly predicts a positive class for a positive input, while a true negative (TN) denotes correct predictions of a negative class for a negative input. Conversely, false positives (FP) occur when the model incorrectly predicts a positive class for a negative input, and false negatives (FN) happen when the model incorrectly predicts a negative (N) class for a positive(P) input.

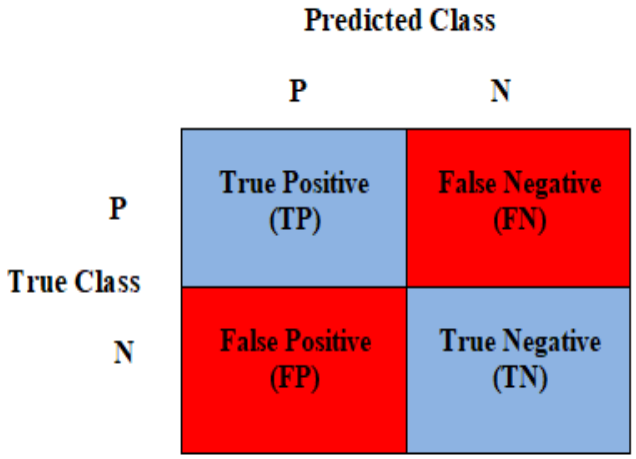


Figure 3.2 A Confusion Matrix Summarizing the Predictions

The following evaluation metrics are commonly used in keyword spotting systems.

3.3.1 Accuracy

One of the most straightforward yet widely used evaluation metrics is accuracy. Accuracy is a measure of how many correctly classified instances are among all instances in a dataset. It is useful when the classes in the dataset are balanced, meaning there's an equal number of instances for each class. It provides a general overview of the model's correctness but might be misleading in imbalanced datasets where the number of instances in one class significantly outweighs the other. For example, in a dataset with 95% negative instances and 5% positive instances, a model that predicts all instances as negative would still achieve 95% accuracy. Accuracy is calculated by dividing the number of accurate classifications by the total number of classifications. In binary

classification scenarios, (distinguishing between keywords and non-keywords), this can be expressed as the sum of true positives and true negatives divided by the total number of classifications, as illustrated in equation (3.1). A high accuracy score suggests that the model is making a substantial number of accurate predictions relative to the total number of predictions. When dealing with multiple keywords, it's a common practice to calculate the evaluation metric separately for each keyword and then average the results [4].

$$\boldsymbol{accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}}. \tag{3.1}$$

This metric measures the overall effectiveness of the system in detecting true keywords and rejecting non-keywords.

3.3.2 Precision and Recall

Precision and recall offer a more in-depth evaluation of the proposed model. Precision is a performance metric that gauges the accuracy of positive predictions made by a model. Specifically, it calculates the proportion of true positive predictions among all instances predicted as positive. Precision provides insights into the reliability of positive predictions. It answers the question: "Out of all instances predicted as positive, how many were truly positive?" High precision indicates that the model is adept at correctly identifying positive instances and minimizing false positives. In applications where the cost of false positives is substantial, such as medical diagnoses or fraud detection, precision is a crucial metric. However, it does not account for instances predicted as negative that were positive (false negatives). Consider a model predicting whether an email is spam. A precision score of 0.90 means that 90% of the emails predicted as spam were indeed spam, while 10% were false alarms. Precision is defined as the ratio of true positive samples to the total number of true positive and false positive samples. This metric is useful in measuring the exactness of the system's positive predictions [4].

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}. \tag{3.2}$$

Recall is a performance metric that measures the ability of a model to capture all actual positive instances. It calculates the proportion of true positive predictions among all actual positive instances. Recall focuses on minimizing false negatives, providing insights into the model's capability to identify all instances of the positive class. It answers the question: "Out of all actual

positive instances, how many did the model correctly predict?" High recall is crucial when the cost of missing positive instances is significant. For instance, in medical diagnoses or security applications, failing to identify a positive case (false negative) may have serious consequences. However, recall does not account for instances predicted as positive that were actually negative (false positives). In a medical diagnostic model predicting the presence of a disease, a recall score of 0.85 means that 85% of the actual positive cases were correctly identified, while 15% were missed. Mathematically, it is defined as [4]:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}. \tag{3.3}$$

3.3.3 Receiver Operating Characteristic (ROC) Curve

The ROC curve is a graphical representation of the trade-off between true positive rate (TPR) and false positive rate (FPR) at various threshold settings [4]. The ROC curve is particularly useful for binary classification tasks, especially when dealing with imbalanced datasets or scenarios where the costs associated with false positives and false negatives differ. The ROC curve illustrates the performance of a model across different discrimination thresholds. It plots the true positive rate against the false positive rate for various threshold values. Mathematically TPR and FPR are defined as in equation (3.1) and (3.1), respectively, as shown below:

$$TPR = \frac{TP}{TP + FN} \tag{3.4}$$

$$FPR = \frac{FP}{FP + TN}. \tag{3.5}$$

The area under the ROC curve (AUC-ROC) provides a summary measure of a model's ability to distinguish between classes. A higher AUC-ROC value indicates better discrimination, with a value of 0.5 suggesting no discrimination beyond random chance [4].

3.3.4 F1 Score

The F1 score is a performance metric that combines precision and recall into a single value, providing a balanced measure that considers both false positives and false negatives. The F1 score is particularly useful in situations where there is an imbalance between the classes or when both false positives and false negatives are equally important. It is the harmonic mean of precision and recall, offering a trade-off between the two. It is denoted as:

$$F = \frac{2pr}{p + r} \tag{3.6}$$

where p and r represent precision and recall, respectively.

The F1 score ranges from 0 to 1, where a higher value indicates a better balance between precision and recall. When both precision and recall are high, the F1 score tends to be high. If either precision or recall is low, the F1 score decreases, reflecting the impact of the lower of the two.

Chapter Four

The Proposed Model for Keyword Spotting

4.1 Overview

In this chapter, we begin by introducing the GAF based CNN-LSTM model designed specifically for Amharic keyword spotting. We provide a thorough examination of its architecture and functionality. A comprehensive description of the preprocessing steps involved in converting the input audio dataset into GAF images dataset is presented in the following section. Emphasis is placed on the significance of GAF representations in capturing temporal dynamics, and we explore how CNN excels at extracting spatial features from these GAF images. Finally, we presented a comprehensive and detailed presentation of the proposed CNN-LSTM model architecture. In doing so, we put plenty of emphasis on highlighting the fundamental building blocks and important elements that support the framework of our proposed model.

4.2 Overview of GAF based CNN-LSTM model.

This section provides a comprehensive overview of the proposed GAF based CNN-LSTM model for Amharic keyword spotting. The GAF based CNN-LSTM architecture is designed to leverage the power of deep learning techniques, combining CNNs and LSTM networks to effectively process and interpret acoustic features derived from GAF representations of raw audio waveforms. The overall system architecture is shown in Figure 4.1.

The block diagram for Keyword Spotting based on GAF based CNN-LSTM is a system that can detect the presence or absence of a specific criminal keyword in an audio signal. The audio input is received from a microphone or any other audio source and preprocessed to remove noise and other unwanted components. The preprocessed audio signal is then converted into a feature vector using the GAF technique, which captures the time-frequency information of the signal.

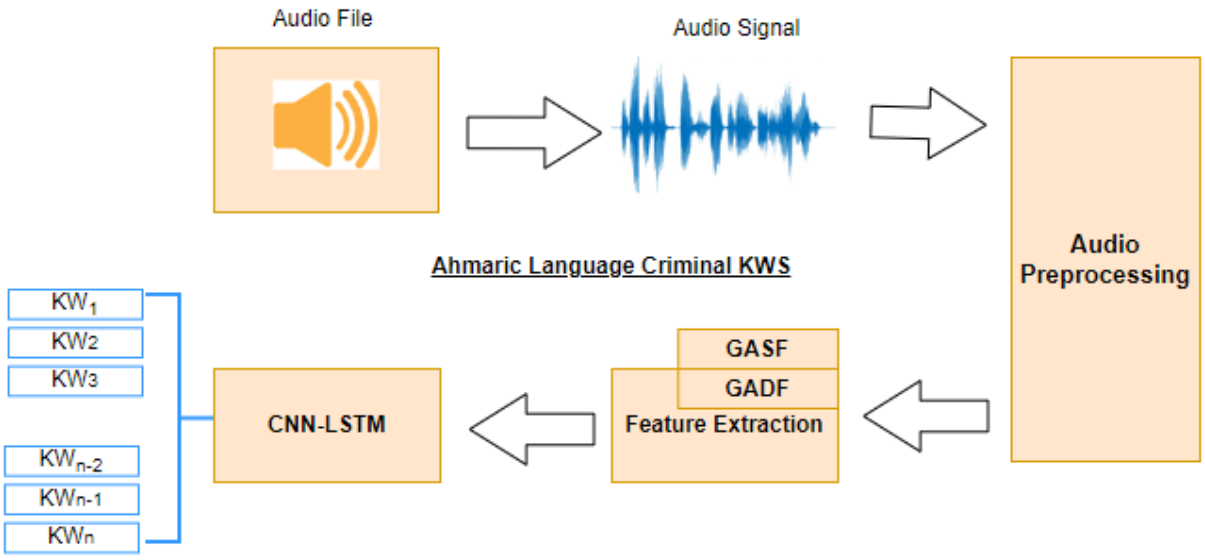


Figure 4.1 The Proposed Amharic KWS System Design

The foundation of our model lies in the transformation of audio waveforms into GAF images. The GAF representation captures the temporal dynamics and patterns inherent in the audio data, converting each audio segment into a two-dimensional matrix. This process preserves the essential sequential information while enabling the application of image-based deep learning techniques, facilitating efficient feature extraction and recognition.

The CNN component serves as the front-end of our model, tasked with extracting meaningful acoustic features from the GAF images derived from the raw audio waveforms. CNNs are adept at learning hierarchical representations through convolutional and pooling layers, which efficiently detect local patterns and aggregate them to learn global acoustic features. The output of the CNN component serves as a compressed representation that retains vital information while reducing irrelevant noise, facilitating robust and discriminative keyword detection.

The LSTM component acts as the temporal processing unit, responsible for modeling the long-term contextual relationships within the acoustic features obtained from the CNN. LSTMs, one type of RNNs, excel at capturing sequential dependencies and mitigating the vanishing gradient problem. The memory cells and gating mechanisms of LSTMs enable them to selectively store and retrieve temporal information, making them well-suited for processing audio sequences with significant context dependencies.

Following the CNN and LSTM components, our model incorporates fusion and decision layers. The fusion layer combines the spatial and temporal information learned from the CNN and LSTM, respectively, into a cohesive and enriched representation. This unified representation captures both local and long-range contextual cues, enhancing the model's ability to discriminate keywords in complex audio data. Subsequently, the decision layer employs fully connected layers and activation functions to perform the final classification, detecting whether or not the keywords are present in the input audio stream.

4.3 Dataset Preprocessing

After data set preparation the next step is preprocessing the prepared audio dataset which involves manipulating and transforming raw audio data into a suitable format for use in the proposed model. The purpose of preprocessing is to remove noise, normalize the data, and extract features that are important for the prediction task. The next subsection describes how the GAF image is generated from the prepared audio data set.

4.3.1 GAF Transformation

Before generating the GAF image, the raw audio data is preprocessed to ensure a consistent format and suitable representation for the transformation. Common preprocessing steps include noise removal, resampling the audio to a specific sampling rate, converting the audio to mono-channel (if it's in stereo), and normalizing the amplitude values to a standardized range. Then, the preprocessed audio data is divided into smaller frames or segments of fixed length. This step is necessary because GAF operates on individual segments of the audio signal rather than the entire signal at once. For each frame, the pairwise distances between the audio samples are computed. The pairwise distances are obtained by calculating the Euclidean distance or cosine distance between each pair of audio samples within the frame. This results in a distance matrix for each frame.

This block diagram given in Figure 4.2 visually represents the essential steps in the conversion process, from the raw audio waveform to a GAF image that encapsulates important temporal information.

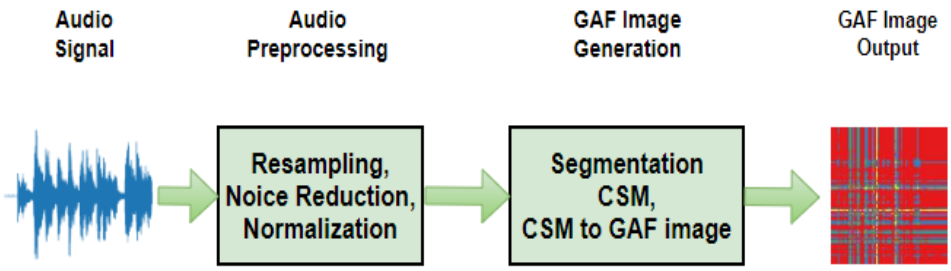


Figure 4.2 Converting Audio Waveform to GAF Image

The process of converting an audio waveform into a GAF image involves key steps. Firstly, the audio waveform is normalized to ensure consistent processing. Subsequently, the normalized waveform is transformed from Cartesian coordinates to polar coordinates. This involves applying an angular encoding function to capture directional information. Finally, a comparison of time correlations is performed for each time point in the polar coordinates. This comparison is executed using a triangular cosine function, referred to as the cosine similarity matrix (CSM), enhancing the discriminative features and revealing patterns within the audio signal. This process results in the creation of a matrix with dimensions nxn, where 'n' represents the number of sampling points within the time series data [36].

Consider a time series signal denoted as $X = \{x_1, x_2, \dots, x_n\}$, which represents a sequence containing 'n' time points along with their respective actual observations x. To mitigate potential bias associated with the inner product reaching its maximum value, the Min-Max scalar technique is used to normalize the time series to within $[0, 1]$. This step is in another expression called normalizing the input time series data. Using the Min-Max scaling formula, X is normalized ensuring that all values of X fall within the interval [0, 1]. This process can be mathematically represented as follows [36]:

$$\tilde{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)}, \quad (4.1)$$

where the notation $\tilde{x}_i$ represents the normalized data between [0, 1].

Following the normalization step, the subsequent phase involves the transformation of the normalized time series data into a polar coordinate system. Here, the rescaled values within the sequence are mapped to angles represented as ϕ_i, while the time 't' is mapped to radii denoted as

'r_i'. The two equations given below shows how to get the angles ϕ_i and radius 'r_i' from the rescaled time series data:

$$\begin{cases} \boldsymbol{\phi_i = arccos(\tilde{x}_i)}, & \boldsymbol{-1 \le \tilde{x}_i \le 1, \tilde{x}_i \in \tilde{X}} \\ \boldsymbol{r_i = \frac{t_i}{N}}, & \boldsymbol{t_i \in N} \end{cases} \tag{4.2}$$

In the equation presented above, t_i corresponds to the time stamp, while 'N' serves as a constant factor designed to standardize the range of the polar coordinate system. The bijective coding used in equation (4.2) has two key features. Firstly, the cosine function is monotonically decreasing within the range of 0 to π when ϕ_i belongs to the same range, ensuring uniqueness within the polar coordinate system and allowing for an inverse mapping that is also unique. Secondly, polar coordinates maintain an absolute temporal relationship, unlike Cartesian coordinates.

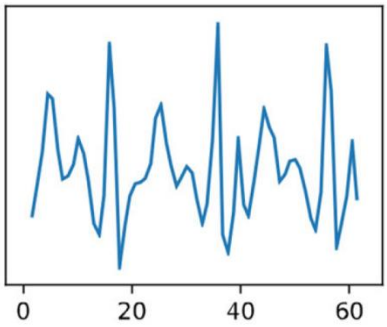
After successfully converting the rescaled time series into polar coordinates, we can easily exploit the angular perspective to examine the trigonometric summation or difference between individual points. Through this analysis, we can pinpoint temporal correlations within distinct time intervals. The Gramian Angular Summation Field (GASF) and Gramian Angular Difference Field (GADF) are mathematically specified as follows [36]:

$$\boldsymbol{GASF = \left[cos\left(\phi_i + \phi_j\right)\right] = \begin{bmatrix} cos(\phi_1 + \phi_1) & \cdots & cos(\phi_1 + \phi_n) \\ cos(\phi_2 + \phi_1) & \cdots & cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ cos(\phi_n + \phi_1) & \cdots & cos(\phi_n + \phi_n) \end{bmatrix}} \tag{4.3}$$

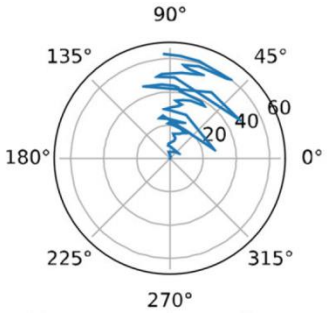
$$\boldsymbol{GADF = \left[sin\left(\phi_i + \phi_j\right)\right] = \begin{bmatrix} sin(\phi_1 + \phi_1) & \cdots & sin(\phi_1 + \phi_n) \\ sin(\phi_2 + \phi_1) & \cdots & sin(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ sin(\phi_n + \phi_1) & \cdots & sin(\phi_n + \phi_n) \end{bmatrix}} \tag{4.4}$$

These two types of GAF algorithm transform one-dimensional time series data into two-dimensional images using normalization, coordinate axis transformation, and trigonometric functions. Figure 4.3 illustrates the mapping relationship between one-dimensional time series data and two-dimensional images. Initially, the time series data undergoes transformation into a

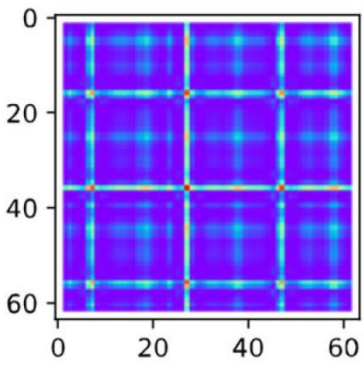
polar coordinate system through the utilization of equation (4.2). GAF images can then be produced using equations (4.3) and (4.4) to create GASF and GADF images, respectively.



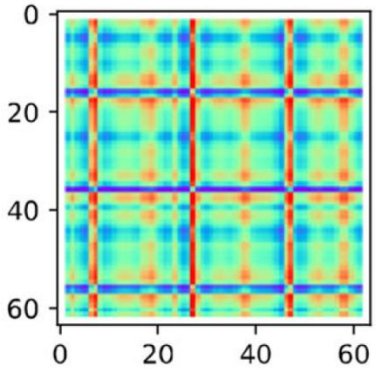
(a) A rescaled one-dimensional time series



(b) Transform time series into a polar coordinate system



(c) GASF



(d) GADF

Figure 4.3 1-D Time Series Converted to 2-D Images in Polar Coordinate System [36].

4.3.2 Visualization of TD and Advantage of GAF in KWS

The GAF image provides a visual representation of the temporal dynamics (TD) within the audio frame. Each pixel in the GAF image corresponds to a pair of audio samples, and the pixel's intensity is determined by the angle obtained from the polar mapping. GAF images inherently capture the sequential relationships and local patterns present in the audio data, allowing deep learning models to effectively analyze the temporal aspects of the keyword spotting task.

The GAF-based representation offers several advantages for keyword spotting in Amharic. By transforming the audio data into images, the model can leverage well-established image analysis techniques, such as CNNs, to extract meaningful spatial features. Additionally, the GAF

transformation effectively encodes long-range dependencies in the audio data, making the model sensitive to sequential patterns critical for accurate keyword detection.

4.4 CNN-LSTM Architecture

The last stage in our design involves utilizing the GAFs as input to CNN-LSTM architecture for keyword spotting. The CNN-LSTM architecture consists of three fundamental components: convolutional layers, recurrent layers, and fully connected layers. The proposed CNN–LSTM models' architecture is depicted in Figure 4.4.

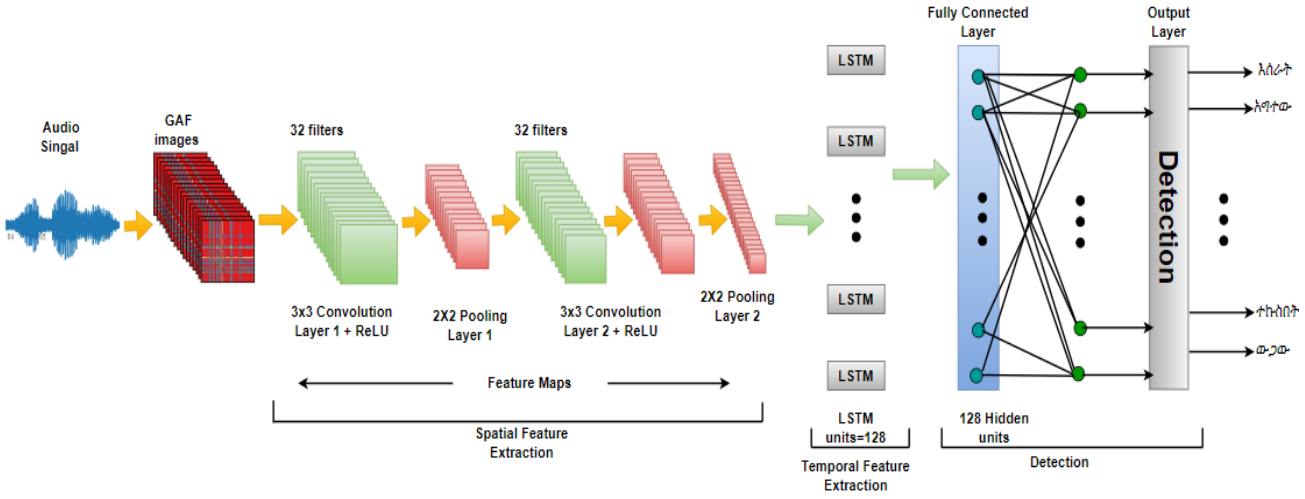


Figure 4.4 The Architecture of the proposed KWS based on GAF based CNN-LSTM Model

Convolutional Layers: the convolutional layers play a pivotal role in extracting features from the GAFs. The GAF images are fed into the CNN, which excels at extracting hierarchical spatial features. The CNN architecture incorporates several convolutional layers that perform convolutions on the GAF images, enabling the detection of intricate patterns and features. Each of these convolutional layers is succeeded by non-linear activation functions, such as ReLU, to introduce non-linearity into the model. Additionally, a max-pooling operation is applied after each convolutional layer. Specifically, our architecture includes two convolutional layers, each equipped with 32 filters. Both convolutional layers use a 3x3 kernel size. Subsequently, each convolutional layer is followed by a Max-pooling layer with a 2x2 dimension, effectively reducing the dimensionality of the feature maps.

LSTM Layer: The output from the CNN is then passed to LSTM layers, which specialize in capturing sequential and temporal dependencies within the GAF-based features. LSTM is well-suited for understanding long-range contextual information in sequential data. We use two LSTM layers with 128 units. The LSTM layers receive the output from the convolutional layers as input and carry forward information across different time steps.

Fully Connected Layers: The unified representation is subsequently passed through fully connected layers for the final classification task. These fully connected layers play a crucial role in identifying the presence of the keyword. They are responsible for determining whether keywords are available in the input audio or not. In our architecture, we used a single fully connected layer with softmax activation, which yields the probabilities associated with each keyword class. This information is derived from the resulting GAF matrix and serves as input to the CNN-LSTM model.

Chapter Five

Result and Discussion

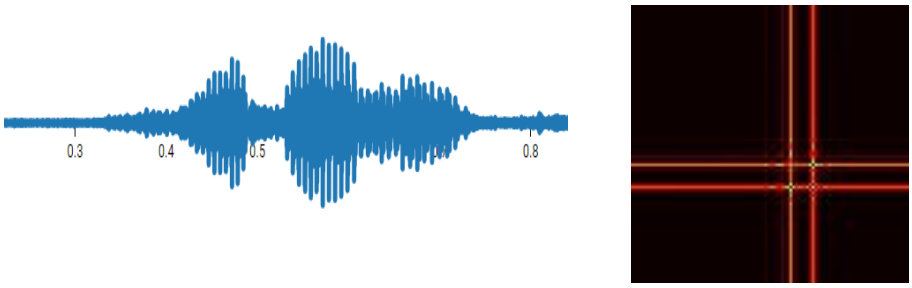
The results of our in-depth research are presented in this chapter, along with a thorough analysis of the findings. Here, we examine the results that are derived from careful testing and evaluation of our proposed method. In the first section, we provide GAF images which are the transformed outcomes of the audio file. Subsequently, we proceed to assess the overall effectiveness of the proposed model using various performance metrics.

5.1 GAF Image Generation for Amharic Audio

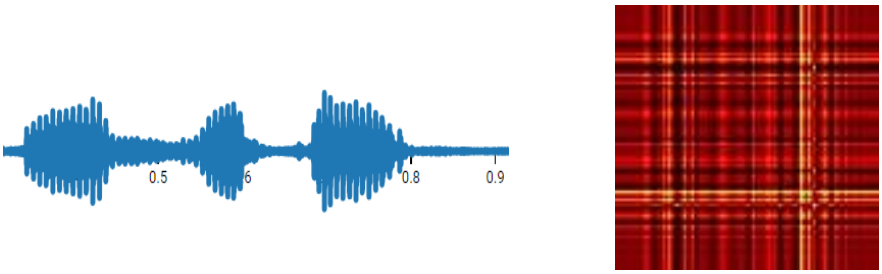
For data preprocessing, the raw audio files were transformed into GAF images, capturing temporal patterns inherent in the audio signals.

In the proposed keyword spotting system for the Amharic language, the generation of GAF images is an important step. The GAF transformation converts the raw Amharic audio signals into visual representations, capturing temporal patterns and facilitating the utilization of deep learning models designed for image processing tasks. In this section, we go over how to create a GAF image and the main methods used to convert audio data into applicable inputs for the following CNN-LSTM model.

Before applying the GAF transformation, the raw audio data underwent preprocessing to ensure a consistent format suitable for image generation. We standardized the audio samples by resampling them to a common sampling rate of 16 KHz, and then converted them to mono-channel audio to remove any stereo effects. Furthermore, we performed audio normalization to bring the amplitude values within a standardized range, which is crucial for preserving the integrity of the temporal patterns during GAF transformation. The GAF images were generated using a sliding window approach, where a fixed-size window of 25ms was used to traverse the audio sequence. At each time step, the data points within the window were selected, and the GAF transformation was applied to create the corresponding GAF image. The image obtained after applying GAF for the audio file “አቃጥለው/Contributor21_06.wav” is given as shown Figure 5.1. The resulting GAF images were then collated to form the final dataset for training and evaluation.



a) “አቃጥለው/Contributor21_06.wav”



b) “ማደለው/Contributor01_08.wav

Figure 5.1 Generated GAF Image for the Audio File

5.2 Training Process

The training process of the proposed keyword spotting system based on the GAF and CNN-LSTM architecture for the Amharic language involved several crucial steps to achieve optimal model performance. We utilized Keras, version 2.10.0 to implement and train the proposed CNN-LSTM model on the collected Amharic audio dataset.

As typically done in machine learning practice, we partitioned the dataset into three distinct groups: training, validation, and testing. To optimize the distribution of instances, we allocated 80% of the dataset to the training set, 20% of the training dataset set to the validation set and 20% to the test set. In our approach to splitting the data, we took care to ensure that all recordings from a particular contributor were grouped within the same subset. This strategic arrangement guarantees that signals with similarities are exclusively present in the training or validation/testing sets, thereby preserving the integrity of the results. Additionally, this practice facilitates the model's ability to generalize to new individuals beyond our dataset, enhancing its overall performance and applicability.

The hyper-parameters are essential to machine learning algorithms due to their direct influence on the behavior of the training algorithms and play a significant role in the model's performance. Hyper-parameter tuning was performed to optimize and maximize the performance of a CNN-LSTM model. A systematic search was conducted over a pre-defined set of parameters, which included learning rates, batch sizes, and hidden units.

5.3 Training and Testing Result Evaluation

In this section, we present the comprehensive evaluation of the proposed keyword spotting system based on the GAF and CNN-LSTM architecture for the Amharic language. The evaluation aims to assess the model's performance in accurately detecting keywords within Amharic audio streams and to measure its overall effectiveness in real-world scenarios. To quantify the model's performance, we employed standard evaluation metrics, including accuracy and precision.

5.3.1 Training and Testing Accuracy Evaluation

The training process specification hyperparameters used for the result showed in Figure 5.2, and Figure 5.3, is given in Table 5.1.

Table 5.1 First Used Hyper-parameters.

Hyper-Parameter	**Value**
Learning Rate	0.001
Batch Size	64
Activation Function	Softmax
Optimizer	Adam
Dropout	0.7
Epoch	150

So that the training and the test accuracy and loss of the proposed model are show in Figure 5.2, and Figure 5.3, respectively.

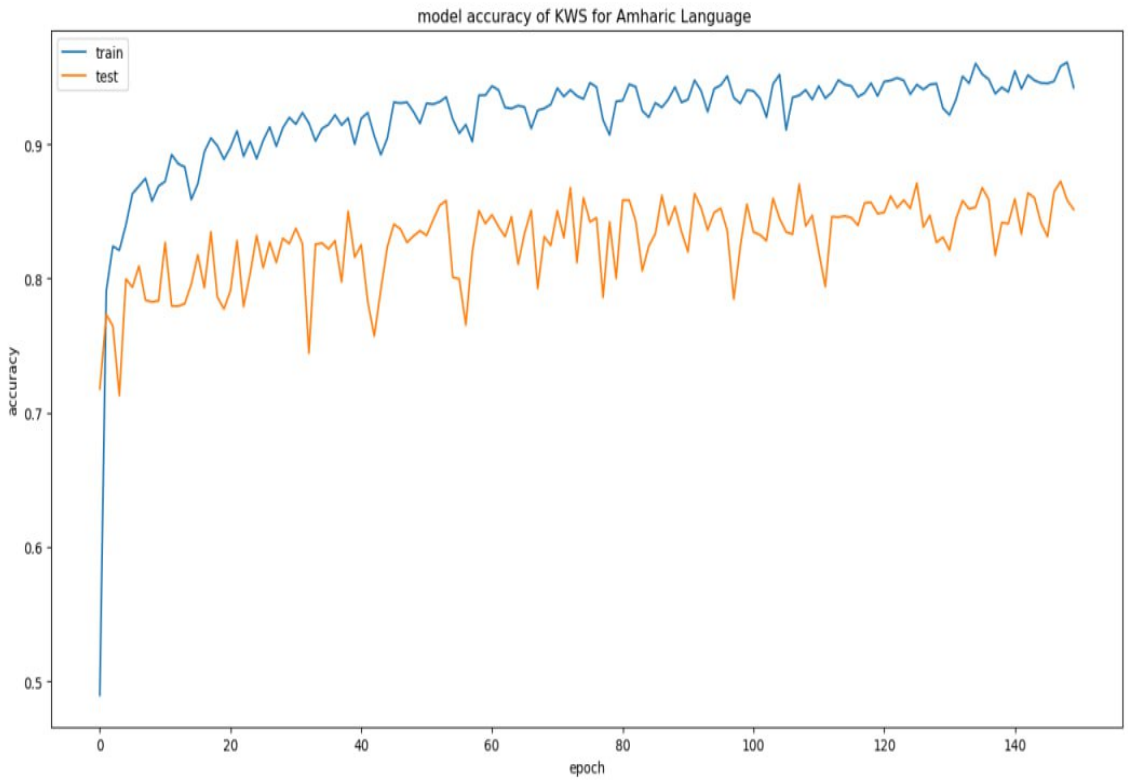


Figure 5.2 Training and Test Accuracy for the Hyperparameters Specified in Table 5.1

Training accuracy refers to the accuracy obtained on the training samples whereas the test accuracy denotes the performances on the test samples, not used for learning and never seen by the network. During each epoch, the network is first presented with training examples and then, with the test and training examples, to assess its performance. The training and test accuracy of the proposed model as indicated in Figure 5.2, results 93.98%, and 84.03%, respectively. Obviously, the training accuracy is always larger than the test one.

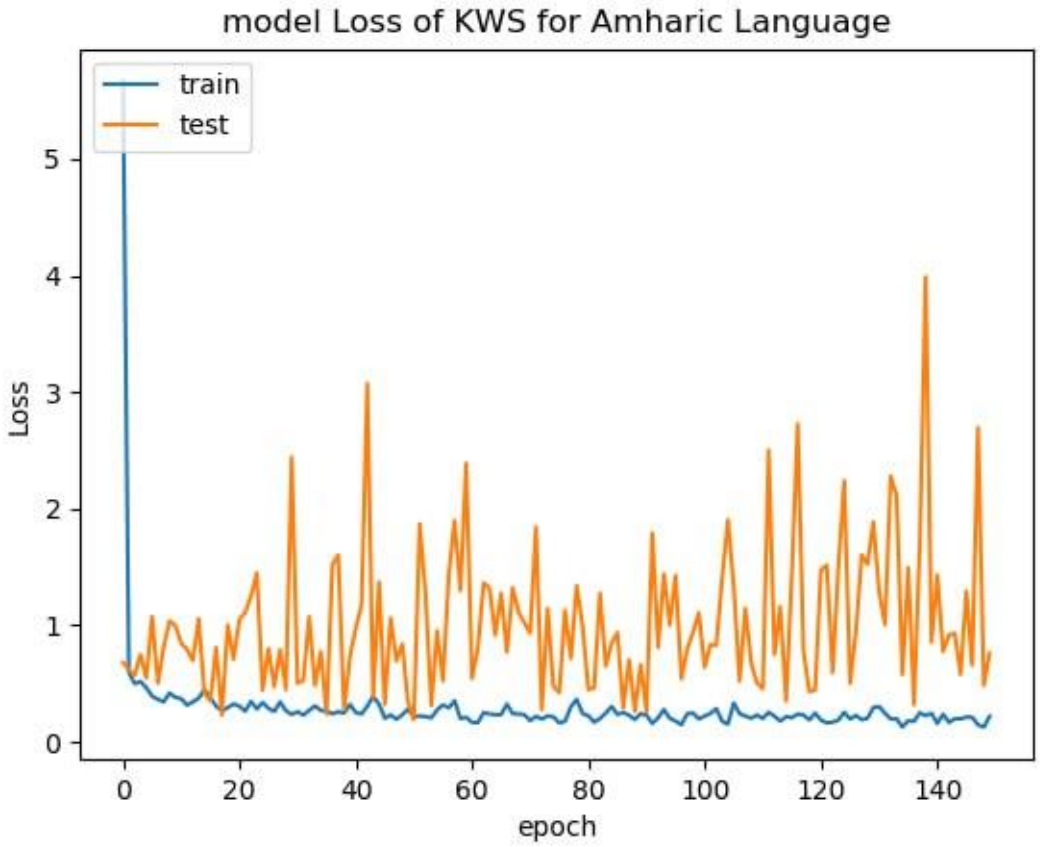


Figure 5.3 Training and Test Loss for the Hyperparameters Specified in Table 5.1

Loss functions are crucial components in deep learning models, measuring the discrepancy between predicted outputs and ground truth labels during training and evaluation. Training loss is a measure of how well a model performs on the training data set. The training loss graph reveals the optimization progress during training epochs and provides insights into the model's ability to minimize errors and fit the training data effectively. As indicated in Figure 5.3, it can be observed that as the number of epochs increases up to 100, its losses become more stable and consistent.

However, for the hyperparameters specified in the Table 5.2, the result of the proposed model is shown in Figure 5.4, and Figure 5.5.

Table 5.2 Second Used Hyper-parameters.

Hyper-Parameter	Value
Learning Rate	0.01
Batch Size	32
Activation Function	Softmax
Optimizer	Adam
Dropout	0.5
Epoch	100

The training and the test accuracy and loss of the proposed model using the hyperparameters specified in Table 5.2 are shown in Figure 5.4, and Figure 5.5, respectively.

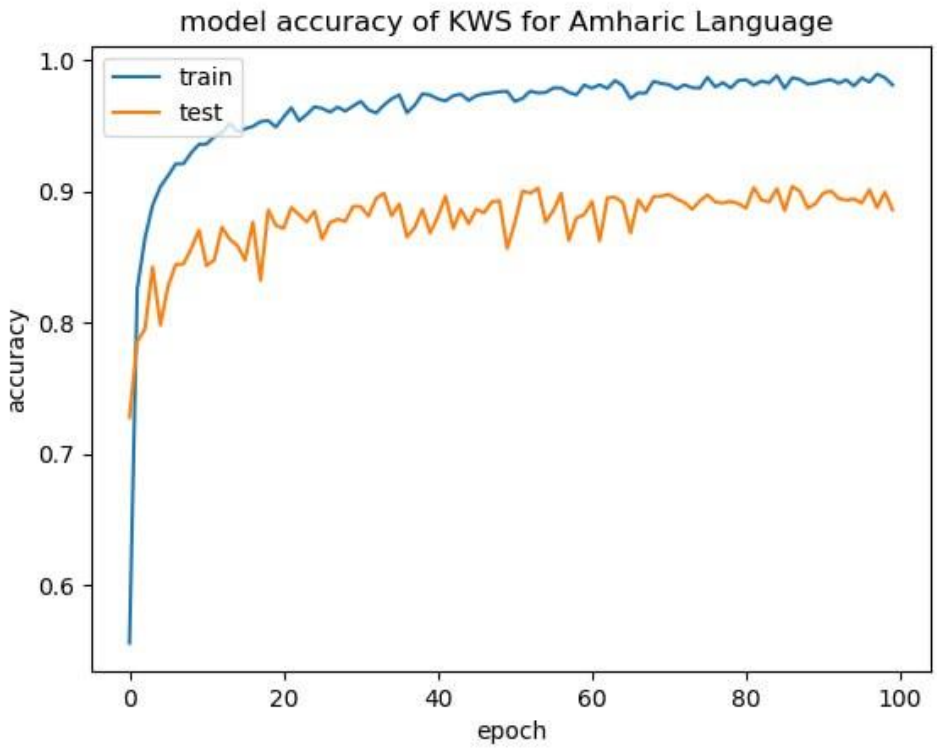


Figure 5.4 Training and Test Accuracy for the Hyperparameters Specified in Table 5.2.

The training and test accuracy of the proposed model for the hyperparameter specified in Table 5.2 as indicated in Figure 5.4 , results 96.73%, and 89.98%, respectively. Obviously, the training accuracy is always larger than the test one.

The training and the test loss verse epoch of the proposed model using the hyperparameters specified in Table 5.2 is shown in Figure 5.5.

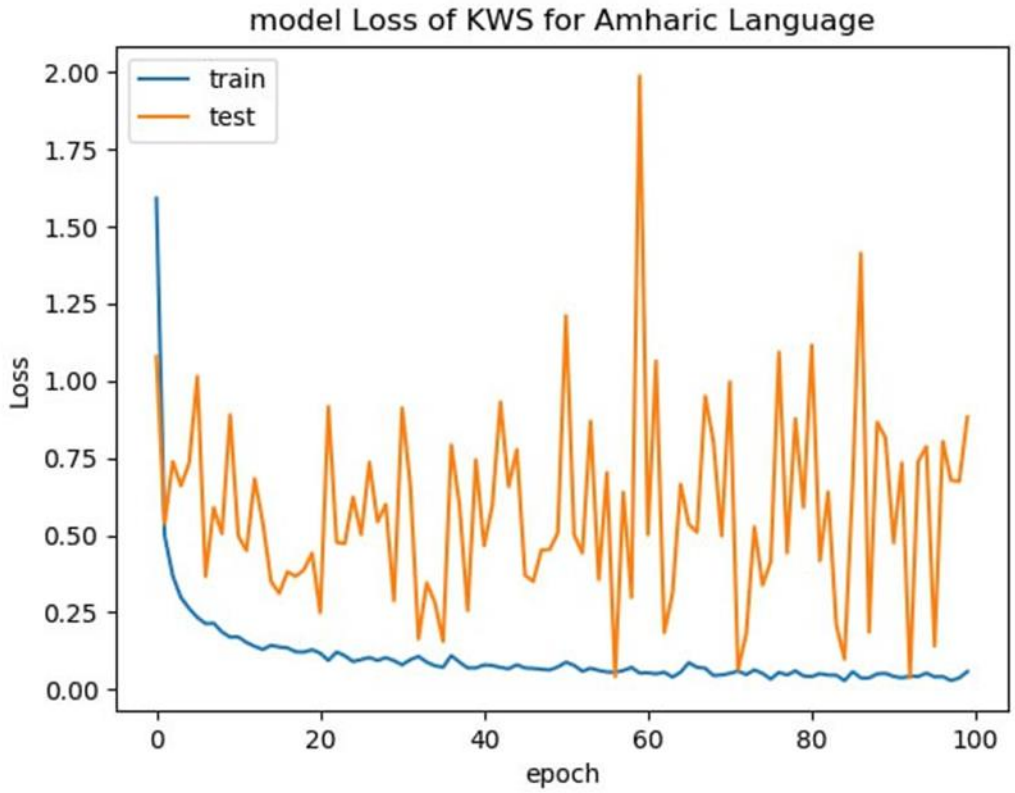


Figure 5.5 Training and Test Loss for the Hyperparameters Specified in Table 5.2.

5.4 **Discussion of Results**

The hyperparameter optimization process aimed to identify the most effective parameters for spotting Amharic keywords. The achieved training accuracy of 96.73% and test accuracy of 89.98% demonstrate the success of the selected hyperparameters. Through rigorous experimentation and fine-tuning, we ensured that the model could effectively learn and generalize from the training data to unseen test data. Notably, the chosen hyperparameters played a crucial role in enhancing the model's performance, with certain parameters showing a significant impact on accuracy.

Our study focused on the critical task of detecting criminal Amharic keywords, with the developed model showcasing promising effectiveness. The test accuracy of 89.98% underscores the model's capability to accurately identify criminal terms within Amharic text. This outcome holds significant implications for law enforcement agencies and security organizations, as it provides a robust tool for automated keyword spotting in the Amharic language, aiding in crime prevention and detection efforts.

Therefore, the hyperparameters specified in Table 5.2 demonstrates better result compared to the hyperparameters specified in Table 5.1. This conclusion enables us to answer which hyperparameters are best for the proposed model effectiveness. Despite this, the training accuracy of 96.73% and test accuracy of 89.98% obtained with the selected hyperparameters, shows the effectiveness of the proposed model.

Chapter Six

Conclusion and Recommendation

6.1 Conclusion

In this thesis, we have presented a comprehensive study on keyword spotting for the Amharic language using the innovative combination of the GAF and CNN-LSTM architecture. The research objectives were successfully achieved, and the proposed model demonstrated high performance in accurately detecting keywords within Amharic audio streams.

ASC dataset is presented, our newly developed criminal Amharic keyword dataset, drawing inspiration from the GSC dataset, and a detailed explanation of the dataset collection and preparation methodology is provided. Through the use of the GAF representation, the CNN-LSTM model effectively captured both spatial and temporal patterns in the audio data, allowing for robust and contextually aware keyword spotting. The model's high precision on the Amharic keyword dataset underscores its proficiency in this challenging keyword spotting task. Specifically, we concluded the following main points:

- The GAF representation effectively encoded temporal patterns, allowing the model to capture long-range dependencies in the audio data, which are crucial for accurate keyword detection.
- The model demonstrated 89.98% precision performance in identifying instances of the target criminal keywords, showcasing its capability to discriminate between positive and negative examples effectively.

The findings of this research have practical implications for real-world applications, particularly in the development of criminal keyword detection for the Amharic-speaking community. Moreover, the accurate and efficient keyword spotting capabilities of the proposed model open up possibilities for a range of applications, including voice search, transcription services, and assistive technologies, catering to the diverse needs of Amharic speakers.

In conclusion, the successful application of deep learning techniques and the GAF representation to the Amharic language demonstrates the potential for advancing speech technology in under-resourced language settings. This research contributes valuable insights to the field of keyword

spotting and lays the foundation for future advancements in Amharic speech processing. The developed CNN-LSTM model offers a promising framework for keyword spotting in Amharic and sets a precedent for further research and innovations in speech recognition for other under-resourced languages. However, the developed promising framework is not evaluated with long audio files, more than one second, containing the criminal keyword. So that this major limitation needs to be addressed for further insights. Moreover, the number of audio files contained in the prepared dataset needs to increase and the model should be evaluated.

6.2 Recommendation

Based on the findings and insights obtained from this research, several recommendations are proposed to further enhance the keyword spotting system for the Amharic language and advance the field of speech recognition in under-resourced languages.

1. **Exploration of Ensemble Models:** Investigate the use of ensemble models that combine multiple CNN-LSTM architectures with diverse hyper-parameter settings. Ensemble techniques can potentially improve the overall robustness and generalization capabilities of the keyword spotting system by leveraging the strengths of different model variations.
2. **Data Augmentation Strategies:** Explore advanced data augmentation techniques tailored to Amharic audio data, such as noise injection, time-frequency masking, and pitch variation. Augmenting the dataset with realistic variations can improve the model's ability to handle challenging acoustic conditions and contribute to more reliable keyword detection.
3. **Transfer Learning and Pre-training:** Consider leveraging transfer learning and pre-training approaches on larger multilingual datasets. Pre-training the CNN-LSTM model on similar languages or general speech data can help bootstrap the model's performance and potentially reduce the need for extensive labeled data in low-resource settings.
4. **Phonetic and Semantic Information Integration:** Investigate the incorporation of phonetic and semantic information into the keyword spotting pipeline. By leveraging linguistic knowledge, the model can gain a deeper understanding of the Amharic language's unique characteristics, leading to more contextually aware keyword spotting.

5. **Extension to Other Under-Resourced Languages:** Extend the research to explore the applicability of the CNN-LSTM model and the GAF representation in keyword spotting for other under-resourced languages. The insights gained from this study could be valuable for researchers working on speech technology for diverse linguistic communities.

Implementing these recommendations can further improve the effectiveness and applicability of the keyword spotting system for the Amharic language, paving the way for more accessible and advanced speech technologies in under-resourced language settings.

References

[1] D. Mulfari, A. Celesti and M. Villari, "Exploring AI-based Speaker Dependent Methods in Dysarthric Speech Recognition," *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2022, pp. 958-964, doi: 10.1109/CCGrid54584.2022.00117.

[2] López-Espejo I, Tan ZH, Hansen JH, Jensen J. Deep spoken keyword spotting: An overview. IEEE Access. 2021 Dec 30; 10:4169-99.

[3] Fernández, Santiago, Alex Graves, and Jürgen Schmidhuber. "An application of recurrent neural networks to discriminative keyword spotting." *International Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2007.

[4] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in Proceedings of Special Interest Group on Information Retrieval (SIGIR), 2007, vol. 7, pp. 51–57.

[5] David RH Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A Lowe, Richard M Schwartz, and Herbert Gish, "Rapid and accurate spoken term detection.," in INTERSPEECH, 2007, pp. 314–317.

[6] Siddika Parlak and Murat Saraclar, "Spoken term detection for Turkish broadcast news," in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2008, pp. 5244–5247.

[7] Mamou, Jonathan, Bhuvana Ramabhadran, and Olivier Siohan. "Vocabulary independent spoken term detection." In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 615-622. 2007.

[8] Brhanemeskel, Getnet Mezgebu, et al. "Amharic Speech Search Using Text Word Query Based on Automatic Sentence-like Segmentation." *Applied Sciences* 12.22 (2022): 11727.

[9] Sainath, Tara N., and Carolina Parada. "Convolutional neural networks for small-footprint keyword spotting." In *Interspeech*, pp. 1478-1482. 2015.

[10] Lei, Jie, et al. "Low-power audio keyword spotting using Tsetlin machines." *Journal of Low Power Electronics and Applications* 11.2 (2021): 18.

[11] Awaid, Mostafa, Sahar A. Fawzi, and Ahmed H. Kandil. "Audio Search Based on Keyword Spotting in Arabic Language." *International Journal of Advanced Computer Science and Applications* 5.2 (2014).

[12] Rohlicek, Jan Robin, et al. "Continuous hidden Markov modeling for speaker-independent word spotting." *International Conference on Acoustics, Speech, and Signal Processing*, IEEE, 1989.

[13] R. Rose and D. Paul, "A hidden Markov model-based keyword recognition system," in Proceedings of ICASSP 1990 – 15th IEEE International Conference on Acoustics, Speech and Signal Processing, April 3-6, Albuquerque, USA, 1990, pp. 129–132.

[14] J. Wilpon, L. Miller, and P. Modi, "Improvements and applications for key word recognition using hidden Markov modeling techniques," in Proceedings of ICASSP 1991 – 16th IEEE International Conference on Acoustics, Speech and Signal Processing, April 14-17, Toronto, Canada, 1991, pp. 309–312.

[15] Abate, S.T.; Menzel, W. "Syllable-based speech recognition for Amharic." In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Prague, Czech Republic, 28–29 June 2007; pp. 33–40.

[16] Abate, S.T. "Automatic Speech Recognition for Amharic". Ph.D. Thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, Hamburg, Germany, 2006.

[17] Brhanemeskel, Getnet Mezgebu, et al. "Amharic Speech Search Using Text Word Query Based on Automatic Sentence-like Segmentation." Applied Sciences 12.22 (2022): 11727.

[18] Chaudhary, A.; Akshatha, K.; Kodlekere, K.; Prasad, S.J. "Keyword based indexing of a multimedia file". In Proceedings of the 2017 IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, 11–13 December 2017; pp. 573–576.

[19] Chen, Guoguo, Carolina Parada, and Georg Heigold. "Small-footprint keyword spotting using deep neural networks." *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014.

[20] Abdelmoula, Ramzi. "Noise robust keyword spotting using deep neural networks for embedded platforms." MS thesis. University of Waterloo, 2016.

[21] Audhkhasi, Kartik, et al. "End-to-end ASR-free keyword search from speech." *IEEE Journal of Selected Topics in Signal Processing* 11.8 (2017): 1351-1359.

[22] Prabhavalkar, Rohit, et al. "Automatic gain control and multi-style training for robust small-footprint keyword-spotting with deep neural networks." *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[23] Zhehuai Chen, Yanmin Qian, and Kai Yu, "Sequence discriminative training for deep learning based acoustic keyword spotting," Speech Communication, 2018.

[24] George Retsinas, Giorgos Sfikas, Nikolaos Stamatopoulos, Georgios Louloudis, and Basilis Gatos, "Exploring critical aspects of CNN-based keyword spotting. a PHOCNet study," in IAPR, 2018, pp. 13–18.

[25] Santiago Ndez, Alex Graves, J Schmidhuber, and rgen, "An application of recurrent neural networks to discriminative keyword spotting," in ICANN, 2009, pp. 220– 229.

[26] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, "Attention-based end-to-end models for smallfootprint keyword spotting," in INTERSPEECH, 2018, pp. 2037–2041.

[27] Yanzhang He, Rohit Prabhavalkar, Kanishka Rao, Wei Li, Anton Bakhtin, and Ian McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," in ASRU, 2017, pp. 474–481.

[28] Wang, Xiong, Sining Sun, and Lei Xie. "Virtual adversarial training for ds-cnn based small-footprint keyword spotting." *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019.

[29] Arik, Sercan O., Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky, Chris Fougner, Ryan Prenger, and Adam Coates. "Convolutional recurrent neural networks for small-footprint keyword spotting." *arXiv preprint arXiv:1703.05390* (2017).

[30] Sun, Ming, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyros Matsoukas, Nikko Strom, and Shiv Vitaladevuni. "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting." In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 474-480. IEEE, 2016.

[31] Michaely, Assaf Hurwitz, Xuedong Zhang, Gabor Simko, Carolina Parada, and Petar Aleksic. "Keyword spotting for Google assistant using contextual speech recognition." In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 272-278. IEEE, 2017.

[32] Wang, Zhiguang, and Tim Oates. "Imaging time-series to improve classification and imputation." arXiv preprint arXiv:1506.00327 (2015).

[33] Saifullah, Khalid, Raunaque Mujeeb Quaiser, and Nadeem Akhtar. "Voice Keyword Spotting on Edge Devices." In *2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, pp. 1-5. IEEE, 2022.

[34] Kavya, H.P. and Karjigi, V., 2014, October. Sensitive keyword spotting for crime analysis. In 2014 IEEE National Conference on Communication, Signal Processing and Networking (NCCSN) (pp. 1-6).

[35] Zheng, N. and Li, X., 2010, June. A robust keyword detection system for criminal scene analysis. In 2010 5th IEEE Conference on Industrial Electronics and Applications (pp. 2127-2131).

[36] Xu, Hongji, Juan Li, Hui Yuan, Qiang Liu, Shidi Fan, Tiankuo Li, and Xiaojie Sun. "Human activity recognition based on Gramian angular field and deep convolutional neural network." IEEE Access 8 (2020): 199393-199405

[37] Eiben AE, Jelasity M. A critical note on experimental research methodology in EC. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) 2002 May 12 (Vol. 1, pp. 582-587). IEEE.