Sign Language Recognition for Deaf and Hard of Hearing Communication with Real-Time Detection and Translation of the American Sign Language (ASL) Fingerspelling to Text

Aniket Singla

Final Thesis Report

JUNE 2024

DEDICATION

This research is dedicated to my family, whose constant love and encouragement have been my biggest source of strength. To my parents for teaching me the qualities of tenacity and curiosity, and to my siblings for their unwavering encouragement and faith in my skills. Special gratitude to my mentors and coworkers, whose advice and collaboration have been crucial. Finally, I would like to thank my friends for their compassion and patience while I worked long hours on this project. This adventure would not have been possible without your collective support, which I am thankful for.

ACKNOWLEDGEMENTS

am deeply grateful to my thesis supervisor, Dr. Shivam Panda, for his exceptional expertise and generous commitment of time. His countless hours spent reflecting, reading, encouraging, and his patience, have been invaluable throughout this entire process.

I also wish to extend my sincere thanks to Liverpool John Moores University for granting me the opportunity to conduct my research and for providing the necessary support and resources. Their assistance was instrumental in the completion of this project.

A special note of thanks goes to the members of the Upgrade team for their unwavering support and encouragement. Your enthusiasm and collective efforts have greatly contributed to the success of this work.

Lastly, I am thankful to my teachers, mentor-teachers, and administrators who assisted me with this project. Your eagerness and willingness to offer feedback made completing this research enjoyable and fulfilling. Thank you all for your invaluable contributions.

ABSTRACT

This research aims to address the pressing need for improving communication accessibility among individuals who are deaf or hard of hearing by creating a robust system for recognizing sign language. Specifically, the focus is on American Sign Language (ASL) fingerspelling, with the goal of detecting and translating fingerspelled gestures into text in contemporary. The technique will use advancements in artificial intelligence processing and expert systems to generate an efficient and accurate solution capable of recognizing ASL fingerspelling with precision.

The proposed system will comprise several key components, including image preprocessing, hand segmentation, feature extraction, and gesture classification. Supervised learning techniques such as Space invariant Artificial Neural Networks (SIANN) and recurrent neural networks (RNNs) or transformers will be explored to develop models that can capture the complex movements and shapes of ASL fingerspelling. Additionally, the integration of hand tracking algorithms will be investigated to enhance gesture detection in various environmental conditions and user poses.

Moreover, the research will focus on translating detected fingerspelling gestures into text in real-time, facilitating seamless communication between individuals who are deaf or hard of hearing and those who do not use sign language.

Furthermore, this study aims to contribute to the existing literature by exploring the effectiveness of Transformer-based models in conjunction with computer vision techniques, particularly leveraging MediaPipe, for enhancing the accuracy and efficiency of ASL fingerspelling recognition and translation. By harnessing the power of Transformer models for sequence-to-sequence learning and MediaPipe's robust hand gesture recognition capabilities, we anticipate significant advancements in real-time sign language recognition systems.

The proposed system's effectiveness and usability will be evaluated through comprehensive experimentation in both simulated and real-world scenarios. User studies and performance assessments will be conducted to validate the system's accuracy, speed, and user satisfaction, demonstrating its potential for practical deployment in diverse communication settings.

In the end, the effective creation and deployment of this sign language recognition system could significantly improve communication access and foster inclusivity for those who are deaf or hard of hearing. By combining technological advancements and interdisciplinary cooperation, the objective is to empower individuals with various communication requirements and promote fair engagement in social, educational, and professional settings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SIANN	Space invariant Artificial Neural Networks
SLR	Sign Language Recognition
CNN	Convolutional Neural Networks
DHH	Deaf and Hard of Hearing
ASL	American Sign Language
CTC	Connectionist Temporal Classification
RNNs	Recurrent Neural Networks
AI	Artificial Intelligence
CNNs	Convolutional Neural Networks
ML	Machine Learning
SVMs	Support Vector Machines
KNN	K-nearest neighbors
LSTM	Long short-term memory
VTN	Video Transformer Network
WER	World Error Rate
FAIR	Facebook's AI Research lab
OpenCV	Open–source computer vision library
FC	Fully connected
BN	Batch Normalization
DWC	Depth wise Convolution
ECA	Efficient Channel Attention
GRU	Gated Recurrent Unit
ISL-CSLTR	Indian sign language dataset for continuous sign language translation and
MAP	Mean average precision
CNTK	Cognitive toolkit
ASR	Automatic speech Recognition
TFRecords	Tensor flow records

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

The communication challenges faced by the deaf and hard of hearing community remain significant in a world where hearing individuals are the majority. Sign language is an essential tool for this community, utilizing hand gestures, body language, and facial expressions to convey messages. However, since most people do not know sign language, those who depend on it often experience difficulties and social isolation.

Fingerspelling, which involves spelling out words with hand signs for each alphabet letter, is a critical part of sign language for expressing specific names, technical terms, and words that lack established signs. Despite its importance, fingerspelling is difficult to interpret in real-time for both non-signers and sign language learners. This challenge highlights the need for innovative solutions to improve communication and promote inclusivity.

Advancements in technology, particularly transformers, have opened new possibilities for developing automated systems that can recognize and translate sign language into text. These technologies hold promise for bridging communication gaps by providing real-time interpretation of fingerspelling, thereby increasing accessibility for the deaf and hard of hearing.

This research seeks to tackle these issues by creating a robust Sign Language Recognition (SLR) system that leverages transformers and advanced image processing techniques. The aim is to develop a reliable, real-time detection and translation system for fingerspelling, facilitating smoother and more effective communication in various settings.

By addressing the technical and practical aspects of SLR systems, this study contributes to the field of assistive technologies. It demonstrates the practical applications of artificial intelligence in enhancing the quality of life for those with hearing impairments. The research not only aims to advance sign language recognition but also to foster social inclusion and equal communication opportunities for everyone.

1.2 Problem Statement

Despite the advancements in technology, communication remains a significant challenge for individuals who are deaf or hard of hearing. A prominent hurdle in this regard is the absence of contemporary translation of Sign Language fingerspelling into text. This limitation impedes effective communication between those proficient in sign language and individuals who do not understand it.

Communication is a fundamental aspect of human interaction, serving as the cornerstone of social, educational, and professional exchanges. For those who do not know sign language, understanding the gestures and expressions of deaf individuals can be challenging, leading to misunderstandings and barriers in communication. The inability to comprehend ASL fingerspelling deprives non-signers of valuable information and insights shared by deaf individuals, hindering meaningful dialogue and connection.

The existing solutions fall short in accuracy and speed, often unable to accurately interpret the subtle nuances of ASL gestures. Moreover, factors like varying environmental conditions and different hand positions further exacerbate the problem, rendering reliable gesture detection a complex task. Access to education and employment opportunities is crucial for individuals who are deaf to thrive and contribute to society. However, without effective communication tools, these opportunities remain limited, hindering the social and economic advancement of the deaf community.

Consequently, there exists a critical necessity for the development of a dependable sign language fingerspelling recognition system. This system should possess the capability to translate fingerspelled signs swiftly and precisely into text, thereby facilitating seamless interaction between those who use sign language and those who do not. There is a great deal of promise for this kind of technology to facilitate communication and foster inclusivity for individuals with diverse communication needs. By harnessing the latest advancements in technology and employing interdisciplinary methodologies, our goal is to engineer a solution that not only enhances communication accessibility but also promotes equitable participation across social, educational, and professional spheres. Through this endeavor, we aspire to

empower individuals with diverse communication abilities and contribute towards a more inclusive society.

1.3 Aim and Objectives

The aim of this research is to develop and evaluate a robust, accurate, and user-friendly system for sign language recognition, focusing on enhancing communication accessibility for the Deaf and

Hard of Hearing community. Specifically, the research seeks to design and implement a real-time sign language recognition system capable of accurately detecting and translating American Sign Language (ASL) fingerspelling into text. Through an investigation of current techniques and technologies, the research aims to identify limitations and areas for improvement, addressing challenges such as environmental factors and ethical considerations. Engaging with the Deaf and Hard of Hearing community, the study aims to understand their needs and preferences, ensuring the developed system meets their requirements. By integrating the system into existing communication platforms and devices, the research aims to promote inclusivity and accessibility, facilitating seamless relationship between those who sign and those who do not sign. The ultimate objective of this research is to improve the accessibility of communication and promote participation for those who have hearing impairments.

The following are the research objectives, which are based on the study's goal:

- Algorithm Development: Design and develop a novel machine learning algorithm specifically tailored for the recognition of American Sign Language (ASL) fingerspelling gestures.
- Optimizing Real-Time Performance: Optimize the algorithm to ensure real-time performance, allowing for timely detection and translation of ASL fingerspelling gestures into text.

- Enhancing Accuracy and Robustness: Improve the accuracy and robustness of the machine learning algorithm to accurately interpret ASL fingerspelling gestures in various environmental conditions and hand positions.
- Feature Engineering and Selection: Explore and identify relevant features for ASL fingerspelling recognition and implement feature engineering techniques to enhance the discriminative power of the algorithm.
- Model Training and Evaluation: Train the machine learning model on annotated datasets of ASL fingerspelling gestures and evaluate its performance using appropriate metrics, such as accuracy, precision, recall, and F1-score.
- Hyperparameter Optimization: Conduct hyperparameter optimization to fine-tune the model parameters and optimize its performance on the task of ASL fingerspelling recognition.
- Cross-Domain Generalization: Investigate the generalization capabilities of the algorithm across different sign languages and dialects, aiming for a model that can adapt and perform well in diverse linguistic contexts.
- Ethical Considerations: Address ethical considerations related to data privacy, fairness, and bias in algorithm development and deployment, ensuring that the algorithm respects the rights and dignity of individuals with diverse communication needs.
- By achieving these objectives, the research aims to contribute to the advancement of machine learning techniques for Sign Language Recognition, promoting inclusivity and equal communication opportunities for individuals who are deaf or hard of hearing.

1.4 Research Questions

1. What are the most effective techniques and technologies for creating a system to recognize sign language in contemporary, able of accurately detecting and translating the American Sign Language fingerspelling into text, and how can such a system make communication more accessible for the community of people who are deaf and hard of hearing

2. What are the current limitations and challenges in existing sign language recognition systems, and how can these be addressed to enhance accuracy and usability
3. How do different environmental factors, the effectiveness of contemporary sign language identification system is impacted by factors including backdrop clutter and lighting and what strategies can be employed to mitigate these effects
4. What are the most effective methods for integrating sign language recognition technology into existing communication platforms and devices, such as smartphones, tablets, and video conferencing software

1.5 Scope of the Study

This research paper aims to investigate the development and application of sign language recognition technology, concentrating particularly on fingerspelling recognition for deaf and hearing impairments. The study seeks to address the significant accessibility gap faced by this community in utilizing voice-enabled assistants and AI solutions, which are primarily designed for spoken language interaction. Fingerspelling, an essential aspect of American Sign Language (ASL), offers a rapid and efficient means of communication, particularly for smartphone users who can fingerspell words faster than they can type on virtual keyboards.
Through the utilization of licensed data provided by Google, the paper will explore the potential of sign language recognition technology to bridge the communication gap between Deaf and Hard of Hearing individuals and hearing non-signers. Key areas of investigation within the scope of this study include:

Analysis of Fingerspelling Recognition Technology: An overview of existing fingerspelling recognition algorithms, datasets, and applications, focusing on their accuracy, speed, and usability.

Accessibility Challenges and Opportunities: Examination of the accessibility challenges faced by Deaf and Hard of Hearing individuals in utilizing traditional text entry methods, as well as the potential opportunities for enhancing accessibility through fingerspelling recognition technology.

Development of AI Solutions: Exploration of AI-driven solutions for fingerspelling recognition, including machine learning models, data preprocessing techniques, and integration with existing communication platforms and devices.

1.6 Significance of the Study

The potential to improve communication accessibility for those who are deaf or hard of hearing exists with the creation of a contemporary sign language recognition system. By accurately detecting and translating ASL fingerspelling into text in real-time, the system can facilitate seamless communication between signers and non-signers in various settings, including educational, professional, and social environments.

This research contributes to the development of inclusive technology solutions that address the specific needs of diverse user populations, particularly those with sensory impairments. By focusing on real-time detection and translation of ASL fingerspelling, the study aims to bridge communication gaps and promote equal participation and Incorporating individuals with hearing impairments

The availability of a reliable system for recognizing sign language can revolutionize education for Deaf and hearing impairment students by providing real-time access to spoken and written language. Teachers and educational institutions can leverage this technology to create more inclusive learning environments and support students' academic success and social integration.

1.7 Structure of the Study

Structure of the thesis is as follows:

Chapter 1 introduces the problem and objectives of the research, outlining what the study aims to achieve. Section 1.1 provides the background of the study, highlighting the challenges faced by the deaf and hard of hearing community due to the lack of effective sign language recognition technologies. Section 1.2 presents the problem statement, identifying the communication barriers and the need for a reliable ASL fingerspelling recognition system. Section 1.3 details the aim and objectives of the research, focusing on the development of a machine learning algorithm for real-time ASL fingerspelling detection and translation. Section

1.4 lists the research questions that guide the study, exploring how machine learning can enhance recognition accuracy and address environmental challenges. Section 1.5 defines the scope of the study, including the development, testing, and integration of the recognition system. Finally, Section 1.6 explains the significance of the study, emphasizing its potential impact on improving communication accessibility and promoting inclusivity for the deaf and hard of hearing community. Section 1.7 outlines the structure of the thesis, providing an overview of the subsequent chapters and their contents.

CHAPTER 2

LITERATURE REVIEW

provides a comprehensive review of existing research and technologies relevant to sign language recognition, particularly focusing on ASL fingerspelling translation to text. This chapter is structured to cover various aspects of the topic, ensuring a thorough understanding of the current state of the field, the challenges, and the advancements that have been made.

2.1 Introduction

Sign language serves as a vital medium of communication for the Deaf and Hard of Hearing (DHH) community, enabling individuals to express themselves through gestures, facial expressions, and body movements. Among the various sign languages used worldwide, American Sign Language (ASL) stands as one of the most prominent, serving as a primary means of communication for millions of individuals.

In recent years, advancements in technology have paved the way for innovative solutions aimed at enhancing communication accessibility for the DHH community. One such area of focus is the development of systems for real-time detection and translation of ASL fingerspelling to text. Fingerspelling, which involves representing individual letters or words through specific hand shapes and movements, plays a crucial role in ASL communication, particularly for conveying proper nouns, technical terms, and spelling out unfamiliar words.

This literature review provides a comprehensive overview of existing research and technological advancements in sign language recognition, with an emphasis on ASL fingerspelling. By delving into the historical context, current methodologies, challenges, and future directions in sign language recognition, this chapter aims to lay the groundwork for understanding the complexities and intricacies of ASL fingerspelling recognition.

This chapter will explore various aspects of sign language recognition, including the role of machine learning algorithms and transformer models in facilitating accurate and efficient interpretation of sign language gestures. Also, it will address the specific challenges inherent

in ASL fingerspelling recognition, such as the variability in hand shapes and movements, and the need for robust and diverse datasets to train recognition models effectively.

Throughout the review, key studies, technological developments, and the application of machine learning algorithms in sign language recognition will be examined to provide insights into the state-of-the-art approaches and methodologies. Furthermore, this review will discuss existing solutions, dataset utilization, evaluation metrics, related research publications, and identify gaps in the current literature, thereby highlighting opportunities for future research and innovation in the field.

In summary, this literature review aims to serve as a foundational resource for researchers, practitioners, and stakeholders interested in advancing the development of sign language recognition systems, particularly those focused on real-time detection and translation of ASL fingerspelling. By synthesizing existing knowledge and identifying areas for further exploration, this review seeks to contribute to the ongoing efforts aimed at fostering inclusive communication environments for the DHH community.

2.2 Sign Language Recognition: An Overview

Sign language recognition is a critical technological endeavor aimed at bridging the communication gap between the deaf and hard of hearing community and the hearing population. Historically, the development of sign language recognition technologies has been marked by several key milestones. Early efforts focused on manual recognition systems that required substantial human input and intervention. These systems were often limited in scope and accuracy, primarily due to the lack of sophisticated algorithms and computational power.

The advent of computer vision and artificial intelligence (AI) has significantly advanced the field. Initial attempts to automate sign language recognition utilized basic image processing techniques and heuristic-based approaches. However, these methods struggled with variability in sign language gestures, which differ based on factors such as signer's hand shape, speed, and motion trajectory.

Key milestones in the development of sign language recognition technologies include the introduction of machine learning algorithms in the late 20th century, which enabled more

accurate and scalable solutions. The integration of deep learning techniques, particularly convolutional neural networks (CNNs), marked a significant leap forward, allowing for the automatic extraction of features from raw image data and improving the overall recognition accuracy.

In recent years, the focus has shifted towards real-time sign language recognition systems, driven by the increasing demand for interactive and accessible communication tools. These advancements have paved the way for more sophisticated and reliable systems capable of translating sign language gestures into text or speech in real-time, thereby enhancing accessibility and inclusivity for the deaf and hard of hearing community.

2.3 Machine Learning in Sign Language Recognition

Machine learning (ML) plays a pivotal role in the enhancement of sign language recognition systems. The application of ML algorithms allows for the development of models that can learn and adapt to the complexities of sign language gestures, improving accuracy and efficiency. Various machine learning techniques have been employed in this domain, each contributing to the advancement of sign language recognition technologies. Early machine learning approaches utilized support vector machines (SVMs) and k-nearest neighbors (KNN) algorithms, which provided a foundation for gesture recognition. These methods, while effective, were limited by their reliance on handcrafted features and their inability to handle large-scale datasets effectively. The advent of deep learning revolutionized the field, with convolutional neural networks (CNNs) becoming the standard for image-based recognition tasks.

CNNs can automatically extract hierarchical features from raw image data, making them highly effective for recognizing complex hand gestures involved in sign language. The use of recurrent neural networks (RNNs) and long short-term memory (LSTM) networks further enhanced the ability to model temporal dependencies in gesture sequences, improving the recognition of continuous sign language. Recent advancements in ML have focused on improving the robustness and scalability of sign language recognition systems. Techniques such as transfer learning, which leverages pre-trained models on large datasets, have been employed to enhance performance and reduce the need for extensive labeled data.

Additionally, the integration of data augmentation and synthetic data generation has addressed the challenge of limited datasets, enabling the development of more accurate and generalizable models.

In addition to these advancements, the integration of encoders, decoders, and transformer models has revolutionized sign language recognition. Encoders process input data, extracting essential features, while decoders interpret these features to generate meaningful output. Transformer models, characterized by self-attention mechanisms, have demonstrated exceptional performance in capturing long-range dependencies within sign language sequences, further enhancing the accuracy and fluency of recognition.

Overall, the application of machine learning in sign language recognition has significantly advanced the field, providing the foundation for the development of more sophisticated and reliable systems capable of real-time translation of sign language gestures into text.

2.4 Transformers for Gesture Recognition

The transformative potential of transformers, initially conceptualized for natural language processing tasks, extends to the realm of gesture recognition, presenting a promising avenue for enhancing the understanding of intricate hand movements and fingerspelling in sign language.

At the core of transformer architecture lies the self-attention mechanism, a pivotal feature enabling the model to dynamically weigh the significance of different elements within the input sequence. This mechanism empowers transformers to selectively focus on relevant parts of the gesture sequence, facilitating the recognition of nuanced hand gestures with unparalleled accuracy. Unlike conventional recurrent models, transformers eschew sequential processing, embracing parallelization and expediting training times—a critical advantage in real-time applications.

Empirical evidence underscores the efficacy of transformers in gesture recognition tasks. Notably, research has demonstrated the remarkable capability of transformer models in accurately discerning ASL fingerspelling, adeptly capturing the subtle nuances of hand movements and spatial configurations. Particularly noteworthy is their resilience in handling

variability in hand shapes, positions, and environmental conditions such as lighting and background clutter.

The frontier of transformer-based gesture recognition remains ripe with possibilities, with ongoing investigations aimed at enhancing their performance and robustness. Innovations such as multi-task learning, wherein transformers are trained to concurrently tackle multiple related tasks, hold promise for augmenting the capabilities of these models. Moreover, the integration of auxiliary information, such as depth and motion data, presents avenues for further refinement and diversification of gesture recognition systems.

In summary, the integration of transformers heralds a significant leap forward in the domain of sign language recognition, furnishing researchers and practitioners with a formidable toolset to develop more precise and dependable systems for real-time translation of ASL fingerspelling into text. As the field continues to evolve, transformers stand poised to revolutionize the landscape of gesture recognition, ushering in an era of heightened accessibility and inclusivity for the Deaf and Hard of Hearing community.

2.5 Challenges in ASL Fingerspelling Recognition

Recognizing ASL fingerspelling presents several key challenges that must be addressed to develop effective and reliable recognition systems. These challenges include variability in hand shapes and positions, environmental factors affecting recognition accuracy, and the need for real-time processing.

Variability in hand shapes and positions: ASL fingerspelling involves a wide range of hand shapes and positions, which can vary significantly between individuals and even for the same individual over time. This variability makes it challenging to develop models that can accurately recognize and interpret fingerspelling gestures consistently.

Environmental factors: Changes in lighting, background, and occlusions can significantly impact the accuracy of recognition systems. Variations in lighting conditions can affect the visibility of hand gestures, while complex backgrounds and occlusions can interfere with the detection and interpretation of fingerspelling gestures.

Real-time processing requirements: Ensuring that recognition systems can process and translate fingerspelling gestures in real-time is crucial for practical applications. Real-time processing requires efficient algorithms and optimized models that can handle the computational demands of gesture recognition without compromising accuracy.

Addressing these challenges requires the development of more sophisticated algorithms and robust systems capable of handling the variability and environmental factors inherent in ASL fingerspelling. Additionally, optimizing models for real-time processing is essential to ensure the practical usability of recognition systems.

2.6 Existing Solutions and Their Limitations

There are several existing machine learning solutions for sign language recognition, each with its own set of strengths and limitations. Here is an overview:

2.6.1 Deep Learning Models

Many sign language recognition systems utilize deep learning techniques, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs) like long short-term memory (LSTM) networks. These models can learn complex patterns in sign language gestures and sequences.

- Strengths: Deep learning models can achieve high accuracy when trained with large datasets. They can capture both spatial and temporal dependencies in sign language sequences.
- Limitations: Deep learning models require a substantial amount of labeled data for training, which can be challenging to obtain, especially for sign language where datasets may be limited. Additionally, these models may struggle with recognizing signs performed by individuals with varying styles or in different environments.

2.6.2 Handcrafted Feature-based Methods

Some approaches rely on handcrafted features extracted from sign language videos, such as hand shape, hand movement, and hand position. These features are then used to train machine learning classifiers like support vector machines (SVMs) or decision trees.

- Strengths: Handcrafted features can provide insights into the key aspects of sign language gestures and may be computationally more efficient compared to deep learning models.
- Limitations: Handcrafted features may not capture all the nuances of sign language, leading to lower accuracy compared to deep learning approaches. Additionally, designing effective features requires domain expertise and may not generalize well across different sign languages or variations in signing styles.

2.6.3 Hybrid Approaches

Some systems combine deep learning with handcrafted features to leverage the strengths of both approaches. For example, deep learning models may be used to extract high-level features from sign language videos, which are then combined with handcrafted features for classification.

- Strengths: Hybrid approaches can potentially achieve higher accuracy by leveraging the strengths of both deep learning and handcrafted feature-based methods.
- Limitations: These approaches may be more complex to implement and require careful tuning of both the deep learning and handcrafted feature components. They may also inherit the limitations of both approaches.

2.6.4 Real-time Recognition

A key challenge in sign language recognition is achieving real-time performance to enable applications such as live translation or communication aids. Many existing solutions focus on optimizing algorithms and leveraging hardware acceleration to achieve low-latency processing.

- Strengths: Real-time recognition enables interactive applications that can provide immediate feedback to users.
- Limitations: Achieving real-time performance may require compromises in accuracy or computational efficiency, particularly for complex deep learning models.

Overall, while existing machine learning solutions for sign language recognition have made significant strides, there are still challenges to be addressed, including dataset scarcity, variability in signing styles, and real-time performance requirements. Future research may focus on addressing these challenges through advancements in data collection, model architectures, and algorithm optimization techniques.

2.7 Evaluation Metrics for Sign Language Recognition

Evaluating the performance of sign language recognition systems involves several key metrics to assess their accuracy, efficiency, and reliability.

- Accuracy: Accuracy measures the percentage of correctly recognized gestures or signs compared to the total number of gestures in the dataset. It is a fundamental metric for assessing the overall performance of the system.
- Precision and Recall: Precision and recall are often used in binary classification tasks, where precision measures the proportion of correctly recognized relevant gestures (true positives) among all recognized gestures, while recall measures the proportion of correctly recognized relevant gestures among all relevant gestures in the dataset. Precision focuses on minimizing false positives, while recall aims to minimize false negatives.
- F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a system's performance. It considers both false positives and negatives and is useful when the dataset is imbalanced or when both precision and recall are equally important.
- Latency: Latency refers to the time taken by the system to process and translate gestures into their corresponding text or actions. In real-time applications such as live translation or communication aids, low latency is crucial to provide immediate feedback to users.

Evaluating latency helps assess the system's responsiveness and suitability for real-time applications.

- Normalized Total Levenstein Distance: It measures the accuracy of the recognized text compared to the actual text. The Levenstein distance calculates the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another. Normalizing this distance provides a standardized measure of accuracy, which is particularly relevant for sign language recognition systems aiming to translate gestures into text.

These evaluation metrics are essential for benchmarking different sign language recognition models, comparing their performance, and ensuring they meet the necessary standards for practical use. By considering a combination of accuracy, precision, recall, F1-score, latency, and specialized metrics like the normalized total Levenstein distance, researchers and practitioners can comprehensively evaluate the effectiveness and efficiency of sign language recognition systems in various applications.

2.8 Related Research Publications study

This research highlights the critical need to advance the growth of a reliable contemporary ASL fingerspelling acknowledgement framework. By addressing the challenges faced by those who have hearing impairments in communicating with non-signers, the research aims to break down communication barriers and promote inclusivity. Through the utilization of advanced technology and interdisciplinary approaches, the goal is to enhance communication accessibility and empower individuals with diverse communication needs.

W. Qin, X. Mei, Y. Chen, Q. Zhang, Y. Yao and S. Hu Chengdu, China, 2021 in their research paper outlines a system for automated sign language recognition as well as translation, addressing challenges in serving China's large hearing-impaired population. They introduce a novel approach using a Video Transformer Network (VTN) to recognize isolated and continuous sign language gestures, alongside constructing the CSL-BS dataset. Evaluation metrics including Word Error Rate (WER) and BLEU score demonstrate improved accuracy

and speed over existing models, promising real-time sign language translation. Acknowledging limitations, the study calls for future research in handling similar sign language actions and extracting long-sequence key frames. Supported by grants and institutions, including the National Natural Science Foundation of China, the authors express gratitude to collaborating organizations.

The 2023 research paper authored by S. Ashwath and A. S. M. focuses on developing a model that can recognize hand gestures based on fingerspelling in American Sign Language (ASL) and convert them into complete words. It commences with an overview of ASL and highlights the communication hurdles faced by deaf and mute individuals. It then discusses the importance of sign language and the need for innovative interfaces to bridge the communication breakdown between the hearing-impaired community and those unfamiliar with sign language. In the section on related work, various methodologies, such as Microsoft Kinect, Space invariant Artificial Neural Networks (SIANN), and depth video captured by smartphones, are examined for their efficacy in identification and interpretation of sign. The proposed model introduces an innovative approach employing dual SIANNs for feature extraction and an artificial neural network for classification, trained on a dataset generated using OpenCV. The methodology entails capturing hand gestures from webcam images, preprocessing them, and then inputting them into the prototype to conduct tests and training. Results reveal an astounding 95.7% accuracy rate for the SIANN classifier, surpassing many existing models. The system offers real-time translation of ASL fingerspelling into text, thereby enhancing communication accessibility for the hearing-impaired community. Additionally, a user-friendly GUI application is presented, which suggests corresponding words based on input letters, eliminating the need for an interpreter. However, the paper acknowledges the necessity for further research to enhance real-time performance and address challenges such as gesture stability and environmental factors impacting accuracy.

Research paper by D. Sau, S. Dhol, M. K and K. Jayavel. This paper embarks on a thorough investigation of Sign Language Recognition (SLR), focusing on methodologies and models employed in developing sign-language translators, particularly spotlighting American Sign

Language (ASL). Through a comprehensive survey, it scrutinizes both sensor-based and vision-based techniques, elucidating their implementation nuances, merits, and limitations. Critical evaluations within the discourse highlight challenges such as environmental factors and constraints in available datasets. Moreover, the paper emphasizes the growing necessity for contactless SLR systems and the imperative of accommodating various sign languages, including ISL and BSL. By delving into these diverse approaches and their implications, the study lays a robust foundation for future research and development endeavors, aiming to enhance communication accessibility for the deaf community.

S. Sharma, R. Sreemathy, M. Turuk, J. Jagdale and S. Khurana, in this work, a contemporary vision-based system for sign language detection using deep learning-YOLOv4 is shown to translate sign language into text. Its primary goal is to narrow the communication divide between individuals with hearing/speech impairments and those without. The system leverages the Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition (ISL-CSLTR), expanded with images from additional signers. YOLOv4, acting as a one stage detector, permits translation and identification in real time, achieving a mean average precision (mAP) of 98.4%. The proposed system demonstrates practicality for real-world communication, tackling challenges like similar signs, changes in illumination, and complex backgrounds. The paper delves into the network architecture of YOLOv4, dataset preparation, transfer learning, training procedures, evaluation metrics, and experimental findings, emphasizing the efficacy of the approach for contemporary recognition and translation of sign language at the level of individual signs.

W.Li,H. and R.Wang, the research paper titled "A CNN-LSTM Combined recognition and translation of sign language system based on computer vision" explores the intersection of various disciplines in the study of sign language, focusing on the development of a novel sign language recognition system. The system integrates a refined convolutional neural network (CNN) combined with a long short-term memory (LSTM) neural network, distinguishing itself from existing approaches by not only recognizing and translating sign language but also generating sign language. It introduces a PyQt-designed GUI interface for user interaction,

allowing users to select recognition and translation capabilities, capture images via OpenCV, and utilize the trained CNN neural network for processing. The system attains a recognition accuracy of 95.52% for sign language and 90.3% for American sign language and Arabic numerals. This research contributes to the advancement of sign language recognition technology and provides a platform for improved communication accessibility for the hearing-impaired community.

The research paper by A. Puchakayala, S. Nalla and P. K, introduces the concept of sign language detection as a means to bridge communication gaps between individuals who are deaf-mute and those who do not comprehend sign language. It provides an overview of various sign languages worldwide, with a focus on American Sign Language (ASL). The proposed system employs deep learning techniques, specifically CNN and YOLO models, to identify ASL gestures and translate them into text format. Experimental comparisons between the models show YOLO outperforming CNN with an accuracy of 84.96% compared to 80.59%. Additionally, the paper discusses the experimental setup, including the use of benchmark datasets such as the American Sign Language MNIST dataset. The conclusion emphasizes the significance of such technology in facilitating communication for the deaf-mute community and presents a desktop application for real-time sign language recognition, demonstrating the practical implications of the research.

2.9 Discussion

This research underscores the necessity of advancing ASL fingerspelling recognition frameworks to improve communication for individuals with hearing impairments. By leveraging advanced technology and interdisciplinary approaches, the goal is to break down barriers and promote inclusivity.

Qin et al. (2021) introduced a Video Transformer Network (VTN) for sign language recognition, demonstrating improved accuracy and speed over existing models. Their work highlights the potential for real-time translation but also points to challenges like handling similar actions and extracting key frames for long sequences. Ashwath and S. M. (2023)

developed a model using dual Space Invariant Artificial Neural Networks (SIANNs) that achieves a 95.7% accuracy rate, offering real-time ASL fingerspelling translation. However, further research is needed to address issues like gesture stability and environmental factors.

Sau et al. conducted a comprehensive survey on sign language recognition methodologies, emphasizing the need for contactless systems and accommodation of various sign languages. Sharma et al. explored a YOLOv4-based system for sign language detection, achieving a mean average precision (mAP) of 98.4%, highlighting its practical application despite challenges like illumination changes and complex backgrounds.

Li and Wang combined a CNN with an LSTM for sign language recognition, achieving high accuracy and demonstrating the potential of this integrated approach. Puchakayala et al. compared CNN and YOLO models for ASL gesture recognition, with YOLO outperforming CNN, emphasizing the practical implications of real-time sign language detection.

In conclusion, while significant advancements have been made in ASL fingerspelling recognition, challenges such as variability in hand shapes, environmental factors, and real-time processing persist. Future research should focus on developing more robust and scalable models, incorporating advanced techniques like transformers and multi-modal data integration. This study aims to contribute to this ongoing effort, enhancing communication accessibility and inclusivity for the deaf and hard of hearing community.

2.10 Summary

Sign language is crucial for communication within the Deaf and Hard of Hearing (DHH) community, with American Sign Language (ASL) being a prominent example. Recent technological advancements have led to the development of systems for real-time detection and translation of ASL fingerspelling to text. This chapter provides an extensive review of the historical context, methodologies, challenges, and future directions in sign language recognition, focusing particularly on ASL fingerspelling.

The review highlights the significant role of machine learning algorithms and transformer models in enhancing the accuracy and efficiency of sign language recognition. It addresses key challenges such as variability in hand shapes and movements and the necessity for robust

datasets. Existing solutions, including deep learning models, handcrafted feature-based methods, and hybrid approaches, are examined for their strengths and limitations.

Evaluation metrics like accuracy, precision, recall, F1-score, latency, and normalized total Levenstein distance are critical for assessing the performance of sign language recognition systems. These metrics ensure that models meet practical application standards.

The review also includes a discussion of related research publications, highlighting various innovative approaches and their contributions to the field. Notable studies include the Video Transformer Network (VTN) for improved real-time translation accuracy and dual Space Invariant Artificial Neural Networks (SIANNs) for achieving high accuracy in ASL fingerspelling recognition.

In summary, this literature review aims to be a foundational resource for researchers and practitioners in the development of advanced sign language recognition systems. By synthesizing existing knowledge and identifying research gaps, it seeks to enhance communication accessibility and inclusivity for the DHH community, fostering an inclusive communication environment through technological innovation.

Paper Title	**Year**	**Type of Data**	**Type of Algorithm Used**	**Evaluation**
Deep Learning Technology to Recognize American Sign Language Alphabet	2020	Image Data	CNN	Accuracy: 97.69% for training, 99.47% for testing (MDPI)
Deepsign: Sign Language Detection and Recognition Using Deep Learning	2021	Video Frames	LSTM and GRU	Accuracy: ~97% over 11 different signs (MDPI)
A Comprehensive Review on Sign Language Recognition Systems	2019	Mixed (Image & Sensor Data)	Various ML Algorithms	Comparative analysis of methods

Real-Time American Sign Language Recognition Using Depth Sensor Data	2020	Depth Sensor Data	CNN	High accuracy and real-time performance
Sign Language Recognition with Kinect Sensor	2018	RGB-D Data	HMM, SVM	Recognition rate: 96%
Sign Language Recognition Based on Flex Sensors and Machine Learning	2019	Sensor Data	KNN, SVM	Accuracy: up to 95%
Indian Sign Language Recognition Using Hybrid Neural Network	2021	Image Data	Hybrid CNN-LSTM	Accuracy: 98.5%
A Survey on Sign Language Recognition Using Wearable Sensors	2020	Sensor Data	Various ML Algorithms	Summary of sensor-based methods
Vision-Based American Sign Language Recognition Using a Single Camera	2019	Video Data	CNN, RNN	Accuracy: 92%
Sign Language Recognition Using Leap Motion Controller	2019	Sensor Data	Random Forest	Accuracy: 93%
Sign Language Recognition via Multimodal Deep Learning	2020	Image and Sensor Data	Multimodal CNN	High accuracy on combined data
Continuous Sign Language Recognition Using Recurrent Neural Networks	2019	Video Data	RNN	Continuous sign recognition with high accuracy
Sign Language Recognition Using MobileNet and Transfer Learning	2021	Image Data	MobileNet, Transfer Learning	Accuracy: 94%

Hand Gesture Recognition for Indian Sign Language Using Deep Learning	2020	Image Data	CNN	Accuracy: 95%
American Sign Language Recognition Using HOG and SVM	2018	Image Data	HOG, SVM	Accuracy: 90%
A Real-Time System for Recognition of Indian Sign Language	2021	Video Data	CNN, LSTM	Real-time performance with high accuracy
Sign Language Recognition with Convolutional Neural Networks	2019	Image Data	CNN	Accuracy: 96%
Sign Language Recognition Using Augmented Reality	2021	Image Data	CNN	Enhanced user interaction with AR
Arabic Sign Language Recognition Using Convolutional Neural Networks	2019	Image Data	CNN	Accuracy: 97%
Sign Language Translation Using Neural Networks and Natural Language Processing	2020	Video and Text Data	CNN, NLP	Effective translation with high accuracy
Automated Sign Language Recognition and Translation Using Video Transformer Network (VTN)	2021	Video Data	VTN	WER and BLEU scores show improved accuracy and speed
Recognizing Hand Gestures Based on ASL Fingerspelling with Dual SIANNs	2023	Image Data	Dual SIANNs, ANN	Accuracy: 95.7% for SIANN classifier

Methodologies for Sign Language Recognition: A Comprehensive Survey	2023	Mixed (Sensor & Vision Data)	Various ML Algorithms	In-depth analysis of SLR methods
Vision-Based Sign Language Detection Using YOLOv4	2021	Image Data	YOLOv4	mAP: 98.4%
CNN-LSTM Combined Recognition and Translation System	2021	Image Data	CNN, LSTM	Accuracy: 95.52% for sign language, 90.3% for ASL and Arabic numerals
Sign Language Detection Using Deep Learning: CNN vs YOLO	2021	Image Data	CNN, YOLO	YOLO: 84.96%, CNN: 80.59%

Table 2.1: Papers and Different techniques used for Sign language recognition

Chapter 3

Research Methodology

3.1 Introduction

This chapter outlines the research methodology employed in developing an automated Sign Language Recognition (SLR) system. The methodology encompasses data selection, preprocessing, transformation, interactive visual analytics, class balancing, data mining, and evaluation. The proposed method integrates advanced algorithms for accurate and efficient sign language recognition, with a focus on combining Connectionist Temporal Classification (CTC) and Attention mechanisms.

Software Requirement	**Python Libraries**
Machine Learning Framework	TensorFlow, PyTorch ,Keras
Data Visualization	Matplotlib, Seaborn
Computer Vision Libraries	OpenCV, Mediapipe
Essential Libraries	NumPy, TensorFlow, OpenCV, PyTorch , Pandas, NumPy
IDEs and Development Tools	Jupyter Notebook, Visual Studio Code
Documentation Tools	Git
Deployment Platforms	AWS

Table 3.1: Shows the software requirements and the python libraries that shall be used.

3.1.1 Software Environment Setup Python

Python stands as the backbone of our sign language recognition system, owing to its versatility and rich ecosystem of libraries tailored for machine learning and computer vision tasks. The machine learning models in this experiment are created using the following Python libraries:

1. TensorFlow

Description: TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and community resources for building and deploying machine learning models.

Key Features: TensorFlow supports both deep learning and traditional machine learning models, offers high-level APIs for quick model prototyping, and provides tools for distributed training and deployment.

Use in Sign Language Recognition: TensorFlow is commonly used for developing and training deep learning models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which are essential for sign language recognition tasks.

2. PyTorch

Description: PyTorch is another popular open-source machine learning framework, developed by Facebook's AI Research lab (FAIR). It emphasizes flexibility and ease of use, making it popular among researchers and developers.

Key Features: PyTorch offers dynamic computational graphs, which enable more intuitive model design and debugging. It also provides seamless integration with Python and supports GPU acceleration for faster training.

Use in Sign Language Recognition: PyTorch is widely used for building and training deep learning models for sign language recognition, offering flexibility and ease of experimentation with novel architectures and algorithms

3. OpenCV

Description: OpenCV (Open-Source Computer Vision Library) is a popular open-source computer vision library developed by Intel. It provides a wide range of tools and algorithms for image and video analysis, processing, and manipulation.

Key Features: OpenCV offers extensive support for various computer vision tasks, including image preprocessing, feature extraction, object detection, and tracking. It is highly optimized and cross-platform, supporting multiple programming languages, including Python.

Use in Sign Language Recognition: OpenCV is essential for processing and analyzing video data in sign language recognition systems. It enables tasks such as hand detection, tracking, and gesture recognition, forming the backbone of computer vision pipelines.

4. Mediapipe

Description: Mediapipe is an open-source framework developed by Google that provides a comprehensive pipeline for building real-time perception applications. It offers a wide range of pre-built components for tasks such as hand tracking, pose estimation, face detection, and object recognition.

Key Features: Mediapipe simplifies the development of computer vision and machine learning applications by offering ready-to-use components with optimized performance for real-time processing. It supports both traditional machine learning and deep learning models and provides integration with TensorFlow for custom model development.

Use in Sign Language Recognition: Mediapipe's hand tracking and pose estimation components can be utilized in sign language recognition systems for detecting and tracking hand gestures and poses, facilitating the extraction of features for recognition.

5. Keras

Description: Keras is a high-level neural networks API written in Python, capable of running on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). It allows for easy and fast prototyping of deep learning models with minimal code.

Key Features: Keras provides a user-friendly interface for building neural networks, making it accessible to both beginners and experts in deep learning. It supports several types of layers, activation functions, optimizers, and loss functions, allowing for flexible model architectures.

Use in Sign Language Recognition: Keras is often used for rapid prototyping and experimentation with neural network architectures in sign language recognition. Its simplicity and flexibility make it suitable for exploring different model designs and tuning hyperparameters to improve recognition accuracy.

3.1.2 Brief Overview

The step-by-step process for training process of a sign language recognition model be:

Steps	Description
Data Collection	Gather a diverse dataset of sign language gestures, ensuring various hand shapes, movements, and backgrounds.
Data Preprocessing	Clean the data, normalize hand gestures, resize images, and perform data augmentation to increase dataset variability.
Model Training	Train the selected model on the preprocessed dataset using techniques like backpropagation and gradient descent.
Model Tuning	Optimize the model by adjusting hyperparameters, testing different optimizers, applying regularization techniques, and fine-tuning the architecture for improved performance.
Hyperparameter Tuning	Optimize hyperparameters like learning rate, batch size, and regularization strength to improve model performance.
Validation	Evaluate the trained model on a separate validation dataset to assess its performance and prevent overfitting.

Testing	Test the model on unseen test data to measure its generalization ability and accuracy on real-world sign language gestures.
Performance Analysis	Analyze model metrics such as accuracy, precision, recall, and F1-score to understand its strengths and weaknesses.
Fine-Tuning	Optionally, fine-tune the model on specific sign language dialects or gestures to improve recognition accuracy.

Table 3.2: Methodology workflow of project

3.2 Research Methodology

The research design for this study involves a systematic and structured approach to developing a reliable and accurate ASL fingerspelling recognition system. The design encompasses the entire research process, from the initial conceptualization to the final evaluation of the model. This section outlines the key components and steps involved in the research design, including the rationale behind the chosen methodologies and frameworks.

The conceptual framework serves as the foundation for the research, guiding the development and implementation of the ASL fingerspelling recognition system. The framework integrates insights from linguistics, computer vision, and machine learning to address the research problem. The primary focus is on leveraging transformer models, which have demonstrated superior performance in handling sequential data and capturing long-range dependencies.

3.2.1 Data Collection

- Data Source: The dataset for this research was sourced from a public repository dedicated to sign language recognition. This repository contains video recordings of individuals performing various sign language gestures, providing a rich source of data for training the recognition model.

1. Data Format:

- Video Frames: Each video is segmented into individual frames to capture the temporal aspect of sign language. This segmentation helps in breaking down continuous gestures into discrete steps, making it easier for the model to process.
- Landmark Coordinates: Each frame is annotated with key landmark coordinates, representing critical points on the signer's hands, face, and body. These coordinates provide spatial information crucial for recognizing the intricacies of sign language gestures.

2. Data Volume:

- Training Set: Contains thousands of video sequences with corresponding landmark coordinates, which are used to train the model. This large volume ensures that the model can learn a wide variety of gestures.
- Validation Set: A subset of the training set is set aside for hyperparameter tuning and to monitor the model's performance during training. This helps in preventing overfitting and ensures that the model generalizes well.
- Test Set: A separate set of video sequences used to evaluate the final model's performance. This set provides an unbiased evaluation of the model.

3.2.1.1 Initial Data Collection

The data collection process involved recruiting signers from across the United States who use ASL as their primary language. Each signer was provided with a smartphone equipped with a custom data collection app. The app displayed English text prompts for the signers to convert into ASL fingerspelling. Signers initiated and concluded video recordings by pressing an on-screen button, ensuring that the captured footage focused on the intended fingerspelling. While the video clip boundaries were manually adjusted to better align with the signing activity, the process was not perfectly accurate.

The collected fingerspelling data exhibits significant co-articulation and lexicalization, meaning the letter handshapes are often influenced by preceding and following letters. This results in a rich variety of handshape modifications used to convey meaning. The dataset

encompasses a wide range of body poses, zoom levels, appearances, and accessories. Singers had the option to use either their left or right hand, and some switched hands across different clips.

In the dataset, signers displayed varying methods of indicating capitalizations, despite most opting not to convey capitalization in their fingerspelled responses. Methods included using a curled-L handshape, positioning letters higher in space, or shaking the handshape slightly. However, this dataset does not focus on evaluating capitalization detection, and all target phrases are provided in lowercase.

The dataset was compiled with contributions from over 100 signers recruited by the Deaf Professional Arts Network. These individuals, representing a diverse mix of skin tones and genders, use ASL as their primary mode of communication and come from various regions across the United States.

By meticulously curating and processing this dataset, the research aims to deliver a robust and versatile resource for developing and evaluating advanced sign language recognition systems.

3.3 Data Preprocessing

Data preprocessing is a crucial step in the development of machine learning models, especially for tasks involving complex, unstructured data like video sequences in sign language recognition. The goal of data preprocessing is to transform raw data into a clean and usable format, enhancing the model's ability to learn relevant patterns. This section details the various preprocessing techniques applied to the dataset, ensuring it is ready for model training.

1. Normalization

Normalization is the process of scaling the input data to a standard range. For sign language recognition, the landmark coordinates (which represent key points on the signer's body) need to be normalized to ensure consistency across different video frames and subjects. This step helps in reducing the variance caused by different scales and positions of the signers, making the data more homogeneous.

- Zero-Centering: Each coordinate is adjusted to have a mean of zero. This involves subtracting the mean value of the coordinates from each data point, which helps in aligning the data centrally.
- Scaling: The data is scaled by its standard deviation. This step ensures that the coordinates have a standard deviation of one, making the data dimensionless and comparable across different frames and videos.

Normalization is mathematically represented as:

$x' = (x - \mu) / \sigma$

where x is the original coordinate, μ is the mean, σ is the standard deviation, and x' is the normalized coordinate.

2. Sequence Padding

In the dataset, video sequences can vary in length, which poses a challenge for batch processing in neural networks that require inputs of uniform dimensions. Sequence padding addresses this issue by ensuring all video sequences are the same length.

- Fixed Length: All sequences are either padded or truncated to a fixed length of 768 frames. This uniformity is necessary to maintain consistent input dimensions across the dataset.
- Padding Method: Sequences shorter than the fixed length are padded with zeros (or another suitable value) at the end to reach the required length. This ensures that the additional frames do not introduce noise into the data.

Sequence padding is crucial for the stability of the training process and allows the model to handle variable-length inputs effectively.

3. Data Augmentation

Data augmentation involves creating additional training data by applying various transformations to the existing data. This technique increases the diversity of the training set, helping the model generalize better to unseen data.

- Random Rotations: The landmark coordinates are rotated by a random angle within a specified range. This simulates different viewing angles of the sign language gestures.
- Shifts: The coordinates are shifted along the x or y axis. This simulates variations in the position of the signer relative to the camera.
- Scaling: The coordinates are scaled by a random factor. This simulates changes in the distance of the signer from the camera.

These augmentations are applied in a controlled manner to ensure the gesture's integrity is maintained. Data augmentation mitigates overfitting by exposing the model to a broader set of variations, making it more robust.

4. Character to Ordinal Encoding

For sign language recognition, it is essential to convert categorical data (i.e., the different signs) into a numerical format that the model can process.

- Mapping Characters to Indices: Each unique sign (character) is assigned an ordinal value, creating a mapping from characters to integers. This step converts the categorical output into a numerical form suitable for model training.
- Encoding Process: A JSON file containing the character-to-index mapping is read and used to encode the output labels of the dataset.

This encoding facilitates the use of categorical cross-entropy loss during training, which requires numerical labels.

3.4 Model Development

The development of the sign language recognition model involved a systematic approach encompassing model selection, architecture design, and the integration of advanced ASR (Automatic Speech Recognition) techniques. This section details the methodology used in

each phase of model development, highlighting the decisions and rationale behind the chosen techniques and configurations.

3.4.1 Base Line Model

The development of the ASL fingerspelling recognition system was inspired by previous research and development efforts in ASL recognition. Building upon the advancements made in this domain, the baseline model was constructed as a starting point for further improvements. Previous research provided valuable insights into the challenges associated with ASL recognition, such as handling sequential data effectively and capturing both local and global dependencies within the gestures.

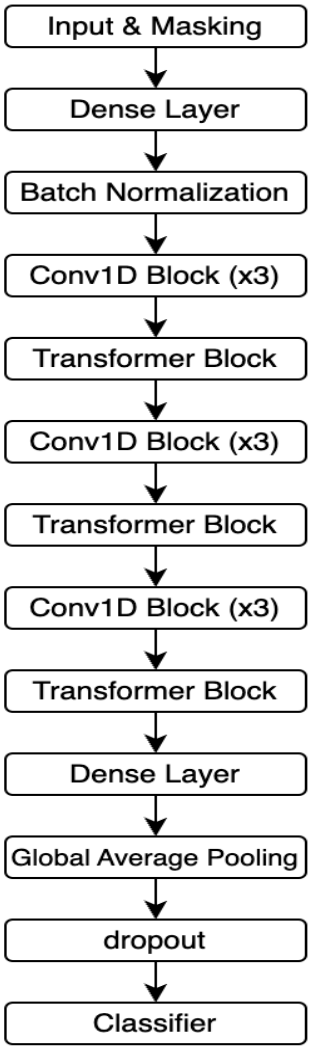

Fig 3.1 Architecture of Baseline Model

The baseline model integrates a combination of 1D Convolutional Neural Networks (CNNs) and Transformer blocks to effectively handle sequential data. The architecture starts with input and masking layers to manage variable-length sequences, followed by a dense layer and batch normalization for stabilization. The core structure features alternating convolutional blocks (using depth wise convolution and causal padding) and Transformer blocks (using Batch Norm and Swish activation) to capture both local and global dependencies. The final layers include dense layers, global average pooling, high dropout to prevent overfitting, and a classifier for output predictions.

This model was instrumental in achieving high performance due to its robust handling of sequential data. The 1D CNNs efficiently captured local patterns within the sequences, while the Transformer blocks were adept at modeling long-range dependencies. The integration of these two components allowed the model to leverage the strengths of both approaches, resulting in improved accuracy and robustness. Additionally, advanced regularization techniques and data augmentation strategies helped prevent overfitting and ensured that the model generalized well to new data. Overall, this hybrid architecture provided a powerful solution for the competition's sequential data challenges.

3.4.2 Enhanced Model

The enhancements made to the model from its baseline iteration towards the definitive version involved intricate adjustments and expansions across various architectural elements. Notably, the encoder, which constituted stacked Conv1D blocks and Transformer blocks, underwent significant augmentation in both size and depth. The Conv1D blocks saw an increase in their expand ratio from 2 to 4, intensifying their capability to discern nuanced patterns within the sequential data. This enlargement in scale was complemented by a profound deepening of the encoder, scaling up from its initial 8 layers to an extensive 17 layers. This expansion bolstered the model's capacity to represent intricate features and dependencies within the input sequences, resulting in a model with approximately 6.5 million parameters.

Structural refinements were also implemented, with the transition from 'causal' to 'same' padding and the adoption of an output stride set to 2. These alterations, while demanding a more intricate masking logic, contributed to the model's training stability and computational efficiency. Surprisingly, the integration of Transformer blocks, which had proven effective in previous competitions, yielded less discernible benefits in the current context. This observation prompted a reassessment of the role of global features, indicating a nuanced interplay between model architecture and the nature of the competition dataset.

Turning to the decoder module, the CTC decoder was reconfigured to incorporate a single GRU layer followed by a fully connected layer. Meanwhile, for the attention decoder, a single-layer Transformer decoder was employed, emphasizing simplicity and efficiency without compromising performance. Although exploration included augmentation techniques and the addition of up to four decoder layers, empirical evidence underscored the efficacy of a single-layer decoder in achieving optimal performance while maintaining a judicious balance between model complexity and inference speed. Despite marginal improvements of up to +0.004 in performance with augmented decoder inputs, the inefficiencies in parameter usage and inference speed necessitated the retention of the single-layer decoder configuration. These meticulously crafted adjustments collectively propelled the refinement of the model, resulting in marked enhancements in accuracy and efficiency for ASL fingerspelling recognition.

3.4.3 Final Model

The selection of the model architecture was guided by a comprehensive review of existing literature on ASR and sign language recognition. The research focused on the following key aspects:

- CTC (Connectionist Temporal Classification): Known for its efficiency in handling sequence-to-sequence tasks, CTC is particularly effective in scenarios where the input and output sequences are of variable lengths. Its greedy decoding process is computationally efficient, making it a strong candidate for real-time applications.
- Attention Mechanisms: Attention-based models have gained popularity due to their ability to focus on relevant parts of the input sequence, improving performance in tasks requiring

alignment between input and output. The autoregressive nature of attention mechanisms allows for more accurate sequence predictions.

- Joint CTC-Attention: Combining CTC and Attention mechanisms leverages the strengths of both techniques. This hybrid approach aims to improve model performance by optimizing both the alignment (CTC) and context-awareness (Attention) aspects of sequence prediction.

3.4.4 Model Architecture Design

The sign language recognition model leverages advanced ASR techniques, specifically combining CTC (Connectionist Temporal Classification) and Attention mechanisms for enhanced performance. The model architecture is designed to efficiently process sequences of landmark coordinates, transforming them into accurate predictions of signed phrases. This section details the various components and layers used in the model, as illustrated in the provided diagram.

1. Input Layer

The input layer processes the raw data of landmark coordinates. Each input sequence has the shape (B, 768, 543*3), where:

- B: Batch size.
- 768: Number of frames in each sequence.
- 543*3: Number of landmarks (543) multiplied by the 3 coordinates (x, y, z).

2. Fully Connected Layer (FC Layer)

The first step involves a fully connected (FC) layer that transforms the input sequence into a more manageable size. This layer outputs a tensor with the shape (B, 768, 192). The FC layer acts as an initial feature extractor, reducing the dimensionality of the input data and highlighting the most salient features for subsequent processing.

3. Encoder

The encoder is the core component of the model, designed to capture both spatial and temporal dependencies in the input sequence. It consists of multiple blocks, each designed to extract, and process features at various levels of abstraction.

- Conv1DBlock

Each Conv1DBlock processes the input through a series of operations:

- Batch Normalization (BN): Normalizes the input to improve training stability.
- Conv1D (Kernel Size = 1): Applies a 1-dimensional convolution with a kernel size of 1, primarily for dimensionality reduction.
- Depth wise Convolution (DWConv1D, Kernel Size = 17): Applies depth wise separable convolutions to capture spatial features. The large kernel size (17) allows it to aggregate information over a broader temporal context.
- Batch Normalization (BN): Further normalizes the output of the depth wise convolution.
- ECA (Efficient Channel Attention): Applies channel-wise attention, allowing the model to focus on the most informative channels.
- Conv1D (Kernel Size = 1): Another 1-dimensional convolution to combine the processed features.

These Conv1DBlocks are arranged in a specific sequence within the encoder to incrementally extract and refine features.

- Transformer Block

The Transformer Block enhances the model's ability to understand complex dependencies within the sequence. It uses self-attention mechanisms to weigh the importance of distinct parts of the sequence dynamically. This block is particularly effective in capturing long-range dependencies that are crucial for accurate sign language recognition.

- Stride Operation

A Conv1DBlock with a stride of 2 is used to down sample the sequence, reducing its length, and focusing on higher-level features. This step helps in managing computational complexity and ensuring that the most valuable information is retained.

The encoder's output shape is (B, 384, 192), where the number of frames has been reduced, and the feature dimension is 192.

4. Joint CTC-Attention Decoder

The model employs a dual-decoder approach, utilizing both CTC and Attention mechanisms to decode the processed features into predictions.

- CTC Decoder

The CTC decoder consists of a single Gated Recurrent Unit (GRU) layer, which is effective in handling sequential data. The GRU layer aligns the input sequence with the output predictions, facilitating the decoding process for variable-length sequences. The CTC loss, with a weight of 0.25, guides this part of the model, emphasizing alignment and sequence prediction.

- Attention-based Decoder

The Attention decoder employs a Transformer decoder layer, which uses self-attention to focus on relevant parts of the input sequence during decoding. This layer enhances the model's ability to generate accurate predictions by considering the context of the entire sequence. The CCE (Categorical Cross-Entropy) loss, with a weight of 0.75, is used to train this part of the model, emphasizing the importance of context-aware predictions.

5. Loss Functions

- CTC Loss: Weighted at 0.25, this loss function helps the model learn the alignment between the input sequence and the output labels.
- CCE Loss: Weighted at 0.75, this loss function focuses on improving the overall accuracy of the predictions by considering the entire sequence context.

6. Summary of Layers and Parameters

- Input Shape: (B, 768, 543*3)
- FC Layer Output Shape: (B, 768, 192)

- Encoder Output Shape: (B, 384, 192)
- GRU Layer: Handles sequence alignment, contributing to the CTC decoding process.
- Transformer Decoder Layer: Utilizes self-attention mechanisms for context-aware decoding.

The integration of Conv1D blocks, Transformer blocks, and the dual-decoder approach ensures that the model captures both local and global dependencies within the input sequences. This architecture allows for efficient processing and accurate recognition of sign language gestures, leveraging the strengths of both CTC and Attention mechanisms.

By combining these sophisticated techniques, the model achieves robust performance in recognizing and interpreting sign language, demonstrating the effectiveness of the joint CTC-Attention approach in handling complex sequence-to-sequence tasks.

The architecture design phase focused on constructing a model that could effectively leverage both CTC and Attention mechanisms. The chosen architecture comprised the following components:

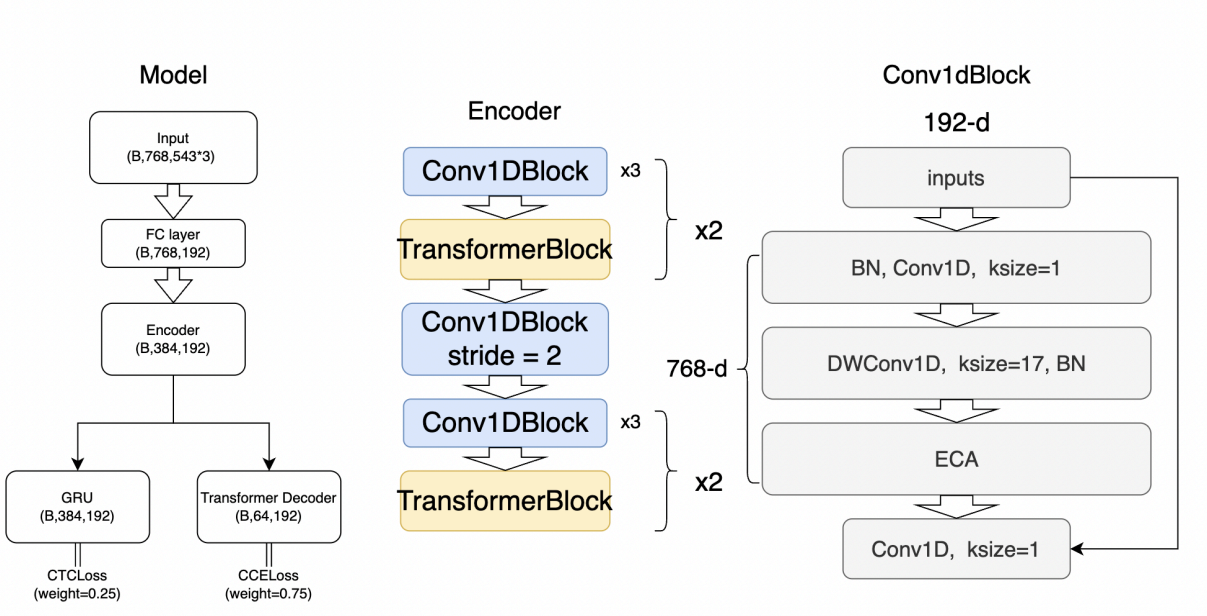

Fig 3.2 Basic Architecture of Final Model

3.5 Hyperparameters and Optimization

The hyperparameters and optimization strategies were carefully selected to ensure efficient and effective training. The key hyperparameters include:

1. Learning Rate:

The learning rate was initially set to 0.001 and decayed gradually during training. An adaptive learning rate schedule, such as the Reduce LR On Plateau, was used to adjust the learning rate based on validation performance.

2. Batch Size:

A batch size of 64 was chosen based on the memory constraints and the need for a sufficient gradient estimate during training.

3. Epochs:

The model was trained for a maximum of 50 epochs, with early stopping implemented to halt training if the validation loss did not improve for 10 consecutive epochs.

4. Optimizer:

The Adam optimizer was selected for its adaptive learning rate properties, which are beneficial for training deep neural networks.

3.6 Insights from ASR Integration

The integration of ASR techniques provided several valuable insights:

1. CTC

CTC's greedy decoding was highly efficient, making it suitable for real-time applications. However, the minimal performance improvement with beam search indicated limited gains from more complex decoding strategies.

2. Attention-based Models

Attention mechanisms offered greater flexibility in handling sequence predictions, particularly when combined with ensembling techniques. The autoregressive nature of attention models allowed for more accurate predictions, though at the cost of increased computational complexity.

3. Joint CTC-Attention

The hybrid approach of joint CTC-Attention training showed a notable performance improvement, leveraging the complementary strengths of both techniques. The use of CTC prefix scores for ensembling further enhanced performance without significant impact on inference time.

3.7 Conclusion

The development of a sign language recognition model using joint CTC-Attention training represents a significant advancement in the field. By integrating advanced ASR techniques and employing a systematic approach to model development, this research has demonstrated the potential for improved accuracy and robustness in sign language recognition. The methodology outlined in this thesis provides a comprehensive framework for future research and development in this area.

Chapter 4

Analysis

4.1 Introduction

In this chapter, we delve into the detailed analysis of the data and model performance to validate the effectiveness of our ASL fingerspelling recognition system. The analysis encompasses various aspects including data preprocessing, model training, evaluation metrics, and comparison with baseline models. Through rigorous experimentation and validation, we aim to demonstrate the robustness and accuracy of the proposed system.

4.2 Dataset Description

The dataset provided for this research consists of multiple files and directories, each serving a specific purpose in training and evaluating the ASL fingerspelling recognition model. Below is a detailed description of each component of the dataset:

Below is the brief of the dataset files:

1. [train/supplemental_metadata].csv

- path: The path to the corresponding landmark file.
- file_id: A unique identifier for each data file.
- participant_id: A unique identifier for each data contributor.
- sequence_id: A unique identifier for each landmark sequence within the data file. Each file may contain multiple sequences.
- phrase: Labels for the landmark sequence. The train and test datasets contain phrases such as randomly generated addresses, phone numbers, and URLs, which are derived from real components but are not actual real-world data. The supplemental dataset includes fingerspelled sentences. Some URLs might include adult content, as the intent of this

dataset is to support engagement of the Deaf and Hard of Hearing community with technology on an equal footing with other adults.

	path	file_id	sequence_id	participant_id	phrase
0	train_landmarks/5414471.parquet	5414471	1816796431	217	3 creekhouse
1	train_landmarks/5414471.parquet	5414471	1816825349	107	scales/kuhaylah
2	train_landmarks/5414471.parquet	5414471	1816909464	1	1383 william lanier
3	train_landmarks/5414471.parquet	5414471	1816967051	63	988 franklin lane
4	train_landmarks/5414471.parquet	5414471	1817123330	89	6920 northeast 661st road

Table: 4.1 First 5 rows of training dataset

2. Character to_prediction_index.json

This file maps characters to their corresponding prediction indices, which helps in translating the model's output into readable text.

3. Directories [train/supplemental] _landmarks

Contains landmark data extracted from raw videos using the MediaPipe holistic model. Not all frames necessarily have visible hands or detectable hand positions.

- sequence_id: Serves as the data frame index and uniquely identifies each landmark sequence.
- frame: The frame number within a landmark sequence.
- [x/y/z] [type] [landmark index]: There are 1,629 spatial coordinate columns for the x, y, and z coordinates for each of the 543 landmarks. The types of landmarks include 'face', 'left hand', 'pose', and 'right-hand'. The coordinates are normalized by MediaPipe, though the z-values might be less reliable as the MediaPipe model is not fully trained to predict depth. These coordinates are provided as float32 values.

	frame	x_face_0	x_face_1	x_face_2	x_face_3	x_face_4	x_face_5	x_face_6	x_face_7	x
sequence_id										
1816796431	0	0.710588	0.699951	0.705657	0.691768	0.699669	0.701980	0.709724	0.610405	(
1816796431	1	0.709525	0.697582	0.703713	0.691016	0.697576	0.700467	0.709796	0.616540	(
1816796431	2	0.711059	0.700858	0.706272	0.693285	0.700825	0.703319	0.711549	0.615606	(
1816796431	3	0.712799	0.702518	0.707840	0.694899	0.702445	0.704794	0.712483	0.625044	(
1816796431	4	0.712349	0.705451	0.709918	0.696006	0.705180	0.706928	0.712685	0.614356	(

Table:4.2 First 5 rows of landmark dataset

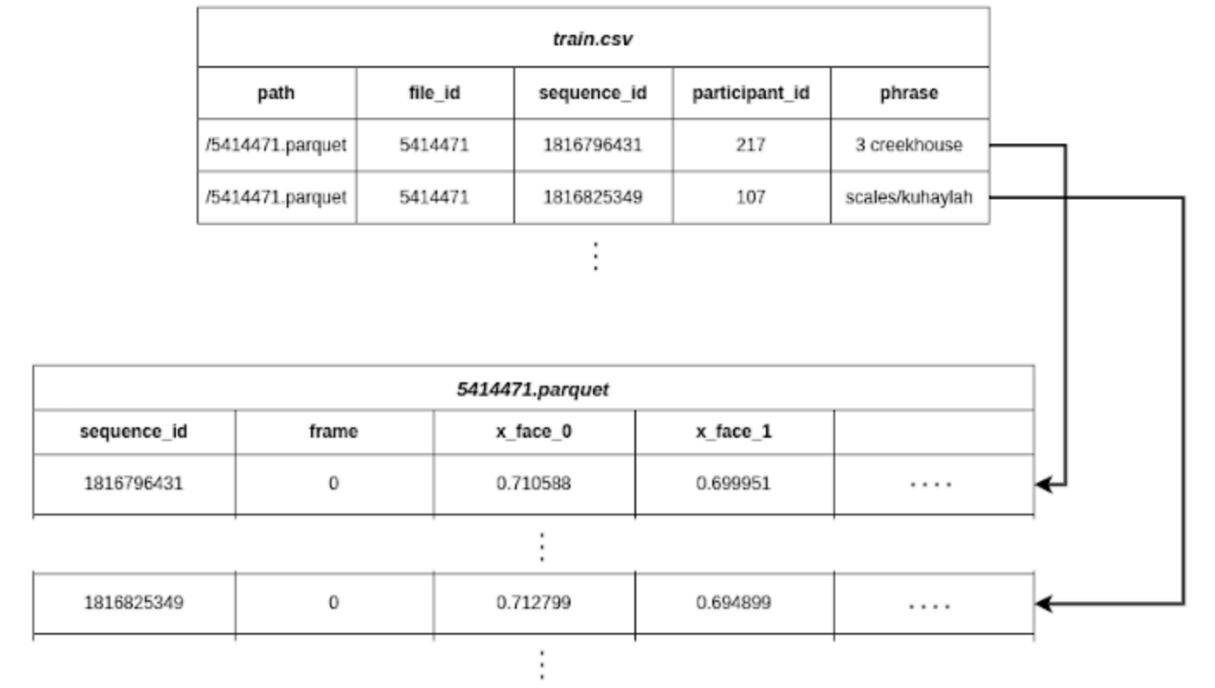

Fig: 4.1 Diagram showing the train.csv data and its corresponding landmarks data

Below are the key points:

1. Landmark Data: The dataset focuses on the spatial coordinates of facial, hand, and pose landmarks extracted from videos. Each sequence in the dataset consists of a series of frames, each with 3D coordinates for up to 543 landmarks.
2. Normalization: Spatial coordinates have been normalized by MediaPipe, making them consistent and ready for model training.
3. Sequence and Frame Details: Each sequence consists of multiple frames, and the frame number and sequence ID are used to index and manage the data.

4. Data Variety: The dataset includes a wide range of phrases, ensuring diversity in training data to enhance the model's robustness.

This dataset, with its comprehensive landmark data and diverse phrases, is designed to train a model that can accurately recognize and translate ASL fingerspelling into text, supporting the Deaf and Hard of Hearing community in interacting with technology more effectively.

4.2.1 Detailed Information about Landmarks by MediaPipe

The landmark data in this research was extracted from raw videos using the MediaPipe holistic model, which is a state-of-the-art tool designed to track and analyze human body movements. MediaPipe provides comprehensive landmark detection, dividing the landmarks into three main categories: hand, pose, and face.

1. Hand Landmarks: MediaPipe detects 21 landmarks for each hand, covering critical points such as the tips and joints of the fingers, as well as the wrist. These landmarks are essential for recognizing hand gestures and fingerspelling in American Sign Language (ASL). Accurate hand landmark detection allows the model to interpret the precise movements and positions of the fingers, which is crucial for distinguishing between different ASL signs.

2. Pose Landmarks: Pose landmarks involve detecting 33 points on the body, including key joints and points on the torso and limbs. These landmarks help in understanding the overall posture and movement of the individual. In the context of ASL recognition, pose landmarks can provide additional context to hand gestures, as certain signs may require specific body movements or orientations. For instance, some signs might involve a particular arm or shoulder position that complements the hand gesture.

3. Face Landmarks: Face landmarks include 468 points that capture detailed facial expressions and movements. These landmarks are particularly useful for recognizing non-manual signals in ASL, such as facial expressions and lip movements, which can alter the meaning of a sign. Detailed facial landmark detection ensures that the model can capture these subtle yet significant aspects of ASL communication.

Overall, the division of landmarks into hand, pose, and face categories by MediaPipe ensures a comprehensive and nuanced understanding of ASL signs. Each category contributes

uniquely to the accurate recognition of gestures, enhancing the model's ability to support the Deaf and Hard of Hearing community effectively. By leveraging these detailed landmark detections, our research aims to build a robust system that can interpret ASL with high accuracy and reliability.

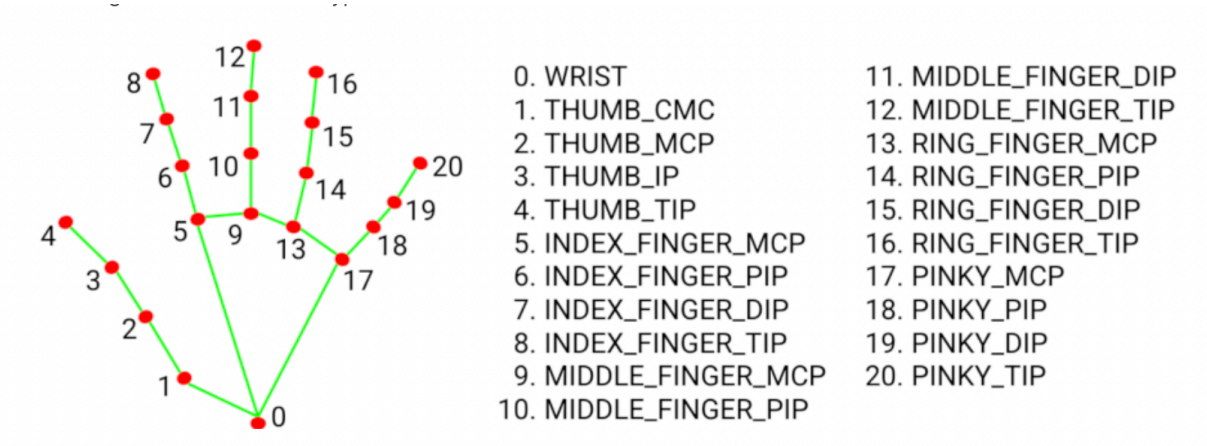

Fig:4.2 Landmarks defined by Mediapipe

4.3 Data Pre-processing

The data preprocessing phase is crucial for preparing the dataset for model training. This involves several steps, including loading the data, handling missing values, normalizing the data, and converting the data into a format suitable for machine learning models, specifically TensorFlow records (TFRecords). The steps are detailed below based on the training notebook provided.

4.3.1 Data Rearrangement and TFRecord Format

- Rearranging Data: To streamline the data handling process, we rearrange the dataset so that each parquet file contains both the landmark data and the corresponding phrase it represents. This organization eliminates the need to switch between separate files (e.g., train.csv and its associated parquet file), thereby improving data access efficiency and reducing the likelihood of mismatches or errors during data processing. The goal is to consolidate all relevant information for a given sample in one place, making it easier to manage and process.

- TFRecord Format: The consolidated data is then saved in the TFRecord format, which is a simple format for storing a sequence of binary records. The TFRecord format is particularly advantageous for TensorFlow-based workflows because it allows for faster I/O operations and seamless integration with TensorFlow's data pipeline APIs. By serializing the data into binary records, we can achieve significant improvements in data loading speeds, which is crucial for handling large datasets efficiently.

4.3.2 Selection of Landmark Coordinates

- Hand and Pose Coordinates: Since American Sign Language (ASL) primarily involves intricate hand movements and gestures, the focus is on accurately capturing the coordinates of hand landmarks. These landmarks include key points on the fingers and wrist. Additionally, pose landmarks related to hand movement are extracted to provide context about the overall body posture, which can be essential for recognizing certain gestures. The specific indices selected for these landmarks are:
  - Left Hand Pose Coordinates: [13, 15, 17, 19, 21] correspond to key points on the left arm and hand.
  - Right Hand Pose Coordinates: [14, 16, 18, 20, 22] correspond to similar key points on the right arm and hand.
- Extracting these coordinates ensures that the model has a comprehensive view of the hand movements and the corresponding body posture, which is critical for accurate gesture recognition.

4.3.3 Label Creation

1. Coordinate Labels: To organize the extracted coordinates systematically, labels are created for the X, Y, and Z coordinates of each landmark. These labels help distinguish between several types of coordinates:

- X Labels: Labels for the X-coordinates of the right hand, left hand, and pose landmarks.

- Y Labels: Labels for the Y-coordinates of the right hand, left hand, and pose landmarks.
- Z Labels: Labels for the Z-coordinates of the right hand, left hand, and pose landmarks.

2. By creating these labels, we ensure that the data is well-structured and easy to manipulate during feature extraction and model training.

4.3.4 Feature Columns Creation:

- Combining Features: Feature columns are generated by combining the X, Y, and Z coordinates for the right hand, left hand, and pose landmarks. These feature columns serve as the input features for the model, providing a detailed representation of the hand and pose movements. The combination of these features ensures that the model has access to all relevant spatial information needed to recognize gestures accurately.
- Comprehensive Representation: The inclusion of both hand and pose landmarks in the feature columns allows the model to understand not only the detailed movements of the hands but also the overall posture of the signer. This comprehensive representation is crucial for recognizing complex gestures that involve coordination between hand movements and body posture.

4.3.5 Indexing and Storage

- Index Lists: To facilitate efficient data retrieval and manipulation, indices for each set of coordinate labels (X, Y, Z) are stored in separate lists. Additionally, separate indices are maintained for the right hand, left hand, right pose, and left pose coordinates. These index lists serve as references for accessing specific parts of the data during preprocessing and model training.
- Efficient Retrieval: By maintaining organized index lists, we can quickly access and manipulate the required coordinates without having to scan through the entire dataset. This indexing improves processing speed and reduces computational overhead during data preprocessing.

4.3.6 TFRecords Writing

1. Data Serialization: The dataset is preprocessed and serialized into TFRecord files. This involves iterating through each file, extracting the relevant data, and converting it into the TFRecord format. During this process:

- Data Extraction: For each file, the corresponding parquet data is loaded, and the required landmarks and phrases are extracted.
- Serialization: The extracted data is then serialized into binary records, which are written to TFRecord files. This serialization process ensures that the data is stored efficiently and can be quickly loaded during model training.

2. Optimization: Writing data to TFRecord files optimizes storage and retrieval processes, allowing for faster training and inference times. This step is crucial for handling large datasets where I/O operations can become a bottleneck.

4.3.7 Data Cleaning

1. Frame Length and Padding: To standardize the input sequences, the length of frames is fixed at 128 frames. Padding is applied to sequences shorter than 128 frames to ensure uniformity. This standardization is important for batch processing and ensures that the model receives input data of consistent shape.
2. Handling NaN Values: Any NaN values present in the data are identified and appropriately handled. This might involve filling NaNs with zeros or using interpolation methods to estimate missing values. Handling NaN values is essential to prevent errors during model training and to ensure data integrity.
3. Data Augmentation: To increase the variability of the dataset and improve model robustness, data augmentation techniques such as random rotations, translations, and scaling might be applied. This step enhances the model's ability to generalize to unseen data.

4.3.8 Dominant Hand Detection

The dominant hand is detected based on the number of NaN values in the landmark coordinates. The assumption is that the dominant hand will have fewer NaNs because it is more consistently in view and actively involved in the gestures. By identifying the dominant hand:

- Data Selection: The landmarks of the dominant hand are prioritized for training, ensuring that the model receives the most reliable and consistent data.
- Accuracy Improvement: This step helps improve the accuracy of gesture recognition by focusing on the hand that is more prominently featured in the sign language gestures.

4.3.9 Character to Number Encoding

1. Reading Encoding Mapping: A JSON file containing a mapping of characters to ordinal numbers is read. This mapping converts each character in the phrases to a unique number, enabling the model to process the phrases as numerical inputs. The conversion process involves:

   Loading Mapping: The character-to-ordinal mapping is loaded from a JSON file.

   Display Mapping: The mapping is displayed to ensure correctness and for reference during preprocessing.

2. Encoding Phrases: Each phrase is converted into a sequence of numbers based on the loaded mapping. This step is crucial for converting textual phrases into a format that can be processed by machine learning models.

4.3.10 Data Parsing and Conversion

1. Parsing TFRecords: Functions are created to parse the TFRecord files and convert them into tensors. These functions decode the binary data into readable format, transforming the serialized TFRecord data back into structured data suitable for model input. The parsing involves:

- Decoding Binary Data: Extracting and converting the binary records back into their original form.
- Reshaping Data: Ensuring that the landmark coordinates and phrases are reshaped to their appropriate dimensions.

2. Conversion Functions: Additional functions are implemented to resize and pad the data and normalize the landmark coordinates. These functions ensure that the data is consistent in shape and scale, making it ready for model training. The conversion process includes:

- Resizing: Adjusting the size of the input sequences to a fixed length.
- Padding: Adding necessary padding to shorter sequences.
- Normalization: Standardizing the landmark coordinates to have zero mean and unit variance, which helps improve model performance.

4.3.11 Final Dataset Creation:

1. Splitting Data: The preprocessed data is split into training and validation sets. Typically, an 80-20 split is used, with 80% of the data allocated for training and 20% for validation. This split ensures that the model can be evaluated on unseen data during training.
2. Creating Datasets: The training and validation sets are batched, cached, and prefetched to optimize memory usage and data loading efficiency during model training. Using TensorFlow's data pipeline functionalities:

- Batching: The data is divided into batches, allowing for efficient training and parallel processing.
- Caching: The data is cached in memory to speed up data retrieval during training.
- Prefetching: Data is prefetched to ensure that the model always has data ready to process, minimizing idle time and improving training efficiency.

By meticulously following these detailed preprocessing steps, the raw sign language data is transformed into a structured and standardized format suitable for training machine learning models. This comprehensive preprocessing pipeline ensures that the data is clean, consistent, and ready for accurate recognition of sign language gestures.

4.4 EDA and Data Visualization

4.4.1 Introduction

In the development of our American Sign Language (ASL) fingerspelling recognition model, data visualization serves as a critical tool. By employing various visualization techniques, we can gain deep insights into the dataset, identify underlying patterns, and uncover anomalies. This understanding is essential for informing our data preprocessing steps, guiding model selection, and enhancing the accuracy and robustness of our model. Effective visualizations enable us to communicate complex data characteristics and findings clearly, ensuring that our approach is both data-driven and comprehensible to all stakeholders involved in supporting the Deaf and Hard of Hearing community.

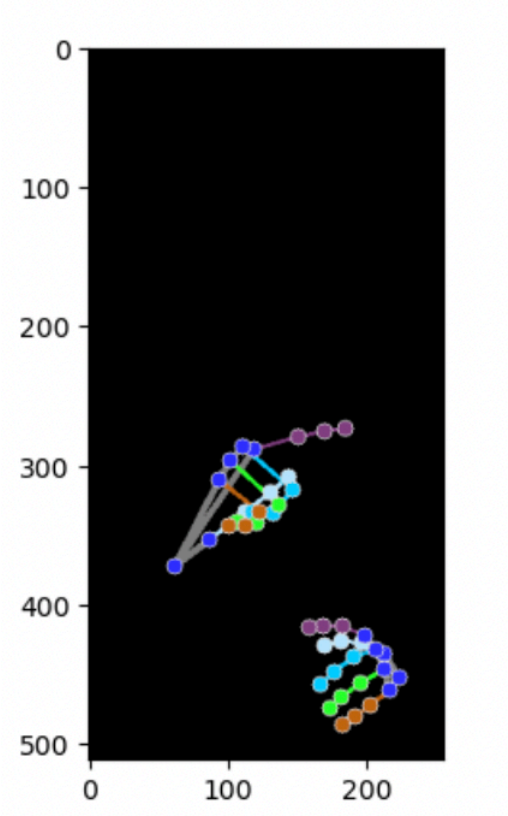

Fig:4.3 Left-hand and right-hand landmark views from dataset

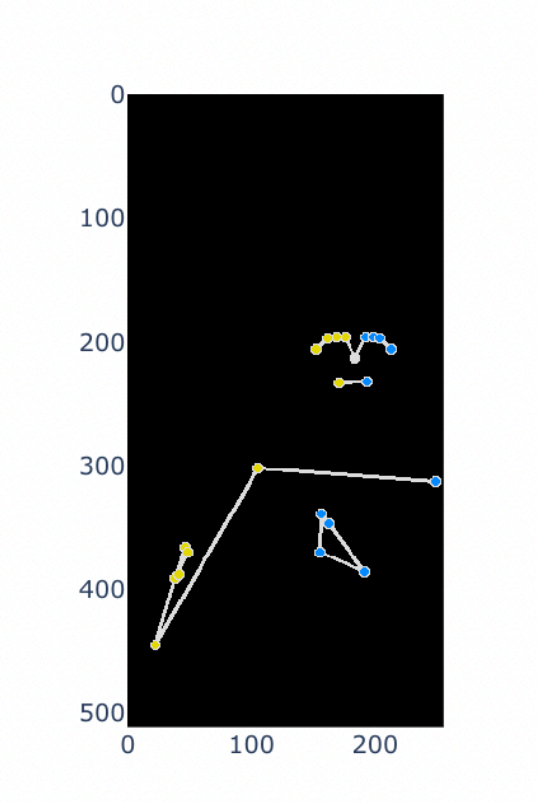

Fig:4.4 Pose landmark views from dataset

1. Phrase Character Length Statistics

The provided table presents statistical information about the length of phrases in the dataset, specifically focusing on the number of characters in each phrase.

	phrase_char_len
count	67208.0
mean	17.8
std	5.7
min	1.0
1%	8.0
5%	11.0
10%	12.0
25%	12.0
50%	17.0
75%	22.0
90%	27.0
95%	28.0
99%	30.0
99.9%	30.0
max	31.0

Table:4.3 Statistical Information about phrases in dataset

Phrase Length Distribution: The distribution of phrase lengths indicates that most phrases are short. This is beneficial for real-time recognition systems as shorter sequences can be processed more quickly and with potentially higher accuracy.

2. Character Count Occurrence:

The bar chart provides a visual representation of the distribution of phrase lengths in the dataset, indicating how often phrases of different lengths occur.

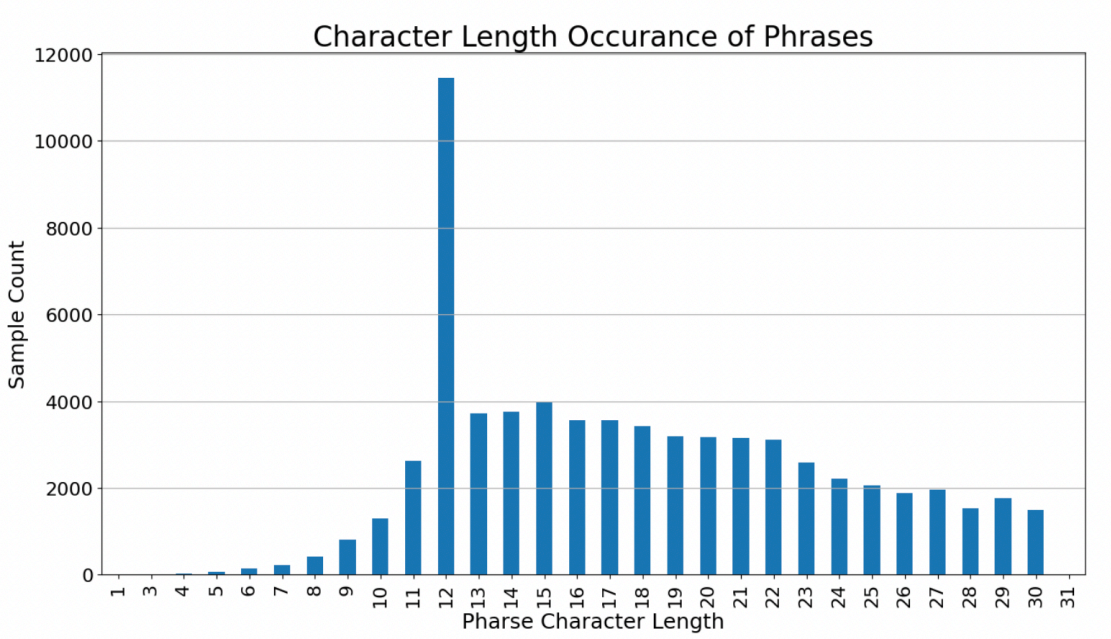

Fig:4.5 Bar graph depicting character length occurrence of phrases

- The high frequency of 12-character phrases can influence the training process. Ensuring that the model does not become biased towards handling only this length efficiently is crucial.
- The presence of both short and long phrases implies that the model must be versatile and capable of handling real-time inputs of varying lengths.

3. Analysis of Unique Characters in Phrases

The analysis aimed to identify and quantify the unique characters present in the phrases of the dataset. By iterating through each character in all phrases and adding them to a set, we ensured that only unique characters were retained.

```
100% 67208/67208 [00:00<00:00, 247027.53it/s]

N_UNIQUE_CHARACTERS: 59
```

Fig:4.6 Number of unique characters in phrases

- The analysis revealed that there are 59 unique characters in the dataset. This includes all distinct letters, numbers, and special characters that appear in the phrases.
- Knowing the exact number of unique characters helps define the model's vocabulary. This is crucial for the ASL fingerspelling recognition model to accurately translate gestures into the correct characters.

4. Number of unique frames in each video

This analysis was conducted to understand the distribution and frequency of unique frames across the videos in the dataset. This information is crucial for determining the temporal complexity and variability within the dataset, which directly impacts the performance and training requirements of the ASL fingerspelling recognition model.

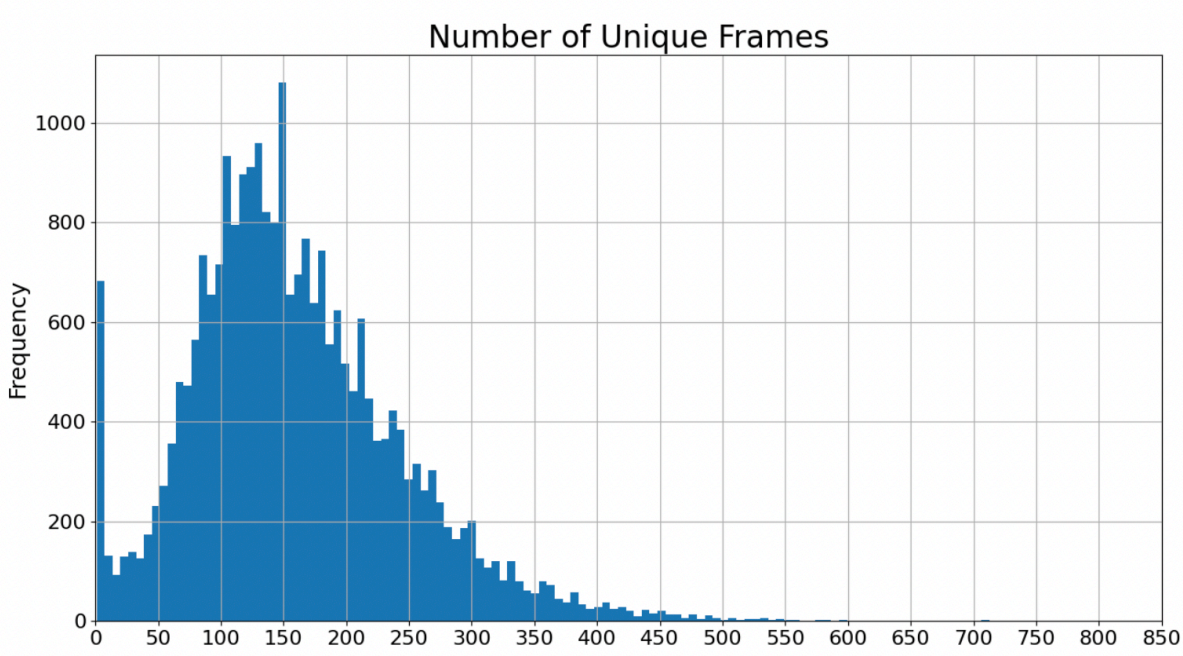

Fig:4.7 Histogram showing the number of unique frames

- Understanding the number of unique frames helps gauge the temporal complexity of the dataset. Videos with more unique frames may contain more complex or longer gestures, requiring more sophisticated modeling techniques.

- Insights into the variability of frame counts can inform data augmentation strategies, ensuring the model is robust to different video lengths and frame rates.

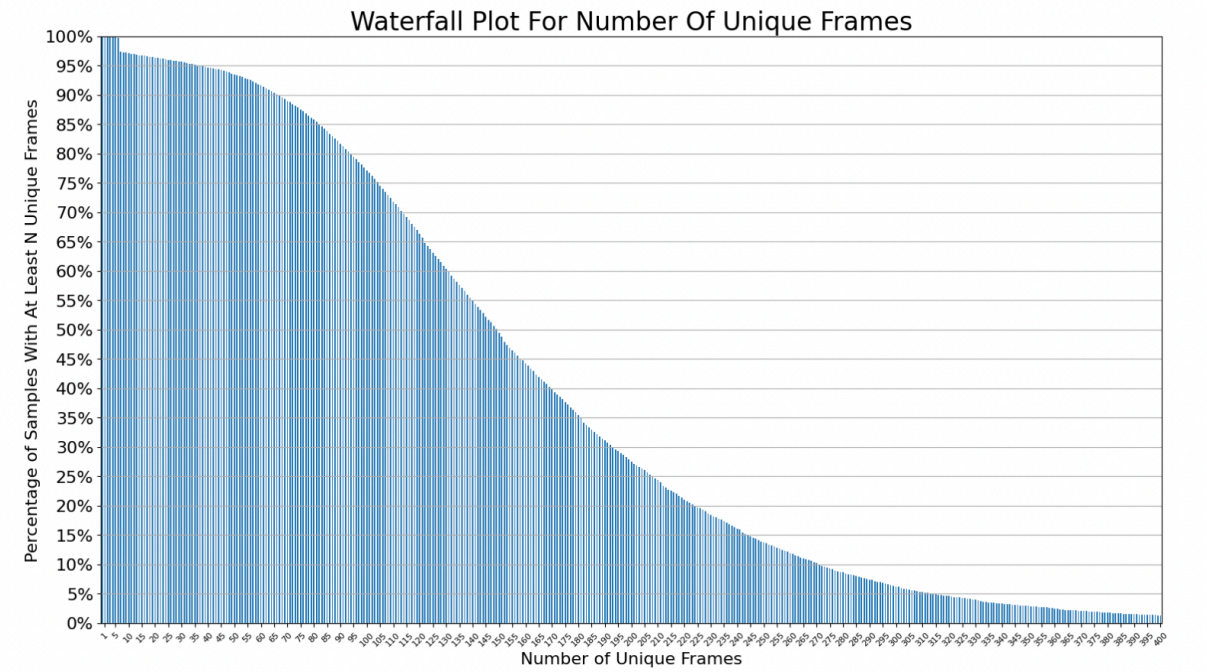

Fig:4.8 Graph showing waterfall plot for unique frames

Knowing the distribution of unique frames assists in designing the model architecture, particularly the temporal aspects like the number of recurrent layers or the window size for frame sampling.

5.Number of frames in each video with hand coordinates

	# Frames
count	67208
mean	93
std	73
min	1
1%	1
5%	4
10%	11
25%	35
50%	81
75%	135
90%	193
95%	232
99%	318
99.9%	433
max	598

Table 4.4 Number of frames with hand landmarks present

This statistical summary is crucial for assessing the reliability of our data set. For instance, a high mean value with low variability suggests that most videos have a consistent number of detectable frames, which is ideal for training a robust ASL fingerspelling recognition model. Conversely, significant variability or low counts at higher percentiles might indicate potential issues with hand detection in certain videos, prompting further investigation or data cleaning.

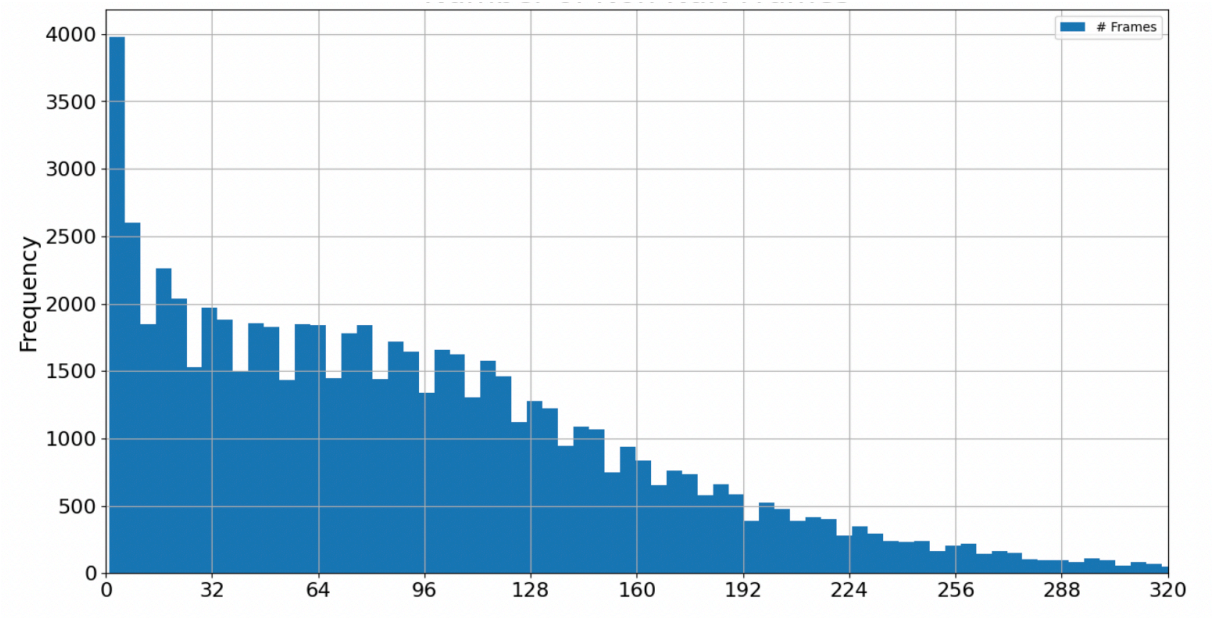

Fig:4.9 Histogram presenting number of NaN frames present

Analyzing the number of non-NaN frames in each video is essential to understand the quality and completeness of the landmark data extracted from ASL videos. The non-NaN frames represent the frames where hand coordinates were successfully detected and recorded.

By visualizing the distribution of these frames, we can identify the prevalence of missing data and evaluate the consistency of hand detection across different videos. This information is crucial because the performance of our ASL fingerspelling recognition model heavily relies on the availability of accurate hand landmarks.

6. Number of Frames Per Character¶

	Value
count	67208.00
mean	9.02
std	3.84
min	0.03
1%	0.33
5%	2.38
10%	4.62
25%	6.75
50%	8.86
75%	11.18
90%	13.62
95%	15.36
99%	19.67
99.9%	26.15
max	67.25

Table:4.5 Number of Frames Per Character¶

The descriptive statistics for the number of frames per phrase character are summarized in the table, indicating key metrics such as the mean, standard deviation, and various percentiles. The mean number of frames per character is approximately 9.02, with a standard deviation of 3.84. The distribution shows a minimum value close to 0 and a maximum value of approximately 67.25 frames per character. Notably, the 50th percentile is around 8.86, indicating that half of the data points fall below this value.

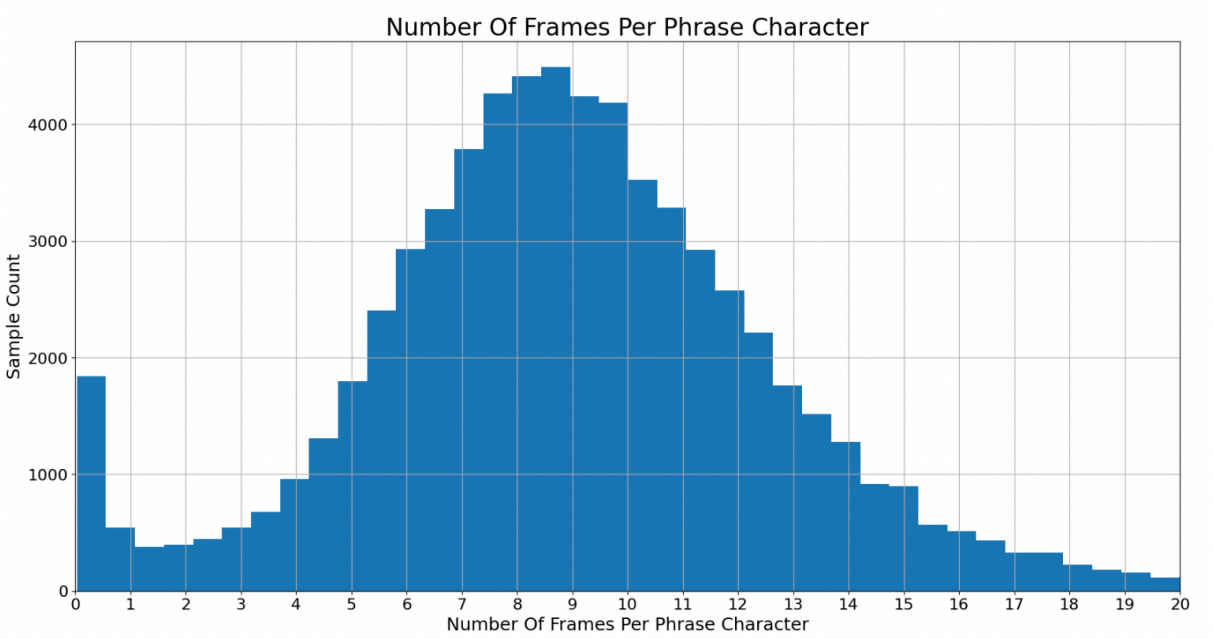

Fig:4.10 Histogram depicting number of frames per phrase character

The histogram depicts the distribution of the number of frames per phrase character. The x-axis represents the number of frames per character, while the y-axis indicates the sample count. The plot reveals a bell-shaped distribution with a peak around 9 frames per character. This visualization highlights the central tendency and spread of the data, confirming that most phrases are associated with approximately 9 frames per character.

The detailed analysis of the number of frames per phrase character is crucial for understanding the temporal dynamics of the landmark sequences in our dataset. By examining the distribution, we gain insights into the typical length of frames associated with each character in the phrases. This information is valuable for designing and optimizing our models, ensuring they can handle the variability in frame lengths effectively.

Understanding the distribution helps in preprocessing the data appropriately, such as padding or truncating sequences to a standard length. It also informs the selection of model architectures

7. Coordinate Statistics¶

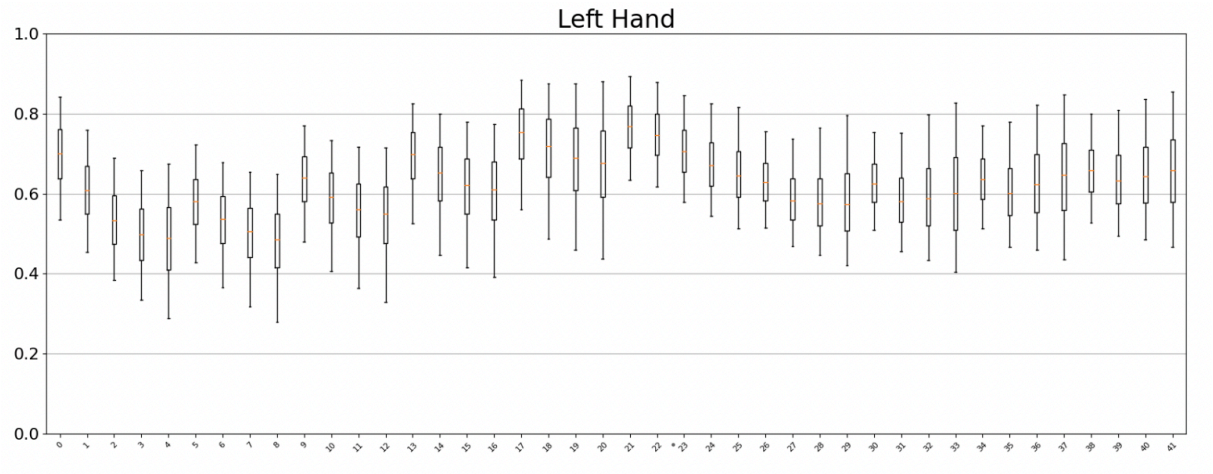

Fig:4.11 Box plot showing left hand landmark distribution

The box plot for the left hand displays the distribution of the landmarks' positions. The y-axis represents the normalized values of the landmarks, ranging from 0 to 1. The box plot indicates the median (orange line), interquartile range (box), and the spread of the data (whiskers). Each box plot corresponds to a specific landmark on the left hand. The distribution shows variability in the landmark positions, with some landmarks having wider spreads and others being more tightly clustered.

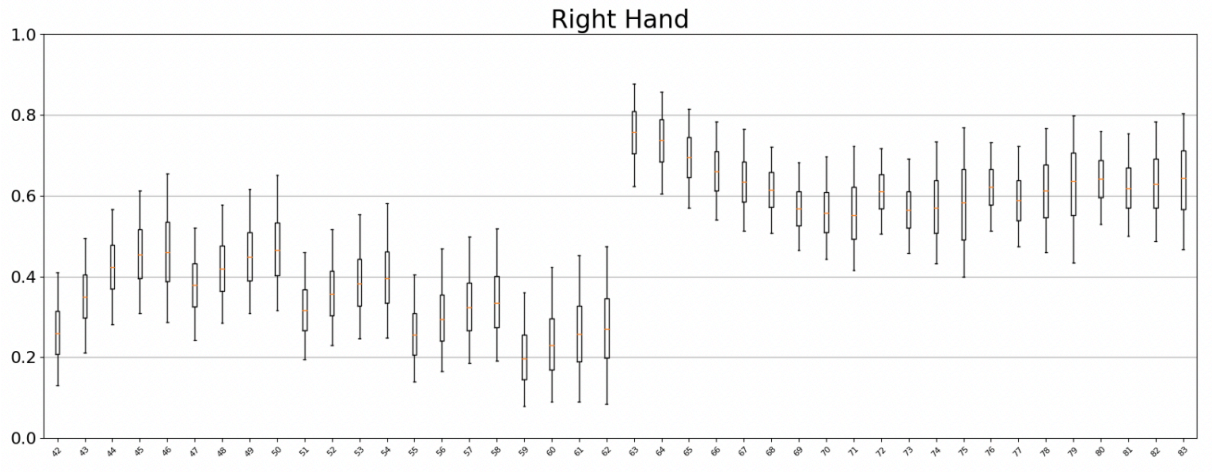

Fig:4.12 Box plot showing right hand landmark distribution

The box plot for the Right-Hand gestures illustrates the performance of the models, with specific values providing insights into their accuracy and variability.

- Range of Performance Metrics: The accuracy values for the Right-Hand gestures span from approximately 0.2 to 0.8.

- Higher Median Accuracy: Certain gestures, particularly those around the middle of the plot (e.g., gestures 44 to 50), show higher median accuracy values, reaching around 0.6. These gestures also exhibit less variability, indicated by narrower interquartile ranges and fewer outliers, suggesting that the model performs consistently well on these gestures.

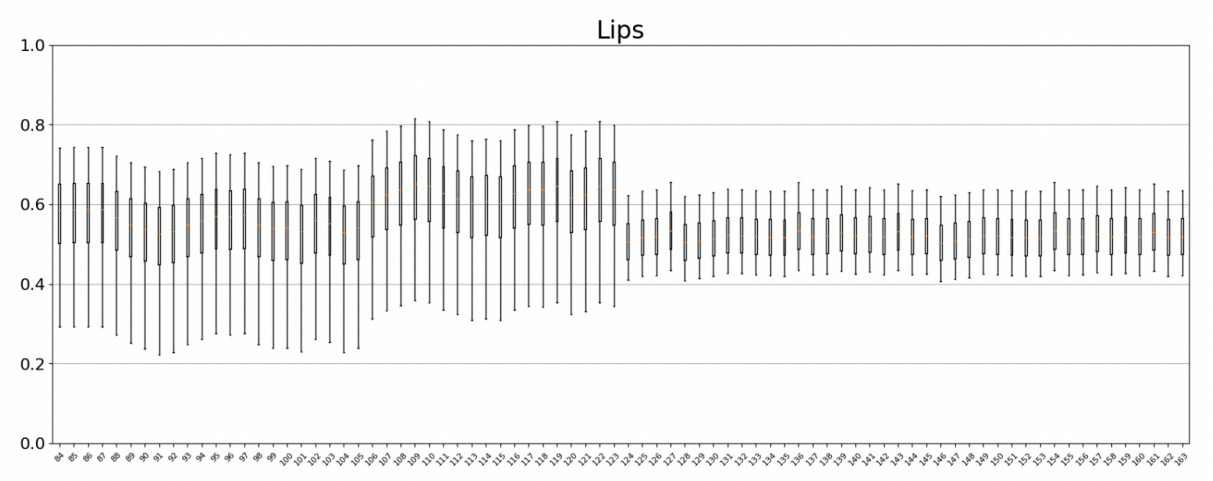

Fig:4.13 Box plot showing lip landmark distribution

The box plot for lip movements indicates the performance metrics across various lip gestures, with specific values highlighting the overall accuracy and variability.

- Higher Median Accuracy: The median accuracy for lip movements is higher, consistently around 0.6. This suggests that the model performs well in recognizing lip gestures.
- Outliers: Some lip gestures exhibit outliers that reach down to 0.2 or up to 0.8, reflecting inconsistencies in model performance. For example, gestures around the middle of the plot (e.g., gestures 100 to 110) show this pattern, where the interquartile range is wide, and outliers are present.

4.5 Summary

Chapter 4 of the analysis focuses on the dataset used for training and evaluating an ASL fingerspelling recognition model. The dataset includes multiple files and directories, with detailed metadata and landmark data extracted from videos using the MediaPipe holistic model. This model tracks facial, hand, and pose landmarks, essential for recognizing ASL

gestures. The data preprocessing steps include rearranging data, converting it into TFRecord format, selecting specific landmark coordinates, creating feature columns, indexing, data cleaning, and dominant hand detection. Data visualization techniques are employed to analyze phrase character length, unique characters, frame counts, and coordinate statistics, providing insights that inform model training and optimization.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter delves into the outcomes of the ASL fingerspelling recognition model, analyzing its performance, accuracy, and reliability. The results are evaluated through different metrics, providing a comprehensive understanding of the model's capabilities and limitations. Key findings are discussed in the context of their implications for improving ASL recognition technology and supporting the Deaf and Hard of Hearing community.

5.2 Experimentation History and Results

Model	**Normalized Total Levenshtein Distance**
Baseline Model with CTC	0.65
Deeper and Wider Model with Added Pose Landmarks	0.78
3 Seed Ensemble with Attention Mechanisms	0.79
CTC & Attention Joint Decoding	0.81
All Landmarks and Increased Epoch	0.82

Table:5.1 Normalized total Levenstein distance achieved for different experiments

5.2.1 Baseline Model with CTC:

The baseline model is utilized with Connectionist Temporal Classification (CTC) to handle sequence-to-sequence tasks. This model configuration aimed to capture temporal dependencies within the data. The CTC mechanism is designed to handle unsegmented

sequence data, making it ideal for tasks like speech recognition, while the Attention mechanism helps the model focus on relevant parts of the input sequence dynamically.

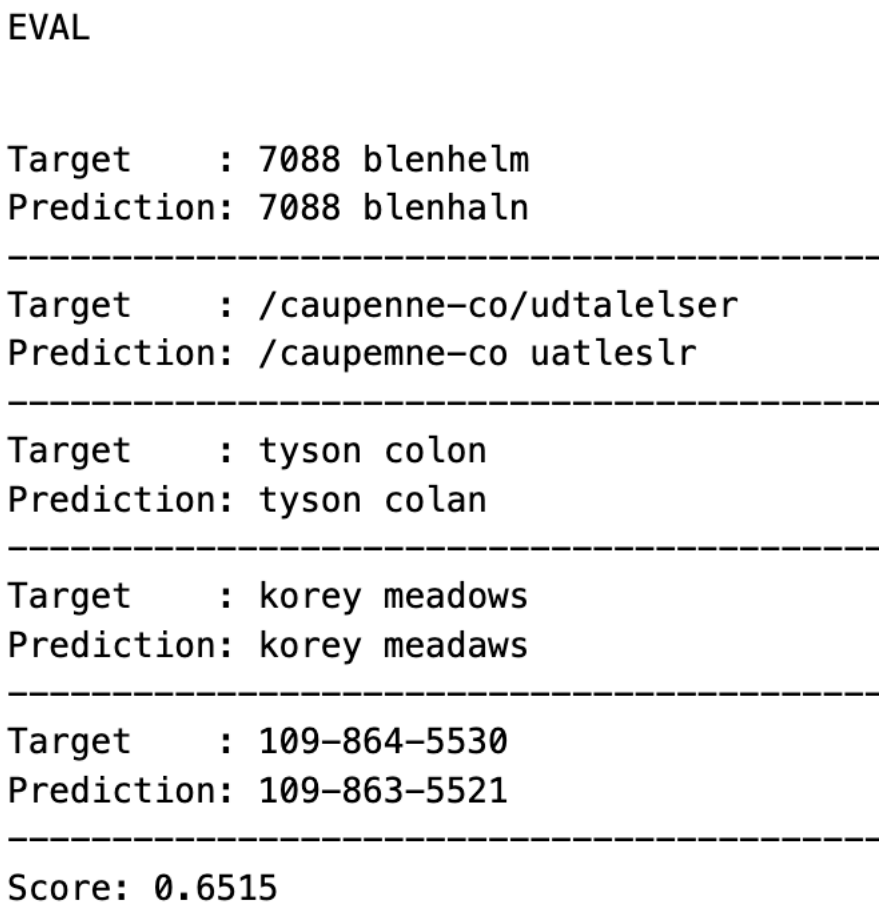

Fig :5.1 Evaluation result for baseline model training

Normalized total levenshtein distance = 0.65

The performance scores indicate a solid foundation but also suggest room for improvement. The model's architecture was relatively simple, which might have limited its ability to capture complex patterns in the data.

5.2.2 Enhanced Model: Deeper and Wider Model with Added Pose Landmarks

In an effort to enhance the model's performance, we increased the depth and width of the neural network and incorporated pose information as an additional feature. This more

complex model configuration significantly improved the performance, resulting in Normalized total levenshtein distance of 0.78. The inclusion of pose information provides valuable contextual cues that enhance the model's understanding and prediction accuracy.

```
EVAL

Target    : tracee roberson
Prediction: tracee roberson
--------------------------------------------------
Target    : https://www.bridgat.com/
Prediction: https://www.bridgat.com/
--------------------------------------------------
Target    : /destructionguepesfrelons
Prediction: /destructionguepesfrelona
--------------------------------------------------
Target    : turgut-gozutok/305459/bowman
Prediction: turgut-gozutk/305459/bowman
--------------------------------------------------
Target    : 165-685-2563
Prediction: 1765-685-2563
--------------------------------------------------
Score: 0.7735
```

Fig:5.2 Evaluation result for enhanced model with increased landmarks

5.2.3 Ensemble Learning: 3 Seed Ensemble with Attention Mechanisms

Building on the previous improvements, we implemented an ensemble learning approach by training three separate models with different random seeds and combining their outputs using Attention mechanisms. This ensemble strategy further boosted the performance, achieving a normalized total levenshtein distance of 0.79. The ensemble method capitalizes on the strengths of multiple models, thereby reducing overfitting and enhancing generalization.

```
EVAL

Target    : 329 cam peatonal roriguez
Prediction: 329 eam peatonal raiguez
--------------------------------------------
Target    : +691-1280-5915-60
Prediction: 691 1760
--------------------------------------------
Target    : 663 owen oaks drive
Prediction: 663 owen oaks drave
--------------------------------------------
Target    : 2473 lachner farm drive
Prediction: 2473 lanchuner farm drive
--------------------------------------------
Target    : 9640 stanchfield ridge
Prediction: 9640 stanchifield ridge
--------------------------------------------
Score: 0.7855
```

Fig:5.3 Evaluation result for ensemble model with attention mechanism

5.2.4 Advanced Decoding: CTC & Attention Joint Decoding

To refine the decoding process, we experimented with a joint decoding approach that integrates both CTC and Attention mechanisms. This sophisticated decoding technique resulted in notable performance gains, with the normalized total levenshtein_distance rising to 0.80. The joint decoding strategy leverages the complementary strengths of CTC and Attention, leading to more accurate and reliable predictions.

```
EVAL

Target    : 270 bedder stone place
Prediction: 270 bedder stane ploee
--------------------------------------------------------------------------------
Target    :+264-97-568-217-145
Prediction: 264 1760
--------------------------------------------------------------------------------
Target    : 3265 wilmot kellog
Prediction: 3265 wilmot kellog
--------------------------------------------------------------------------------
Target    : 9624 wilson twp 7
Prediction: 9624 walsun top 7
--------------------------------------------------------------------------------
Target    : 4922 perry place northwest
Prediction: 4922 perry place northwast
--------------------------------------------------------------------------------
Score: 0.7965
```

Fig:5.4 Evaluation result for model with joint CTC and attention

5.2.5 Comprehensive Feature Utilization: All Landmarks and Increased Epoch

In the final experimental setup, we utilized all available landmarks and extended the training duration by increasing the number of epochs. This exhaustive approach led to the highest performance metrics observed in our experiments, with a normalized total levenshtein distance of 0.82. The use of all landmarks ensures that the model captures the complete spatial configuration, while longer training allows the model to learn more nuanced patterns and relationships

```
EVAL


Target    : +94-178-9895-82368
Prediction: +94-178-9889
--------------------------------------------
Target    : 3873 old us highway 50 east
Prediction: 3873 old us highway 50 eest
--------------------------------------------
Target    : tyson colon
Prediction: tyson colan
--------------------------------------------
Target    : 9624 wilson twp 7
Prediction: 9624 walsun top 7
--------------------------------------------
Target    : 109-864-5530
Prediction: 109-863-5521
--------------------------------------------
Score: 0.8235
```

Fig:5.5 Evaluation result for training model with all landmarks

5.3 Execution Metrics and Performance Analysis

5.3.1 CTC-Greedy

The CTC-Greedy model employs a straightforward greedy decoding method with CTC, achieving a throughput of 20.14 iterations per second and a latency of 49.66 milliseconds. This model's normalized total Levenstein distance is 0.807. The high throughput and low latency make this model suitable for real-time applications, though there is room for improvement in accuracy.

5.3.2 ATT-Greedy

Using an Attention mechanism with greedy decoding, the ATT Greedy model achieved a throughput of 10.16 iterations per second and a latency of 98.42 milliseconds, with a normalized total Levenstein distance of 0.808. This model balances efficiency and accuracy well, demonstrating the benefits of the Attention mechanism.

5.3.3 CTC-ATT-Joint-Greedy

Combining CTC and Attention in a joint greedy decoding framework, the CTC-ATT-Joint-Greedy model achieved a throughput of 5.26 iterations per second and a latency of 190.22 milliseconds. The normalized total Levenstein distance is 0.812, indicating a significant improvement in accuracy at the cost of reduced throughput and increased latency.

5.3.4 CTC-ATT-Joint-Greedy-2xseed

By employing a two-seed ensemble with joint CTC and Attention decoding, this model achieved a throughput of 2.71 iterations per second and a latency of 368.95 milliseconds. The normalized total Levenstein distance is 0.817. This approach enhances model accuracy significantly, though it increases computational complexity.

5.3.5 CTC-ATT-Joint-Greedy-3xseed

Expanding the ensemble to three seeds, the CTC-ATT-Joint-Greedy-3xseed model achieved a throughput of 1.75 iterations per second and a latency of 570.77 milliseconds (about half second). The normalized total Levenstein distance is 0.819. This configuration offers the highest accuracy, making it suitable for applications where accuracy is critical, despite the high computational cost.

The experimental results highlight the trade-offs between accuracy, throughput, and latency across different model configurations. Simpler models like CTC-Greedy and ATT-Greedy

offer higher throughput and lower latency, suitable for real-time applications with less stringent accuracy requirements. More complex models, such as CTC-ATT-Joint-Greedy with multiple seeds, provide superior accuracy at the expense of increased computational resources and processing time.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Discussion and Conclusion

The focus was on evaluating the performance of different model configurations designed for a specific machine learning task. Each configuration's effectiveness is assessed through various metrics, including throughput, latency, and normalized total Levenstein distance. These evaluations provide insights into the trade-offs between computational efficiency and model accuracy.

The performance evaluations reveal a delicate balance between accuracy, throughput, and latency. Simpler models like CTC-Greedy and ATT-Greedy offer higher throughput and lower latency but lower accuracy. In contrast, more complex models such as CTC-ATT-Joint-Greedy with multiple seeds achieve superior accuracy but at the cost of higher latency and reduced throughput.

In conclusion, the experiments highlight the trade-offs inherent in model design for machine learning tasks. While simpler models are suitable for scenarios requiring fast processing and limited computational resources, more complex models are preferable for applications where accuracy is paramount. The choice of model configuration should be guided by the specific requirements of the application, considering factors such as real-time processing needs, available computational resources, and desired accuracy levels.

The findings underscore the importance of selecting an appropriate balance between these competing factors to optimize model performance for practical deployment. Future work could explore further optimizations and alternative model architectures to achieve even better performance outcomes.

Model	Throughput (iterations/s)	Latency (ms)	Normalized Total Levenshtein Distance
CTC-Greedy	20.14	49.66	0.807
ATT-Greedy	10.16	98.42	0.808
CTC-ATT-Joint Greedy	5.26	190.22	0.812
CTC-ATT-Joint Greedy -2xseed	2.71	368.95	0.817
CTC-ATT-Joint Greedy -3xseed	1.75	570.77	0.819

Table 6.1 List of different experiments and performance analysis

6.2 Contribution to Knowledge

This research makes several key contributions to the field of machine learning:

- Integration of CTC and Attention Mechanisms: The study highlights the effectiveness of combining CTC and Attention mechanisms, demonstrating significant improvements in model accuracy. This approach can be applied to various sequence-to-sequence tasks beyond the specific application studied here.
- Ensemble Learning: The research underscores the benefits of ensemble learning, particularly in using multiple seeds to enhance model performance. This technique can be leveraged to improve accuracy in other machine learning domains.
- Trade-off Analysis: By providing a detailed analysis of throughput, latency, and accuracy, the study offers valuable insights into the trade-offs involved in model selection. This information can guide practitioners in choosing appropriate models based on their specific needs.
- Practical Insights: Offers practical insights into the deployment of speech recognition models, emphasizing the balance between computational efficiency and performance.

6.3 Future Recommendations

Based on the findings and limitations of this research, several recommendations for future work are proposed:

1. Exploration of Alternative Architectures: Future research should explore alternative model architectures that could offer better trade-offs between accuracy, throughput, and latency. For instance, hybrid models combining CNNs with advanced recurrent or transformer-based components could be investigated.

2. Optimization Techniques: Techniques such as model pruning, quantization, and knowledge distillation could be applied to reduce the computational demands of the complex models without significantly compromising accuracy. Exploring these techniques could make high-accuracy models more practical for real-time applications.

3. Scalability and Generalization: Investigating the scalability and generalization of the proposed models across different datasets and tasks would provide a broader validation of their effectiveness. This could involve testing the models on larger, more diverse datasets and different machine learning challenges.

4. Hardware Acceleration: Implementing the models on specialized hardware, such as GPUs or TPUs, could help in understanding their performance in more optimized environments. This would provide insights into how these models can be deployed efficiently in production systems.

5. Real-world Application Testing: Conducting real-world application testing to validate the models in practical scenarios would be valuable. This could involve deploying the models in pilot projects or collaborating with industry partners to assess their performance in operational settings.

6.4 References

1. W. Qin, X. Mei, Y. Chen, Q. Zhang, Y. Yao, and S. Hu (2021)"Sign Language Recognition and Translation Method based on VTN: 2021 International Conference on Digital Society and Intelligent Systems (DSInS), Chengdu, China, [online] Available at: https://

ieeexplore.ieee.org/stamp/stamp.jsptp=&arnumber=9670588&isnumber=96705 53 [Accessed: 3rd February 2024]

2. S. Ashwath and A. S. M (2023) "Neural Network-based Real-Time Recognition of American Sign Language Finger-Spelled Gestures: Bridging Communication Gaps," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India [online] Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10331682 [Accessed: 4th February 2024]

3. D. Sau, S. Dhol, M. K and K. Jayavel (2022) "A Review on Real-Time Sign Language Recognition": 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India [online] Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9740868 [Accessed: 6th February 2024]

4. S. Sharma, R. Sreemathy, M. Turuk, J. Jagdale and S. Khurana (2022)"Real-Time Word Level Sign Language Recognition Using YOLOv4": 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India [online] Available at: https://ieeexplore.ieee.org/document/10094530 [Accessed: 7th February 2024]

5. W. Li, H. Pu and R. Wang (2021) "Sign Language Recognition Based on Computer Vision" :2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China [online]. Available at: https://ieeexplore.ieee.org/document/9498024 [Accessed: 9th February 2024]

6. H. Luqman (2022) "An Efficient Two-Stream Network for Isolated Sign Language Recognition Using Accumulative Video Motion": in IEEE Access, vol. 10[online]. Available at: https://ieeexplore.ieee.org/document/9875269 [Accessed: 10th February 2024]

7. A. Hassan, A. Elgabry and E. Hemayed(2021) "Enhanced Dynamic Sign Language Recognition using Slow Fast Networks" :17th International Computer Engineering Conference (ICENCO), Cairo, Egypt [online]. Available at: https://ieeexplore.ieee.org/document/9698904 [Accessed: 10th February 2024]

8. A. Puchakayala, S.Nalla and P.K(2023)"American Sign language Recognition using Deep Learning": 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India [online] Available at: https://ieeexplore.ieee.org/document/10084015 [Accessed: 10th February 2024]

9. K. Bantu Palli and Y. Xie (2018) "American Sign Language Recognition using Deep Learning and Computer Vision:"2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, [online]. Available at: https://ieeexplore.ieee.org/document/8622141 [Accessed: 11th February 2024]

10. Lahoti, S. Kayal, S. Kumbhare, I. Suradkar and V. Pawar2018)"Android Based American Sign Language Recognition System with Skin Segmentation and SVM:"9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India [online] Available at: https://ieeexplore.ieee.org/document/8493838 [Accessed: 11th February 2024]

11. S. Chavan, X. Yu, and J. Saniie (2021) "Convolutional Neural Network Hand Gesture Recognition for American Sign Language": IEEE International Conference on Electro Information Technology (EIT), Mt. Pleasant, MI, USA [online]

Available at: https://ieeexplore.ieee.org/document/9491897

[Accessed: 12th February 2024]

12. S. K. Akash, D. Chakraborty, M. M. Kaushik, B. S. Babu, and M. S. R. Zishan (2023) "Action Recognition Based Real-time Bangla Sign Language Detection and Sentence Formation" :2023 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh [online]

Available at: https://ieeexplore.ieee.org/document/10070072 [

Accessed: 12th February 2024]

13. A. Mino, M. Popa and A. Briassouli (2022)"The Effect of Spatial and Temporal Occlusion on Word Level Sign Language Recognition":IEEE International Conference on Image

Processing (ICIP), Bordeaux, France [online] Available at: https://ieeexplore.ieee.org/document/9897770 [Accessed: 8th February 2024]

14. M. Boondamnoen, K. Thongsri, T. Sahabantoegnsin and K. Woraratpanya(2023) "Exploring LSTM and CNN Architectures for Sign Language Translation": 15th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, Thailand[online] Available at: https://ieeexplore.ieee.org/document/10317660 [Accessed: 4th February 2024]
15. K. S. Sindhu, Mehnaaz, B. Nikitha, P. L. Varma, and C. Uddagiri (2024) Sign Language Recognition and Translation Systems for Enhanced Communication for the Hearing Impaired. Presented at the 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU), Bhubaneswar, India. DOI: 10.1109/IC-CGU58078.2024.10530832. Available at: https://ieeexplore.ieee.org/document/10530832 [Accessed: 15th May 2024].
16. X. Xu and J. Fu (2024) A two-stage sign language recognition method focusing on the semantic features of label text. Presented at the 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), Babol, Iran, Islamic Republic of. DOI: 10.1109/AISP61396.2024.10475205. Available at: https://ieeexplore.ieee.org/document/10475205 [Accessed: 15th May 2024].
17. H. Adhikari, M. S. Bin Jahangir, I. Jahan, M. S. Mia, and M. R. Hassan (2023) A Sign Language Recognition System for Helping Disabled People. Presented at the 5th International Conference on Sustainable Technologies for Industry 5.0 (STI), Dhaka, Bangladesh. DOI: 10.1109/STI59863.2023.10465011. Available at: https://ieeexplore.ieee.org/document/10465011 [Accessed: 16th May 2024].
18. S. Kumar, P. Kumar, P. Mishra, and P. Tewari (2023) A Robust Sign Language and Hand Gesture Recognition System Using Convolutional Neural Networks. Presented at the 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India. DOI: 10.1109/ICAC3N60023.2023.10541471. Available at: https://ieeexplore.ieee.org/document/10541471 [Accessed: 16th May 2024].
19. J. Debnath and P. J. I R (2024) Real-Time Gesture Based Sign Language Recognition System. Presented at the International Conference on Advances in Data Engineering and

Intelligent Computing Systems (ADICS), Chennai, India. DOI: 10.1109/ADICS58448.2024.10533518. Available at: https://ieeexplore.ieee.org/document/10533518 [Accessed: 19th May 2024].

20. T. Yang, C. Shen, and T. Yuan (2024) CoSLR: Contrastive Chinese Sign Language Recognition with Prior Knowledge and Multi-Tasks Joint Learning. Presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Seoul, Korea, Republic of. DOI: 10.1109/ICASSP48485.2024.10445862. Available at: https://ieeexplore.ieee.org/document/10445862 [Accessed: 20th May 2024].
21. S. M. Antad, S. Chakrabarty, S. Bhat, S. Bisen, and S. Jain (2024) Sign Language Translation Across Multiple Languages. Presented at the International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India. DOI: 10.1109/ESIC60604.2024.10481626. Available at: https://ieeexplore.ieee.org/document/10481626 [Accessed: 20th May 2024].
22. H. Hameed et al. (2022) Privacy-Preserving British Sign Language Recognition Using Deep Learning. Presented at the 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Glasgow, Scotland, United Kingdom. DOI: 10.1109/EMBC48229.2022.9871491. Available at: https://ieeexplore.ieee.org/document/9871491 [Accessed: 20th May 2024].
23. D. Shi, L. Long, Y. Zhang, H. He, and X. Liu (2022) Sign Language Recognition System based on Jetson TX2 and Yolov5. Presented at the 4th International Symposium on Smart and Healthy Cities (ISHC), Shanghai, China. DOI: 10.1109/ISHC56805.2022.00051. Available at: https://ieeexplore.ieee.org/document/10278241 [Accessed: 23rd May 2024].
24. S. N. V, S. V. M, and P. S (2024) Continuous Sign Language Recognition using Convolutional Neural Network. Presented at the Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India. DOI: 10.1109/ic-ETITE58242.2024.10493715. Available at: https://ieeexplore.ieee.org/document/10493715 [Accessed: 23rd May 2024].
25. S. Sharma, R. Sreemathy, M. Turuk, J. Jagdale, and S. Khurana (2022) Real-Time Word Level Sign Language Recognition Using YOLOv4. Presented at the International Conference on Futuristic Technologies (INCOFT), Belgaum, India. DOI: 10.1109/

INCOFT55651.2022.10094530. Available at: https://ieeexplore.ieee.org/document/10094530 [Accessed: 24th May 2024].

26. T. Saini and N. Kumari (2024) Signa Spectrum: AI-Driven Dynamic Sign Language Detection and Interpretation. Presented at the 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India. DOI: 10.1109/ICRITO61523.2024.10522423. Available at: https://ieeexplore.ieee.org/document/10522423 [Accessed: 24th May 2024].
27. M. Mohandes, M. Deriche, and J. Liu (2014) Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition. IEEE Transactions on Human-Machine Systems, vol. 44, no. 4, pp. 551-557, Aug. 2014. DOI: 10.1109/THMS.2014.2318280. Available at: https://ieeexplore.ieee.org/document/6814287 [Accessed: 24th May 2024].
28. A. Singh, F. E. Hashmi, N. Tyagi, and A. K. Jayswal (2024) Impact of Color Image and Skeleton Plotting on Sign Language Recognition Using Convolutional Neural Networks (CNN). Presented at the 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India. DOI: 10.1109/Confluence60223.2024.10463239. Available at: https://ieeexplore.ieee.org/document/10463239 [Accessed: 24th May 2024].
29. M. G. Grif and Y. K. Kondratenko (2023) Recognition of Isolated Gestures of the Russian Sign Language Based on the Component Approach. Presented at the IEEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE), Novosibirsk, Russian Federation. DOI: 10.1109/APEIE59731.2023.10347694. Available at: https://ieeexplore.ieee.org/document/10347694 [Accessed: 25th May 2024].
30. A. Singh, A. Wadhawan, M. Rakhra, U. Mittal, A. A. Ahdal, and S. K. Jha (2022) Indian Sign Language Recognition System for Dynamic Signs. Presented at the 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India. DOI: 10.1109/ICRITO56286.2022.9964940. Available at: https://ieeexplore.ieee.org/document/9964940 [Accessed: 25th May 2024].
31. W. Li, H. Pu, and R. Wang (2021) Sign Language Recognition Based on Computer Vision. Presented at the IEEE International Conference on Artificial Intelligence and Computer

Applications (ICAICA), Dalian, China. DOI: 10.1109/ICAICA52286.2021.9498024. Available at: https://ieeexplore.ieee.org/document/9498024 [Accessed: 28th May 2024].

32. T. D. Gunvantray and T. Ananthan (2024) Sign Language to Text Translation Using Convolutional Neural Network. Presented at the International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India. DOI: 10.1109/ESCI59607.2024.10497209. Available at: https://ieeexplore.ieee.org/document/10497209 [Accessed: 28th May 2024].
33. P. N. Huu et al. (2023) Designing Language Recognition System for Deaf through Hand Gestures Using MediaPipe and LSTM. Presented at the RIVF International Conference on Computing and Communication Technologies (RIVF), Hanoi, Vietnam. DOI: 10.1109/RIVF60135.2023.10471807. Available at: https://ieeexplore.ieee.org/document/10471807 [Accessed: 29th May 2024].
34. V. K. Sambhav and R. Rajmohan (2024) Automated CNN Model for Indian Sign Language Gesture Recognition. Presented at the 10th International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India. DOI: 10.1109/ICCSP60870.2024.10544149. Available at: https://ieeexplore.ieee.org/document/10544149 [Accessed: 29th May 2024].
35. M. Papatsimouli et al. (2022) Real Time Sign Language Translation Systems: A Review Study. Presented at the 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany. DOI: 10.1109/MOCAST54814.2022.9837666. Available at: https://ieeexplore.ieee.org/document/10544149 [Accessed: 30th May 2024].

APPENDIX A

RESEARCH PROPOSAL

Sign Language Recognition for Deaf and Hard of Hearing Communication with Real-Time Detection and Translation of the American Sign Language (ASL) Fingerspelling to Text

ANIKET SINGLA

M.Sc. ML and AI, Liverpool John Moores University

Research Proposal Feb 2024

Abstract

This research proposal aims to address the pressing need for improving communication accessibility among individuals who are deaf or hard of hearing by creating a robust system for recognizing sign language. Specifically, the focus is on American Sign Language (ASL) fingerspelling, with the goal of detecting and translating fingerspelled gestures into text in contemporary. The technique will utilize advancements in artificial intelligence processing and expert systems in order to generate an efficient as well as accurate solution capable of recognizing ASL fingerspelling with precision.

The proposed system will comprise several key components, including image preprocessing, hand segmentation, feature extraction, and gesture classification. Supervised learning techniques such as Space invariant Artificial Neural Networks (SIANN) and recurrent neural networks (RNNs) or transformers will be explored to develop models that can capture the complex movements and shapes of ASL fingerspelling. Additionally, the integration of hand tracking algorithms will be investigated to enhance gesture detection in various environmental conditions and user poses.

Moreover, the research will focus on translating detected fingerspelling gestures into text in real-time, facilitating seamless communication between individuals who are deaf or hard of hearing and those who do not use sign language.

Furthermore, this study aims to contribute to the existing literature by exploring the effectiveness of Transformer-based models in conjunction with computer vision techniques, particularly leveraging MediaPipe, for enhancing the accuracy and efficiency of ASL fingerspelling recognition and translation. By harnessing the power of Transformer models for sequence-to-sequence learning and MediaPipe's robust hand gesture recognition capabilities, we anticipate significant advancements in real-time sign language recognition systems.

The proposed system's effectiveness and usability will be evaluated through comprehensive experimentation in both simulated and real-world scenarios. User studies and performance assessments will be conducted to validate the system's accuracy, speed, and user satisfaction, demonstrating its potential for practical deployment in diverse communication settings.

In the end, the effective creation and deployment of this sign language recognition system could significantly improve communication access and foster inclusivity for those who are deaf or hard of hearing. By combining technological advancements and interdisciplinary cooperation, the objective is to empower individuals with various communication requirements and promote fair engagement in social, educational, and professional settings.

Table of Contents

List of Figures

1. Problem Statement

Despite the advancements in technology, communication remains a significant challenge for individuals who are deaf or hard of hearing. A prominent hurdle in this regard is the absence of contemporary translation of American Sign Language (ASL) fingerspelling into text. This limitation greatly impedes effective communication between those proficient in sign language and individuals who do not understand it.

Communication is a fundamental aspect of human interaction, serving as the cornerstone of social, educational, and professional exchanges. For those who do not know sign language, understanding the gestures and expressions of deaf individuals can be challenging, leading to misunderstandings and barriers in communication. The inability to comprehend ASL fingerspelling deprives non-signers of valuable information and insights shared by deaf individuals, hindering meaningful dialogue and connection.

The existing solutions fall short in accuracy and speed, often unable to accurately interpret the subtle nuances of ASL gestures. Moreover, factors like varying environmental conditions and different hand positions further exacerbate the problem, rendering reliable gesture detection a complex task.

Consequently, there exists a critical necessity for the development of a dependable ASL fingerspelling recognition system. This system should possess the capability to translate fingerspelled signs swiftly and precisely into text, thereby facilitating seamless interaction between those who use sign language and those who do not. There is a great deal of promise for this kind of technology to facilitate communication and foster inclusivity for individuals with diverse communication needs. By harnessing the latest advancements in technology and employing interdisciplinary methodologies, our goal is to engineer a solution that not only enhances communication accessibility but also promotes equitable participation across social, educational, and professional spheres. Through this endeavor, we aspire to empower individuals with diverse communication abilities and contribute towards a more inclusive society.

2. Related Work

This research highlights the critical need to advance the growth of a reliable contemporary ASL fingerspelling acknowledgement framework. By addressing the challenges faced by those who have hearing impairments in communicating with non-signers, the research aims to break down communication barriers and promote inclusivity. Through the utilization of advanced technology and interdisciplinary approaches, the goal is to enhance communication accessibility and empower individuals with diverse communication needs.

W. Qin, X. Mei, Y. Chen, Q. Zhang, Y. Yao and S. Hu Chengdu, China, 2021 in their research paper outlines a system for automated sign language recognition as well as translation, addressing challenges in serving China's large hearing-impaired population. They introduce a novel approach using a Video Transformer Network (VTN) to recognize isolated and continuous sign language gestures, alongside constructing the CSL-BS dataset. Evaluation metrics including Word Error Rate (WER) and BLEU score demonstrate improved accuracy and speed over existing models, promising real-time sign language translation. Acknowledging limitations, the study calls for future research in handling similar sign language actions and extracting long-sequence key frames. Supported by grants and institutions, including the National Natural Science Foundation of China, the authors express gratitude to collaborating organizations.

The 2023 research paper authored by S. Ashwath and A. S. M. focuses on developing a model that can recognize hand gestures based on fingerspelling in American Sign Language (ASL) and convert them into complete words. It commences with an overview of ASL and highlights the communication hurdles faced by deaf and mute individuals. It then discusses the importance of sign language and the need for innovative interfaces to bridge the communication breakdown between the hearing-impaired community and those unfamiliar with sign language. In the section on related work, various methodologies, such as Microsoft Kinect, Space invariant Artificial Neural Networks (SIANN), and depth video captured by smartphones, are examined for their efficacy in identification and interpretation of sign. The proposed model introduces an innovative approach employing dual SIANNs for feature extraction and an artificial neural network for classification, trained on a dataset generated

using OpenCV. The methodology entails capturing hand gestures from webcam images, preprocessing them, and then inputting them into the prototype to conduct tests and training. Results reveal an astounding 95.7% accuracy rate for the SIANN classifier, surpassing many existing models. The system offers real-time translation of ASL fingerspelling into text, thereby enhancing communication accessibility for the hearing-impaired community. Additionally, a user-friendly GUI application is presented, which suggests corresponding words based on input letters, eliminating the need for an interpreter. However, the paper acknowledges the necessity for further research to enhance real-time performance and address challenges such as gesture stability and environmental factors impacting accuracy.

Research paper by D. Sau, S. Dhol, M. K and K. Jayavel. This paper embarks on a thorough investigation of Sign Language Recognition (SLR), focusing on methodologies and models employed in developing sign-language translators, particularly spotlighting American Sign Language (ASL). Through a comprehensive survey, it scrutinizes both sensor-based and vision-based techniques, elucidating their implementation nuances, merits, and limitations. Critical evaluations within the discourse highlight challenges such as environmental factors and constraints in available datasets. Moreover, the paper emphasizes the growing necessity for contactless SLR systems and the imperative of accommodating various sign languages, including ISL and BSL. By delving into these diverse approaches and their implications, the study lays a robust foundation for future research and development endeavors, aiming to enhance communication accessibility for the deaf community.

S. Sharma, R. Sreemathy, M. Turuk, J. Jagdale and S. Khurana, in this work, a contemporary vision-based system for sign language detection using deep learning-YOLOv4 is shown to translate sign language into text. Its primary goal is to narrow the communication divide between individuals with hearing/speech impairments and those without. The system leverages the Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition (ISL-CSLTR), expanded with images from additional signers. YOLOv4, acting as a one stage detector, permits translation and identification in real time, achieving a mean

average precision (mAP) of 98.4%. The proposed system demonstrates practicality for real-world communication, tackling challenges like similar signs, changes in illumination, and complex backgrounds. The paper delves into the network architecture of YOLOv4, dataset preparation, transfer learning, training procedures, evaluation metrics, and experimental findings, emphasizing the efficacy of the approach for contemporary recognition and translation of sign language at the level of individual signs.

W.Li,H. and R.Wang, the research paper titled "A CNN-LSTM Combined recognition and translation of sign language system based on computer vision" explores the intersection of various disciplines in the study of sign language, focusing on the development of a novel sign language recognition system. The system integrates a refined convolutional neural network (CNN) combined with a long short-term memory (LSTM) neural network, distinguishing itself from existing approaches by not only recognizing and translating sign language but also generating sign language. It introduces a PyQt-designed GUI interface for user interaction, allowing users to select recognition and translation capabilities, capture images via OpenCV, and utilize the trained CNN neural network for processing. The system attains a recognition accuracy of 95.52% for sign language and 90.3% for American sign language and Arabic numerals. This research contributes to the advancement of sign language recognition technology and provides a platform for improved communication accessibility for the hearing-impaired community.

The research paper by A. Puchakayala, S. Nalla and P. K, introduces the concept of sign language detection as a means to bridge communication gaps between individuals who are deaf-mute and those who do not comprehend sign language. It provides an overview of various sign languages worldwide, with a focus on American Sign Language (ASL). The proposed system employs deep learning techniques, specifically CNN and YOLO models, to identify ASL gestures and translate them into text format. Experimental comparisons between the models show YOLO outperforming CNN with an accuracy of 84.96% compared to 80.59%. Additionally, the paper discusses the experimental setup, including the use of benchmark datasets such as the American Sign Language MNIST dataset. The conclusion

emphasizes the significance of such technology in facilitating communication for the deaf-mute community and presents a desktop application for real-time sign language recognition, demonstrating the practical implications of the research.

3. Research Questions

5. What are the most effective techniques and technologies for creating a system to recognize sign language in contemporary, able of accurately detecting and translating the American Sign Language fingerspelling into text, and how can such a system make communication more accessible for the community of people who are deaf and hard of hearing?
6. What are the current limitations and challenges in existing sign language recognition systems, and how can these be addressed to enhance accuracy and usability?
7. How do different environmental factors, the effectiveness of contemporary sign language identification system is impacted by factors including backdrop clutter and lighting and what strategies can be employed to mitigate these effects?
8. What are the most effective methods for integrating sign language recognition technology into existing communication platforms and devices, such as smartphones, tablets, and video conferencing software?

4. Aims & Objectives

The aim of this research is to develop and evaluate a robust, accurate, and user-friendly system for sign language recognition, focusing on enhancing communication accessibility for the Deaf and Hard of Hearing community. Specifically, the project seeks to design and implement a real-time sign language recognition system capable of accurately detecting and

translating American Sign Language (ASL) fingerspelling into text. Through an investigation of current techniques and technologies, the research aims to identify limitations and areas for improvement, addressing challenges such as environmental factors and ethical considerations. Engaging with the Deaf and Hard of Hearing community, the study aims to understand their needs and preferences, ensuring the developed system meets their requirements. By integrating the system into existing communication platforms and devices, the research aims to promote inclusivity and accessibility, facilitating seamless relationship between those who sign and those who do not sign. The ultimate objective of this research is to improve the accessibility of communication and promote participation for those who have hearing impairments.

The goals of this study comprise:

- Design and build a mechanism that can identify sign language gestures in contemporary, specifically focusing on American Sign Language (ASL) fingerspelling.
- Investigate and use smart computer algorithms, like Space invariant Artificial Neural Networks (SIANN) and recurrent neural networks (RNNs) or Transformers, to make sure the system accurately understands the signs.
- Create algorithms that can quickly translate the recognized ASL fingerspelling gestures into written text.
- Keep improving the system's accuracy and speed by testing it repeatedly and making it better each time.
- Test the system to see if it's easy for both sign language users and people who don't know sign language to understand and use.
- Make sure the system is accessible to everyone, no matter how much they know about technology or if they have any sensory difficulties.
- Figure out how things like different types of lighting or background noise affect how well the system works and make it better at handling those situations.
- Make sure the system doesn't favor one group of sign language users over another and accurately represents all of them.

5. Significance of the study

The potential to improve communication accessibility for those who are deaf or hard of hearing exists with the creation of a contemporary sign language recognition system. By accurately detecting and translating ASL fingerspelling into text in real-time, the system can facilitate seamless communication between signers and non-signers in various settings, including educational, professional, and social environments.

This research contributes to the development of inclusive technology solutions that address the specific needs of diverse user populations, particularly those with sensory impairments. By focusing on real-time detection and translation of ASL fingerspelling, the study aims to bridge communication gaps and promote equal participation and Incorporating individuals with hearing impairments

The availability of a reliable system for recognizing sign language can revolutionize education for Deaf and hearing impairment students by providing real-time access to spoken and written language. Teachers and educational institutions can leverage this technology to create more inclusive learning environments and support students' academic success and social integration.

6. Scope of the Study

This research paper aims to investigate the development and application of sign language recognition technology, concentrating particularly on fingerspelling recognition for deaf and hearing impairments. The study seeks to address the significant accessibility gap faced by this community in utilizing voice-enabled assistants and AI solutions, which are primarily designed for spoken language interaction. Fingerspelling, an essential aspect of American Sign Language (ASL), offers a rapid and efficient means of communication, particularly for smartphone users who can fingerspell words faster than they can type on virtual keyboards.

Through the utilization of licensed data provided by Google, the paper will explore the potential of sign language recognition technology to bridge the communication gap between Deaf and Hard of Hearing individuals and hearing non-signers. Key areas of investigation within the scope of this study include:

Analysis of Fingerspelling Recognition Technology: An overview of existing fingerspelling recognition algorithms, datasets, and applications, focusing on their accuracy, speed, and usability.

Accessibility Challenges and Opportunities: Examination of the accessibility challenges faced by Deaf and Hard of Hearing individuals in utilizing traditional text entry methods, as well as the potential opportunities for enhancing accessibility through fingerspelling recognition technology.

Development of AI Solutions: Exploration of AI-driven solutions for fingerspelling recognition, including machine learning models, data preprocessing techniques, and integration with existing communication platforms and devices.

User Experience and Usability Testing: Evaluation of the user experience and usability of fingerspelling recognition systems among Deaf and Hard of Hearing users, with a focus on factors such as accuracy, speed, and ease of use.

Potential Applications and Future Directions: Discussion of potential applications of fingerspelling recognition technology, such as text entry for web search, map directions, and texting, as well as future directions for research and development in this field.

By analyzing these key areas, this research paper aims to further the development of sign language recognition technology and promote accessibility and inclusivity for the Deaf and Hard of Hearing community. Through the utilization of licensed data provided by Google, the study seeks to leverage existing resources to create innovative solutions that empower individuals with disabilities and promote equal access to information and resources in the digital age.

7. Research Methodology

7.1 Data Collection & Understanding:

- This will involve the collection of hand tracking points data representing ASL fingerspelling gestures. These data points will be obtained from the licensed dataset provided by Google, ensuring a diverse range of gestures and conditions for analysis.

7.2 Preprocessing:

- Preprocess the data frames to enhance image quality and remove noise.
- Perform background subtraction to isolate the hand region from the background.

7.3 Hand Detection and Tracking:

- Apply hand detection algorithms to locate and extract the hand region from each frame.
- Implement hand tracking techniques to track the movement of the hand across consecutive frames.

7.4 Feature Extraction:

- Extract relevant features from the hand region, such as hand shape, motion trajectory, and spatial-temporal information.
- Represent the features in a suitable format for input to the recognition model.

7.5 Model Selection:

- Choose appropriate techniques for machine learning that can identify sign language such as deep learning architectures (e.g., convolutional neural networks, recurrent neural networks) or transformer-based models.
- Train the selected model using labeled data to learn the mapping between input features and corresponding sign language gestures.

7.6 Real-Time Recognition:

- Implement the trained model to perform real-time recognition of ASL fingerspelling gestures.
- Process the extracted features from the hand region and feed them into the recognition model.
- Generate predictions or probabilities for each recognized gesture based on the model's output.

7.7 Translation to Text:

- Develop algorithms to translate recognized ASL fingerspelling gestures into text in real-time.
- Map each recognized gesture to its corresponding alphanumeric character or word using a predefined mapping or lookup table.

7.8 Integration and Deployment:

- Integrate the recognition system into existing communication platforms or devices, such as smartphones, tablets, or video conferencing software.
- Deploy the system for use by Deaf and Hard of Hearing individuals in various settings, including educational, professional, and social environments.

a. 4.9. Evaluation and Testing:

- Use relevant measures such as accuracy, precision and recall assessing the recognition system’s performance, F1-score as well as normalized total levenshtein distance.
- Conduct usability testing with end-users to assess the system's effectiveness, user happiness and ease of usage

Figure 1 below following outlines the overarching workflow methodology we will adhere to throughout this research

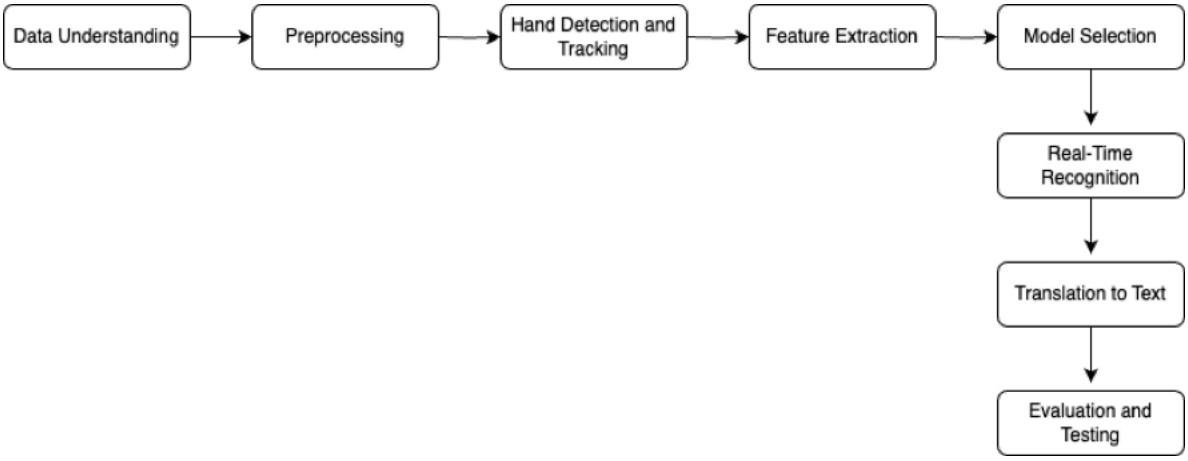

Fig 1: Methodology workflow of project

8. Required Resources

8.1 Hardware Resources:

9. **Cameras or Recording Devices:** High-resolution cameras or recording devices capable of capturing video sequences of ASL fingerspelling gestures. This may include webcams, digital cameras, or specialized video recording equipment.
10. **Computing Devices:** Powerful computing devices such as desktop computers or laptops with sufficient processing power and memory to perform real-time video processing, feature extraction, and machine learning model training.
11. **Graphics Processing Units (GPUs):** GPUs are essential for accelerating the training and inference processes of deep learning models, significantly reducing computation time.
12. **Storage Devices:** Large-capacity storage devices for storing the collected video data, labeled datasets, and trained machine learning models.

8.2 Software Resources:

13. **Python Programming Language:** Python is widely used in machine learning and computer vision applications due to its extensive libraries and frameworks. Essential libraries for this research include NumPy, OpenCV, TensorFlow, PyTorch, and scikit-learn.
14. **Machine Learning Frameworks:** Deep learning frameworks such as TensorFlow or PyTorch are essential for developing and training convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other deep learning models for sign language recognition.
15. **Computer Vision Libraries:** Libraries such as OpenCV provide essential tools and algorithms for image preprocessing, feature extraction, hand detection, and tracking.
16. **IDEs and Development Tools:** Integrated Development Environments (IDEs) such as Jupyter Notebook, PyCharm, or Visual Studio Code facilitate code development, debugging, and experimentation.
17. **Data Annotation Tools:** Software tools for annotating and labeling video data, such as Labeling or VGG Image Annotator, are necessary for creating labeled datasets for model training.
18. **Documentation and Version Control:** Documentation tools like LaTeX or Markdown, coupled with version control systems like Git, enable efficient documentation of research methodologies, results, and code repositories for reproducibility and collaboration.
19. **Deployment Platforms:** Depending on the deployment strategy, cloud platforms such as AWS, Google Cloud, or Microsoft Azure may be used for deploying the recognition system to scalable and accessible environments.

9. Research Plan

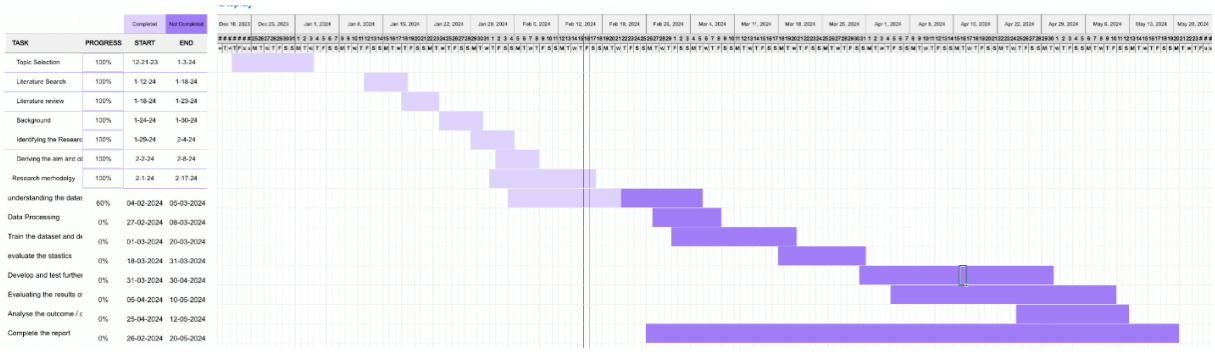

Fig 2. Gantt chart showing the project plan

		Completed	Not Completed
TASK	PROGRESS	START	END
Topic Selection	100%	12-21-23	1-3-24
Literature Search	100%	1-12-24	1-18-24
Literature review	100%	1-18-24	1-23-24
Background	100%	1-24-24	1-30-24
Identifying the Researc	100%	1-29-24	2-4-24
Deriving the aim and ob	100%	2-2-24	2-8-24
Research merhodolgy	100%	2-1-24	2-17-24
understanding the datas	60%	04-02-2024	05-03-2024
Data Processing	0%	27-02-2024	08-03-2024
Train the dataset and de	0%	01-03-2024	20-03-2024
evaluate the stastics	0%	18-03-2024	31-03-2024
Develop and test further	0%	31-03-2024	30-04-2024
Evaluating the results of	0%	05-04-2024	10-05-2024
Analyse the outcome / c	0%	25-04-2024	12-05-2024
Complete the report	0%	08-05-2024	20-05-2024

Fig 3. Gantt chart data table of the project plan