People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 May 1945 – Guelma
Faculty of Mathematics, Computer Science, and Material Sciences
Computer Science Department

Final Thesis

Field:
Computer Science

Option:
Science And Technology Of Information And Communication

Topic

A Deep Learning-Based System for Bidirectional Communication between Deaf and Hearing Users using Hand Gesture Sign Language

Jury Members:
President: Dr. *Wafa Louafi*
Supervisor: Dr. *Samir Hallaci*
Examiner: Dr. *Hiba Abdelmoumene*
Examiner: Dr. *Lazhar Hani Gueddoum*

Presented by:
GUERGOUR Ghada Malak

Academic Year 2024/2025

Acknowledgments

لَئِن شَكَرْتُمْ لَأَزِيدَنَّكُمْ
Qur'an, 14:7

First and foremost, I extend my deepest gratitude to **Allah, the Most Gracious, the Most Merciful**, who granted me strength, patience, and clarity throughout this journey.

I am sincerely thankful to my **family**, my unwavering support system, whose prayers, unconditional love, and constant encouragement lifted me during moments of doubt and helped me persevere through every challenge.

To those who stood by me, thank you for believing in me, even when I doubted myself. Your support, your words, and sometimes even your silence planted the seeds of resilience within me.

I would also like to express my heartfelt thanks to **Dr. Samir Hallaci** for his invaluable advice and guidance.

I extend my sincere gratitude to the **members of the jury** for their time, constructive remarks, and valuable contributions. Their insightful feedback and evaluation helped elevate the quality of this work.

This thesis is a piece of my soul, shaped by faith and love.

Dedication

*To my **Ohana**.*

Ohana means family, and family means
nobody gets left behind or forgotten.

ملخص

يظلُّ التواصل الفعال بين الأفراد الصمّ وضعاف السمع تحديًا مجتمعيًا رئيسيًا، خاصة في السياقات التي لا تُفهم فيها لغة الإشارة من قبل عامة الناس. وعلى الرغم من أن لغة الإشارة هي اللغات الطبيعية الكاملة، إلا أن نقص المعرفة المشتركة بهذه اللغة لا يزال يعيق إمكانية الوصول والإندماج في مجالات حيوية مثل التعليم والرعاية الصحية والتوظيف. استجابةً لهذه المشكلة، تقدم هذه الأطروحة نظامًا قائمًا على التعلم العميق لتمكين التواصل ثنائي الاتجاه في الزمن الحقيقي بين المستخدمين الصمّ وضعاف السمع، باستخدام لغة الإشارة المعتمدة على حركات اليد كوسيط أساسي.

يُدمج النظام المقترح تقنيات الرؤية الحاسوبية ومعالجة اللغة الطبيعية والرسوم المتحركة ثلاثية الأبعاد لترجمة لغة الإشارة إلى لغة منطوقة أو مكتوبة والعكس. تم تنفيذ وتقييم ثلاث بنى نموذجية: CNN-LSTM وMediaPipe-Bi-LSTM وMediaPipe-GCN-BERT. بينما حقق نموذج MediaPipe-LSTM دقةً تزيد عن 98% في مهام التعرف على الإشارات المنفردة، إلا أنه أظهر قيودًا في معالجة التسلسلات الطويلة بسبب بنيته المعتمدة على الذاكرة. للتغلب على هذا، تم اعتماد نهج قائم على الرسوم البيانية، حيث تم تمثيل العلاقات المكانية بين معالم اليد باستخدام شبكات التلافيف البيانية (GCNs)، مدمجةً مع تضمينات BERT للسياق الدلالي. أدى ذلك إلى تحسين الأداء والقدرة على التعميم في التعامل مع الإشارات المعقدة والمستمرة.

تم نشر النظام كتطبيق جوال باستخدام Native React وExpo، متكاملًا مع تقنيات التعرف على الكلام في الزمن الحقيقي، وترجمة الإشارة إلى نص. أكدت التقييمات التجريبية باستخدام التحقق المتقاطع ومصفوفات الالتباس ومعدل خطأ الكلمات (WER) على متانة النظام ودقته وقابليته للاستخدام في سيناريوهات الزمن الحقيقي. تمثل هذه المساهمة خطوة مهمة نحو تطوير تقنيات تواصلية شاملة وميسورة لمجتمعات الصمّ وضعاف السمع.

الكلمات المفتاحية: التعرف على لغة الإشارة، التعلم العميق، MediaPipe، LSTM، شبكات التلافيف البيانية، BERT، التواصل في الزمن الحقيقي، إمكانية الوصول، الذكاء الاصطناعي المرتكز على الإنسان.

Abstract

Effective communication between Deaf and hearing individuals remains a major societal challenge, particularly in contexts where sign language is not understood by the general population. Sign languages are complete natural languages, yet the lack of shared linguistic knowledge continues to hinder accessibility and inclusion in vital domains such as education, healthcare, and employment. In response to this issue, this thesis presents a deep learning-based system for real-time, bidirectional communication between Deaf and hearing users, using hand gesture sign language as a primary medium.

The proposed system integrates computer vision, and 3D animation technologies to translate between sign language and spoken/written language. Three model architectures were implemented and evaluated: CNN-LSTM, MediaPipe-Bi-LSTM, and MediaPipe-GCN-BERT. While the MediaPipe-LSTM model achieved over 98% accuracy on isolated gesture recognition tasks, it exhibited limitations in handling longer sequences due to its memory-based structure. To overcome this, a graph-based approach was adopted, where spatial relationships between hand landmarks were modeled using Graph Convolutional Networks (GCNs), combined with BERT embeddings for semantic context. This resulted in improved generalization and performance on complex and continuous gestures.

The system was deployed as a mobile application built with React Native and Expo, integrating real-time speech recognition, and sign-to-text translation. Experimental evaluations using cross-validation, confusion matrices, and Word Error Rate (WER) confirmed the robustness, accuracy, and usability of the platform in real-time scenarios. This work contributes a significant step toward accessible and inclusive communication technology for the Deaf and hard-of-hearing communities.

Keywords: Sign Language Recognition, Deep Learning, MediaPipe, LSTM, Graph Convolutional Network, BERT, Real-Time Communication, Accessibility, Human-Centered AI

Résumé

La communication efficace entre les personnes sourdes et entendantes demeure un défi sociétal majeur, en particulier dans les contextes où la langue des signes n'est pas comprise par la population générale. Les langues des signes sont des langues naturelles à part entière, pourtant l'absence de connaissances linguistiques partagées continue d'entraver l'accessibilité et l'inclusion dans des domaines essentiels tels que l'éducation, la santé et l'emploi. Pour répondre à cette problématique, ce mémoire présente un système basé sur l'apprentissage profond, permettant une communication bidirectionnelle en temps réel entre utilisateurs sourds et entendants, en utilisant la langue des signes gestuelle comme principal moyen de communication.

Le système proposé intègre des technologies de vision par ordinateur et d'animation 3D pour assurer la traduction entre la langue des signes et la langue orale/écrite. Trois architectures de modèles ont été implémentées et évaluées : CNN-LSTM, MediaPipe-BiLSTM et MediaPipe-GCN-BERT. Bien que le modèle MediaPipe-LSTM ait atteint une précision supérieure à 98% sur des tâches de reconnaissance de gestes isolés, il a montré des limites dans le traitement de séquences longues en raison de sa structure basée sur la mémoire. Pour surmonter cela, une approche basée sur les graphes a été adoptée, où les relations spatiales entre les points clés de la main sont modélisées à l'aide de réseaux de neurones convolutifs sur graphes (GCNs), combinés avec des embeddings BERT pour le contexte sémantique. Cela a permis une meilleure généralisation et des performances accrues sur les gestes complexes et continus.

Le système a été déployé sous forme d'application mobile développée avec React Native et Expo, intégrant la reconnaissance vocale en temps réel ainsi que la traduction de la langue des signes en texte. Les évaluations expérimentales, réalisées à l'aide de la validation croisée, de matrices de confusion et du taux d'erreur de mots (WER), ont confirmé la robustesse, la précision et la convivialité de la plateforme dans des scénarios en temps réel. Ce travail constitue une avancée significative vers une technologie de communication accessible et inclusive pour les communautés sourdes et malentendantes.

Mots-clés : Reconnaissance de la langue des signes, Apprentissage profond, MediaPipe, LSTM, Réseau de neurones convolutifs sur graphes, BERT, Communication en temps réel, Accessibilité, Intelligence artificielle centrée sur l'humain

Contents

List of Figures

"When you wake up one day and you don't hear the refrigerator hum or you don't hear paper rustle, it's scary. You want to deny it, say it's temporary, just a head cold, you'll hear better later. But you don't. . . . People just don't understand what it's like." Eric[1]

General Introduction

Communication is a fundamental human need that transcends cultural, linguistic, and social boundaries. However, for over 430 million people worldwide living with significant hearing loss [2], communication remains a daily challenge, particularly in interactions with hearing individuals. The language barrier between Deaf and hearing communities contributes to a lack of inclusion in essential areas such as education, healthcare, employment, and social services.

Sign languages, which are full-fledged natural languages with their own grammar, syntax, and structure, represent the primary mode of communication for many Deaf individuals [3]. They are visual-gestural languages that convey meaning through manual signs, facial expressions, and body posture. Despite this, the general population—especially hearing individuals, often lacks the knowledge or training necessary to understand sign languages, leading to communication breakdowns and societal exclusion. In response to this issue, the integration of artificial intelligence (AI) into sign language processing presents an innovative and inclusive solution to bridge this gap [4].

This thesis proposes a deep learning-based system for real-time, bidirectional communication between Deaf and hearing users. The system combines gesture recognition based on skeletal keypoints, and avatar-based sign language generation to enable seamless translation between sign and spoken language. It is designed to operate on mobile devices with low computational overhead, making it accessible in both academic and daily-life contexts.

The **first chapter** provides a historical and sociolinguistic overview of sign languages. It traces their development, marginalization through oralist policies such as those institutionalized after the 1880 Milan Congress, and their eventual recognition as legitimate languages [5]. It also examines the structural differences between signed and spoken languages and highlights the social consequences of communication barriers, particularly in sensitive contexts like healthcare. The chapter emphasizes the need for automated translation tools that respect the linguistic richness and sociocultural significance of sign languages.

Chapter two reviews the current state of the art in Sign Language Recognition (SLR). It discusses traditional machine learning methods (e.g., SVM, Random Forest, KNN) and modern deep learning models, such as Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), Graph Convolutional Networks (GCN), and transformer-based models like BERT. Additionally, it highlights datasets (e.g., WLASL, RWTH-PHOENIX-Weather) and evaluation metrics (e.g., accuracy, F1-score, BLEU, ROUGE, Word Error Rate (WER)) used to benchmark these systems. The chapter also identifies key research gaps, such as limited support for continuous signs and lack of multimodal context understanding.

Chapter three outlines the methodological framework and system design. It details the implementation of three distinct model architectures: CNN-LSTM, MediaPipe-LSTM, and

MediaPipe-GCN-BERT—for gesture classification. The system leverages MediaPipe for real-time hand tracking, TensorFlow for deep learning, and Blender for 3D avatar generation. A focus is also placed on data preprocessing, model training pipelines, and the algorithmic complexities involved in building the sign-to-speech and speech-to-sign modules. This chapter emphasizes the lightweight design required for real-time performance on CPUs, supported by frame rates exceeding 6 FPS in practice.

Chapter four presents the full system implementation, experimental results, and performance analysis. The proposed application integrates both voice recognition and sign language translation into a mobile interface developed with React Native and Expo. The models demonstrated strong performance in terms of accuracy, robustness under varying lighting conditions, and ability to distinguish between visually similar gestures.

For instance, the MediaPipe-Bi-LSTM model achieved over **98.2%** accuracy on the validation set for isolated gesture classification. However, its architecture, which relies on sequential memory mechanisms, showed limitations when processing longer or continuous sign sequences. Specifically, the LSTM struggled to retain meaningful spatial-temporal dependencies over extended input lengths, often focusing only on the most dominant hand movements while neglecting subtle contextual cues. This constraint made it insufficient for real-world scenarios where gestures form part of fluid and semantically rich sentence structures.

To address this, we adopted a graph-based architecture by integrating Graph Convolutional Networks (GCN) with BERT embeddings. The MediaPipe-GCN-BERT pipeline leveraged the spatial structure of hand landmarks by modeling them as nodes in a graph, enabling the system to capture joint-level relationships more effectively. Combined with the contextual understanding capabilities of BERT, this architecture provided enhanced generalization on complex and visually similar gestures, as well as on longer sign sequences. Evaluation through cross-validation, confusion matrices, and Word Error Rate confirmed the practical relevance, robustness, and real-time stability of the proposed solution in dynamic communication contexts. Specifically, MG-BERT achieved a 97.5% recognition rate and maintained an average inference speed of 6.85 FPS on standard CPU-based systems.

This research presents not just a technical solution but a human-centered innovation aimed at reducing societal barriers. By uniting artificial intelligence with sign language linguistics, this work contributes to the broader goal of promoting accessibility, inclusion, and mutual understanding between Deaf and hearing individuals. The findings also open promising pathways for applications in education, healthcare, customer service, and smart environments.

Chapter 1

Sign Language and Ai-Driven Solutions for Communication Challenges

1.1 Introduction

According to statistics from the World Health Organization (WHO) [2], approximately 5% of the global population suffers from hearing impairment. Furthermore, projections by the United Nations [6] estimate that by 2050, the number of individuals with hearing loss will reach 900 million.

Communication is fundamental to human interaction, enabling individuals to express their needs, emotions, and ideas. For the Deaf community, effective communication is equally essential to ensure inclusion and active participation in society. However, Deaf individuals often face significant challenges when communicating with hearing individuals, leading to potential isolation and exclusion. Bridging this communication gap is crucial not only for social inclusion but also for fostering mutual understanding and equality.

Sign language comprises multiple elements, including hand gestures, facial expressions, body movements, and finger positioning. These aspects significantly impact the effectiveness of SLR systems. Artificial intelligence (AI) has emerged as a powerful tool in developing Sign Language Recognition (SLR) systems, which aim to interpret complex gestures and facial expressions. SLR is inherently challenging due to the diversity of sign languages and the intricate combination of manual and non-manual features involved. In this section, we explore the fundamental characteristics of sign language, the unique challenges associated with its recognition, and the necessity of advanced AI-driven solutions. By addressing these challenges, we can enhance communication accessibility, empowering the Deaf community through seamless interaction with technology.

1.2 The History of Sign Language

In this section, we explore the evolution of sign languages and their significance within the Deaf community. From their early foundations to their recognition as complete and independent languages, sign languages have played a vital role in communication, culture, and identity for Deaf individuals. This historical overview sheds light on the challenges and victories

experienced by the Deaf community in preserving and promoting their linguistic heritage.

The history of sign language is deeply connected to the broader struggles for Deaf autonomy, education, and cultural identity. From the late 18th century to the mid-20th century, sign languages evolved not only as tools of communication but also as symbols of resistance and empowerment within Deaf communities [7].

1.2.1 Early Recognition and Institutionalization

The late 18th and early 19th centuries marked a turning point in the recognition of sign language. The establishment of the first schools for the Deaf, such as the American School for the Deaf in Hartford 1.1, Connecticut (1817), played a pivotal role in legitimizing sign language as a medium of instruction [7]. Many of these institutions were founded by Deaf individuals or hearing allies who supported manual communication methods. During this period, sign language thrived as a cornerstone of Deaf identity and social integration, fostering the growth of vibrant Deaf communities.

Figure 1.1: First school for the Deaf in the USA, Hartford, 1817.

Legacy and Continuity of Early Deaf Schools

Remarkably, many of these pioneering institutions remain active today, preserving their historical missions while adapting to modern educational needs. The American School for the Deaf (ASD), for instance, continues to operate in Hartford [8] (see Figure 1.2), offering bilingual (ASL/English) education and serving as a cultural touchstone for the Deaf community [7]. Similarly, Europe's oldest Deaf school, the Institut National de Jeunes Sourds de Paris (founded in 1760), still functions as a center for Deaf education and advocacy [5].

These schools' endurance underscores their foundational role in institutionalizing sign language and sustaining Deaf cultural heritage. Their ongoing relevance highlights the resilience of Deaf communities in the face of shifting pedagogical trends, such as the oralist movement of the late 19th century [9, 10].

Figure 1.2: American School for the Deaf campus, Hartford.

1.2.2 Deaf Resistance and the Preservation of Sign Language

Despite the rise of oralism, the Deaf community actively resisted the erasure of their language and cultural identity. Organizations such as the *National Association of the Deaf (NAD)*, founded in 1880, became key advocates for the preservation and recognition of sign language [7]. Deaf leaders emphasized that sign language was not only a natural and effective form of communication but also essential to the intellectual and social development of Deaf individuals. Through education, advocacy, and cultural expression, the Deaf community fought to maintain their linguistic heritage and challenge the stigmatization imposed by oralist ideologies.

1.2.3 The Rise of Oralism and the Suppression of Sign Language

By the mid-19th century, a new educational movement known as oralism emerged, challenging the use of sign language. Oralism promoted teaching Deaf individuals to speak and lip-read, excluding the use of sign language. This approach gained significant support after the 1880 Milan Congress, where a majority of hearing educators voted in favor of oralist methods [7]. Following this decision, sign language was systematically banned from many schools, driven by ideologies of normalization and eugenics that viewed it as an obstacle to social assimilation. As a result, sign language was marginalized, despite its importance for communication and cultural continuity within Deaf communities.

Despite the rise of oralism, the Deaf community actively resisted the erasure of their language and cultural identity. Organizations such as the *National Association of the Deaf (NAD)*, founded in 1880, became key advocates for the preservation and recognition of sign language [7]. Deaf leaders emphasized that sign language was not only a natural and effective form of communication but also essential to the intellectual and social development of Deaf individuals. Through education, advocacy, and cultural expression, the Deaf community fought to maintain their linguistic heritage and challenge the stigmatization imposed by oralist ideologies.

1.2.4 Revival and Recognition in the 20th Century

The mid-20th century witnessed a revival of interest in sign languages, spurred by academic research and the growing visibility of Deaf culture. Pioneering linguists such as William Stokoe demonstrated that American Sign Language (ASL) possessed a complete linguistic structure, with its own grammar and syntax [7]. This breakthrough helped shift public and academic

perceptions, legitimizing sign language as a true language. The rise of Deaf activism further reinforced this momentum, culminating in significant events like the 1988 *Deaf President Now* movement at Gallaudet University, which emphasized the importance of Deaf leadership and sign language in higher education.

1.3 Difference between Sign Language and Spoken Language

Sign Language (SL) [11] is a visual-gestural form of communication primarily used by Deaf and hard-of-hearing individuals. Unlike spoken languages that rely on vocal and auditory faculties, SL uses hand gestures, body movements, and spatial orientation, and is processed visually. It has its own grammar and syntax, distinct from spoken or written languages [12].

A Sign Language Recognition (SLR) system translates SL into text or speech using digital image processing and classification techniques. It typically involves gesture modeling, analysis, recognition, and application [13].

SL is a standalone language with its own alphabet, numerals, and vocabulary, although generally more limited than spoken languages. In many developing countries, SL is still evolving, but progress in computer-based recognition is notable [12]. Additionally, SL often mirrors the alphabet and numerals of its corresponding spoken language, and its vocabulary is shaped by the cultural context of its users.

1.4 Communication Barriers between Deaf and Hearing Individuals

Communication between Deaf and hearing individuals is hindered by linguistic and perceptual differences. With over 300 distinct sign languages globally, each with its own grammar and vocabulary, cross-linguistic communication poses a significant challenge [14]. Furthermore, the medical model views deafness as a disability needing correction, while the social model sees it as a cultural and linguistic identity [1, 15]. This affects communication preferences, with many Deaf individuals favoring sign language over spoken or written forms [15].

Hearing individuals often lack the skills to interpret sign language, while Deaf individuals may struggle with spoken language, leading to frequent miscommunication—especially in critical contexts like healthcare [1]. Lip reading is commonly overestimated, even though only 30–45% of English sounds are visibly distinguishable on the lips [1].

To overcome these barriers, technological solutions like real-time sign language translation using machine learning and computer vision are essential [16]. Additionally, greater awareness and professional training are necessary to promote inclusion and improve accessibility for the Deaf community [14, 15].

1.5 Importance of Bridging the Communication Gap for Inclusivity and Accessibility

Bridging the communication gap between Deaf and hearing individuals is essential for fostering understanding and dismantling misconceptions. Misunderstandings often stem from a lack of awareness about Deaf culture and communication methods, leading to marginalization and exclusion [1]. Effective communication is the cornerstone of an inclusive society, ensuring that all individuals, regardless of hearing ability, can fully engage in social, educational, and professional environments. Too often, the hearing community perceives Deaf individuals as "disabled" rather than recognizing them as part of a linguistic and cultural minority, further deepening the divide [1, 17].

1.5.1 Importance of Communication in Health Care

Effective communication is fundamental in health care, ensuring that patients fully understand medical procedures, diagnoses, and treatments. However, individuals who are Deaf or hard of hearing face significant communication barriers that can lead to discomfort, fear, and even medical errors. As highlighted in [1], inadequate communication can have severe consequences, including misdiagnosis, medication errors, and distress during medical procedures.

One striking example is the experience of a patient undergoing a gynecological examination: *"They didn't tell me what they were going to do. There I was in the stirrups—I couldn't see what was going on. The doctor didn't say to me, 'This might be uncomfortable,' or tell me how much pain to expect. I never went again"* [1].

Similarly, a male patient undergoing his first testicular examination described his fear and confusion: *"I was scared. I didn't know if I was being molested or raped or if this was a sexual advance... A hearing doctor with a hearing patient will talk through the entire exam, but when the patient is Deaf, they just do it. Some doctors keep on talking. They forget I'm Deaf"* [1].

These experiences underscore the necessity of patient-centered communication strategies. Research on American Sign Language (ASL) emphasizes that ASL is not merely a direct translation of English but a fully developed language with its own structure and grammar [17]. Miscommunication arises when health care providers assume that written notes or lip reading are sufficient methods of interaction. In reality, only 30–45% of English sounds are distinguishable through lip reading, leading to misunderstandings and patient frustration [1].

The development of technologies that bridge this communication gap is not only about facilitating interaction but also about fostering a sense of belonging, dignity, and respect for the Deaf community [16]. These technologies play an integral role in promoting a sense of equality, ensuring that Deaf individuals have the same opportunities to thrive in all aspects of life.

1.6 Challenges and Limitations in Sign Language

The diversity of sign languages worldwide presents significant challenges, as each has distinct grammar, vocabulary, and syntax shaped by unique cultural and linguistic contexts. Languages such as ASL, BSL, and ArSL are not mutually intelligible [18, 19, 17].

Sign language recognition research must distinguish between **isolated sign recognition** and **continuous sign language interpretation**. The former involves identifying individual signs, while the latter requires understanding complex, motion-based sequences [20].

Translation between spoken and sign languages is complicated by structural differences. Issues include one-to-one, one-to-many, and many-to-one mappings of semantic concepts to signs, as noted in Spanish–LSE and Arabic–ArSL translation efforts [21, 22].

Unlike spoken languages, sign languages allow for **simultaneous expression** through hands, facial expressions, and body movement. For instance, Thai Sign Language uses parallel spatial elements, contrasting with the linear structure of Thai speech [23, 24, 25].

Key features of sign languages include:

- **Non-manual markers** (e.g., facial expressions, head movements) for grammar and meaning.
- **Use of space** to encode phonological and lexical information.
- **Context-dependent signs** that may serve as different parts of speech.
- **Classifiers** that convey shape, movement, and object characteristics.
- **Syntax variations** often diverging from the common Subject-Verb-Object (SVO) order [26].

These unique features introduce complexities in automated recognition and translation, requiring sophisticated systems to ensure accurate and meaningful communication.

1.6.1 Exploring the Diversity of Sign Languages: Illustrative Examples

To highlight the variations and diversity inherent in sign languages, we explore examples from three prominent systems: Arabic Sign Language (ArSL), British Sign Language (BSL), and American Sign Language (ASL).

Arabic Sign Language (ArSL)

Arabic Sign Language (ArSL) is a distinct visual-gestural language with its own grammar, vocabulary, and structure, which differs from both spoken Arabic and other sign languages. Although the League of Arab States and ALECSO attempted to standardize ArSL in 1999 with a dictionary comprising approximately 3,200 signs [18], regional variations remain prevalent, complicating mutual understanding among signers [27]. ArSL generally follows a subject-first word order, with question words commonly placed at the end of interrogative sentences, as illustrated in Figure 1.3 [28].

ArSL	Arabic
؟ انت اسم ماذا	ما اسمك؟
? Ant Asm mA*A	mA Asmk?
? YOU NAME WHAT	What is your name?

Figure 1.3: Structure of a question sentence in Arabic Sign Language (ArSL)

Unlike spoken Arabic, ArSL does not employ inflections for gender or number; instead, it uses distinct signs. Non-manual markers—such as facial expressions, eye gaze, and body posture—are essential for conveying grammatical nuances and emotions [29]. However, compared to American Sign Language (ASL) and British Sign Language (BSL), ArSL lacks robust Sign Language Recognition (SLR) systems due to limited datasets [30]. Recent initiatives, such as the dataset shown in Figure 1.4, aim to address this gap and support the development of ArSL recognition technologies [31].

Figure 1.4: Sample signs from the Arabic Sign Language (ArSL) dataset

British Sign Language (BSL)

British Sign Language (BSL) is the primary sign language used by the Deaf community in the United Kingdom [32]. It is a fully independent language, with its own grammar, syntax, and vocabulary, and is entirely distinct from spoken English [19]. BSL should not be confused with American Sign Language (ASL) or other national sign languages, as it significantly differs in both structure and lexicon [33].

Technological advances have contributed to BSL recognition, such as the intelligent vision system developed by Quinn and Olszewska. This system leverages machine learning and computer vision—specifically, Support Vector Machines (SVM)—to recognize BSL signs in real time, thereby enhancing communication accessibility for Deaf users [34]. Figure 1.5 depicts the BSL manual alphabet.

Figure 1.5: British Sign Language (BSL) manual alphabet

American Sign Language (ASL)

American Sign Language (ASL) is one of the most widely used sign languages, particularly in the United States and parts of Canada. ASL is a complete and natural language with its own unique grammar, vocabulary, and syntax, which are markedly different from spoken English [17]. It conveys meaning through a combination of handshapes, facial expressions, and body movements. Importantly, ASL does not follow English grammatical rules and has its own linguistic structure.

Figure 1.6 presents the ASL alphabet and number gestures, showcasing the distinct hand configurations used for each letter and digit. These differ significantly from those used in British Sign Language (BSL).

Figure 1.6: Manual alphabet and number gestures in American Sign Language (ASL)

1.7 Artificial Intelligence

Artificial Intelligence (AI) refers to the ability of machines and computer systems to perform tasks that typically require human intelligence. This includes decision-making, pattern recognition, language understanding, and adaptive learning. AI systems may emulate human cognitive processes or adopt approaches inspired by natural phenomena such as evolution and neural activity [35].

One of the practical applications of AI is in sign language recognition, where AI techniques analyze hand gestures, movements, and facial expressions to interpret and translate them into spoken or written language. This application plays a crucial role in enhancing accessibility for the deaf and hard-of-hearing communities.

1.8 Sign-to-Text or Spoken Language Translation for Deaf Individuals

The primary goal of sign language translation systems is to accurately capture and interpret signed gestures, converting them into coherent text or spoken language. This process is inherently complex due to the distinct grammar, syntax, and spatial dynamics of sign languages, which differ significantly from spoken language structures [36].

Traditional Sign Language Recognition (SLR) approaches have focused largely on isolated sign classification. While useful for recognizing individual gestures, such methods often fail to capture the rich linguistic and contextual nuances present in continuous sign language. Recent advancements in Neural Sign Language Translation (SLT) have introduced deep learning-based sequence-to-sequence models, often augmented with attention mechanisms, to translate continuous sign language videos into semantically meaningful spoken language sentences [36]. These models are capable of learning complex temporal dependencies and aligning sign gestures with spoken language constructs, thereby overcoming issues related to word order and syntactic mismatch.

1.9 Spoken Language to Equivalent Sign Translation for Hearing Individuals

Conversely, translating spoken language into its equivalent sequence of sign language gestures enables hearing individuals to communicate effectively with the deaf community. This task involves not only transcribing speech but also rendering it into a visual-gestural format that accurately represents the linguistic structure and expressive elements of the target sign language [16].

The translation process relies heavily on Artificial Intelligence, particularly Natural Language Processing (NLP), to understand the semantic and syntactic context of spoken input. Once the meaning is extracted, computer vision and deep learning models are employed to synthesize corresponding sign gestures. These models must take into account several parameters, including hand shape, movement trajectory, palm orientation, facial expressions, and spatial positioning [16].

One of the key challenges in this direction stems from the non-linear, multidimensional nature of sign languages. While spoken languages typically follow a linear sequence, sign languages convey information through simultaneous and spatially distributed channels. Rule-based translation systems have struggled with this complexity. However, recent deep learning frameworks have demonstrated greater success by modeling sequential dependencies and contextual features within multimodal data [16].

Before further exploring the specific AI architectures and training strategies used in sign language translation, it is helpful to reaffirm the relevance of AI in this context. AI technologies—including speech recognition, natural language understanding, and gesture synthesis—serve as the foundational tools that enable effective and scalable sign language communication systems.

1.9.1 Computer Vision in Sign Language Recognition

Computer vision, inspired by the mechanisms of human visual perception [37], plays a pivotal role in sign language recognition systems. In this context, cameras act as the ”eyes” that capture visual input, while AI-powered algorithms serve as the ”brain,” processing and interpreting the information. The output is a meaningful translation that bridges communication gaps and enhances accessibility for the Deaf community.

Computer vision is essential for developing systems capable of recognizing and interpreting sign language gestures. These systems leverage advanced techniques in image processing, machine learning, and deep learning [38] to detect hand gestures, facial expressions, and body movements. The processed data is then translated into text or speech, enabling real-time sign recognition and facilitating communication between Deaf and hearing individuals [16].

Machine learning (ML), a subfield of artificial intelligence, lies at the heart of sign language recognition. While AI refers broadly to machines performing intelligent tasks, ML specifically focuses on algorithms that learn from data to make predictions or decisions [35]. In the context of sign language, ML algorithms classify visual inputs such as hand gestures and facial cues to interpret signs accurately.

Deep learning, a more specialized subset of ML, employs neural networks to tackle complex tasks like image recognition and sequential data modeling [35]. Unlike traditional ML, deep learning models can learn intricate features from raw input data (e.g., video frames), eliminating the need for manual feature extraction [16].

In sign language recognition, deep learning excels at modeling both spatial and temporal dynamics. Convolutional Neural Networks (CNNs) extract spatial features from individual frames, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) capture temporal dependencies across gesture sequences [16].

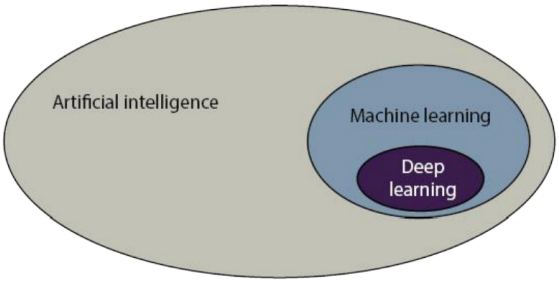


Figure 1.7: Hierarchical relationship among AI, ML, and Deep Learning.

Natural Language Processing (NLP), another branch of AI, focuses on enabling machines to understand and generate human language in a meaningful way [39]. In spoken-to-sign language translation, NLP processes spoken input, extracts meaning, and generates corresponding sign sequences while considering syntax, semantics, and context [40]. NLP complements computer vision in bidirectional translation systems between spoken and sign languages.

1.9.2 Sign Language Recognition

The development of Sign Language Recognition (SLR) systems has accelerated significantly due to advancements in artificial intelligence and deep learning technologies.

To design effective systems, it is essential to understand the various types of SLR, as illustrated in Figure 1.8 [41].

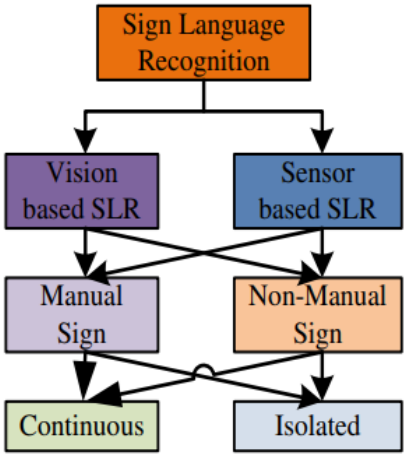


Figure 1.8: Types of Sign Language Recognition

1.9.3 Vision-Based Approach

In the vision-based approach, cameras capture hand, palm, and finger movements from video input. Image processing algorithms extract features that are subsequently classified. While suitable for real-time environments, this approach is sensitive to lighting conditions, background noise, and image blurriness. Therefore, effective preprocessing, feature extraction, and classification are critical for accuracy [41].

1.9.4 Sensor-Based Approach

Sensor-based methods involve physically mounted devices—such as gloves or motion sensors—that detect finger trajectories, hand articulations, and head movements. These systems offer higher precision in controlled environments and are less affected by external noise. Compared to vision-based systems, they often provide more consistent data, though at the cost of wearability and user comfort [41].

1.9.5 Continuous and Isolated Sign Language Recognition

SLR can be further classified based on recognition style—**isolated** versus **continuous**—and on the nature of gestures—**manual** versus **non-manual**.

Manual SLR recognizes hand configurations, orientations, and motion paths. In **isolated manual SLR**, each sign is recognized as a distinct unit, simplifying classification. In contrast, **continuous manual SLR** requires segmenting overlapping gestures within sequences, often addressed using models like Hidden Markov Models (HMMs) and LSTMs [41].

Non-manual SLR includes facial expressions, eyebrow movements, lip shapes, and head tilts that contribute to grammatical and emotional nuance. **Continuous non-manual SLR** poses significant challenges due to the need for temporal modeling and multimodal fusion. Deep learning techniques—especially CNNs and RNNs—are well-suited for managing these dynamics and have shown promising results in tackling occlusion and feature variability.

1.9.6 Hand Gesture Recognition

Hand gestures can be divided into two main categories:

Dynamic Hand Gesture Recognition

Dynamic gestures involve movement over time and are typically processed through video analysis. Earlier systems relied on hand-crafted features such as motion trajectories or body skeletons [42, 43, 44, 45]. Recently, spatial-temporal deep learning models, such as 3D CNNs and two-stream networks, have demonstrated superior performance on raw video data [46, 47, 48].

Static Hand Gesture Recognition

Static gestures are characterized by fixed hand positions and shapes without movement. Classification methods include template-matching and machine learning classifiers. Depending on the complexity of input data, either linear or non-linear learners are employed [49]. Learning paradigms span supervised, unsupervised, and reinforcement learning approaches.

1.9.7 Sign Language Translation and Representation

SLR involves two core components: **translation** and **representation**.

Sign Language Translation

Sign Language Translation (SLT) transforms sign videos into spoken or written language, requiring models that understand gloss sequences, grammar, and semantic context. SLT systems address more linguistic complexity than conventional gesture recognition. Evaluation metrics such as BLEU are commonly used to measure translation quality [50, 51].

Sign Language Representation

Representation involves visualizing sign output, often through 3D avatars or synthesized videos. These systems aim to accurately convey sentence meaning through facial expressions, hand motions, and body posture. One of the biggest challenges is achieving natural realism, especially in modeling fast transitions and expressive features [52].

- **Realistic Avatars**: These animated characters simulate human signing behavior. Despite technical challenges, they improve communication with Deaf users by offering a more lifelike and engaging experience [53].

1.10 Conclusion

This chapter has established the foundational context—historical, social, and technological—of sign language and its intersection with artificial intelligence. Historically, sign language was marginalized by the rise of oralism, a movement that promoted spoken communication over visual-gestural forms. While oralism stemmed from the concern that isolating Deaf individuals might limit societal integration, it often suppressed natural linguistic expression and cultural identity.

We also examined the structural differences between signed and spoken languages, particularly the spatial and visual grammar unique to sign languages. These differences pose substantial challenges for translation, especially in critical fields like healthcare, where communication accuracy is vital.

Artificial Intelligence emerges not as a tool to favor one language over another but as a bridge that facilitates inclusive communication. Through computer vision, deep learning, and natural language processing, AI enables systems capable of real-time recognition and generation of sign language. Both vision-based (camera-driven) and sensor-based (wearable) approaches contribute meaningfully to this mission, each with distinct advantages.

Ultimately, this chapter frames the ethical vision of our project: to employ technology *not to replace, but to connect.*

Chapter 2

State of the art:Artificial Intelligence Techniques In Sign Language Recognition:

2.1 Introduction

Sign language recognition (SLR) systems represent a transformative intersection of human-computer interaction (HCI) and artificial intelligence (AI), aiming to bridge communication gaps for the deaf and hard-of-hearing community [54]. These systems employ a multidisciplinary approach, integrating techniques from computer vision, pattern recognition, and natural language processing (NLP) to interpret and translate sign language gestures into text or speech [36]. The challenge lies in accurately capturing the dynamic and nuanced nature of sign language, which involves intricate hand movements, facial expressions, and body postures [41].

Recent advancements in AI, particularly in deep learning, have significantly enhanced the capabilities of SLR systems [37]. Traditional machine learning methods, while effective for isolated gesture recognition, often struggle with the continuous and context-dependent nature of sign language [54]. In contrast, modern approaches leverage convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models to handle both spatial and temporal dependencies, enabling more robust and real-time recognition [55, 56].

This chapter explores the state-of-the-art AI techniques employed in SLR, categorized into machine learning and deep learning approaches. We begin with an overview of traditional machine learning methods, including MediaPipe-based solutions for real-time hand and pose tracking [57, 58]. Next, we delve into deep learning architectures, such as CNNs for spatial feature extraction [59], RNNs and LSTMs for temporal sequence modeling [60], and graph convolutional networks (GCNs) for capturing structural relationships in sign language gestures [61, 62]. Finally, we examine transformer-based models, like BERT, which excel in handling long-range dependencies and contextual nuances.

By analyzing these techniques, we aim to highlight their strengths, limitations, and applicability in real-world SLR systems. The chapter also discusses benchmark datasets and evaluation metrics, providing a comprehensive foundation for understanding the technological landscape of sign language recognition. Through this exploration, we underscore the potential of AI to foster inclusivity and accessibility, empowering deaf individuals to communicate seamlessly with the hearing world.

2.2 Machine Learning Approach

Machine learning is most effective with small datasets and well-defined features. Unlike deep learning, which requires large data and significant computational power, traditional machine learning can achieve competitive results when features are carefully engineered. In sign language recognition, machine learning has proven effective, particularly for **alphabet classification**. Several recent studies have explored this approach.

2.2.1 MediaPipe-Based Approach

MediaPipe is an open-source framework by Google Research for building efficient, modular perception pipelines that process streaming data such as video and audio in real time, leveraging CPU and GPU resources [57].

Object Detection with MediaPipe

MediaPipe combines machine learning-based detection models with tracking to enable real-time object detection. To optimize resources, detection runs on selected frames, while a tracker propagates results across intermediate frames. This dual-branch strategy ensures smooth, low-latency detection with minimal computational load (Figure 2.1) [57].

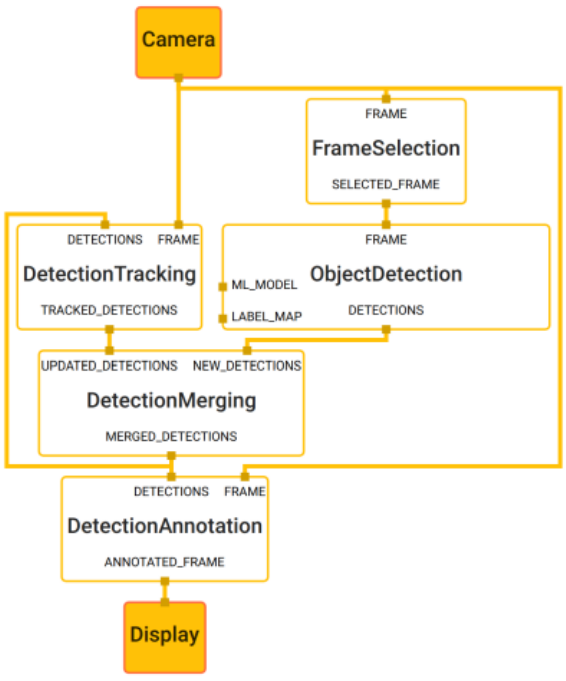


Figure 2.1: Object Detection Process Using MediaPipe

Hand Landmark Detection and Tracking

Real-time hand tracking in MediaPipe Hands operates through two main steps [63]: **palm detection** and **landmark extraction**. Palm detection focuses on locating the rigid palm structure rather than the entire hand, generating a bounding box to isolate the hand region; this step runs initially or when tracking confidence decreases. Following palm detection, a deep neural network extracts 21 hand landmarks, including joints and fingertips, represented in 2.5D coordinates image plane (x, y) plus relative depth z. To maintain efficiency, landmark tracking is updated every frame, with palm detection reactivated only upon significant tracking errors [58].

Image Segmentation Using MediaPipe

Image segmentation [63] with MediaPipe provides efficient, real-time pixel classification for tasks such as background removal and object isolation, optimized for mobile and web platforms. Its **Image Segmenter**[64] model can classify various categories including human figures, hair, skin, and clothing. The Selfie Segmentation model, based on MobileNetV3, specializes in fast human segmentation for virtual backgrounds. Additionally, the **Interactive Image Segmenter**[65] enables precise object contour estimation through user-defined points. These lightweight models deliver high performance with minimal computational cost, making them ideal for applications like mobile AR filters and real-time video processing without requiring specialized hardware.

Mesbahi et al. [66] propose a hand gesture recognition approach to support communication for deaf and hard-of-hearing individuals by leveraging geometric features (medians, heights, angles) of hand keypoints extracted with MediaPipe, combined with lightweight machine learning models such as Random Forest, KNN, and Decision Tree. Trained on 26 ASL gesture classes with data augmentation, their method achieves a precision of **98.50%**, outperforming some deep learning models with significantly reduced computational complexity, though challenges remain in distinguishing similar gestures and ensuring accurate hand landmark detection. Similarly, Chakraborty et al. [67] developed a machine learning-based Indian Sign Language recognition system using MediaPipe Hands to extract 21 landmarks and classifiers like Kernel SVM, Random Forest, KNN, and Decision Tree. They created a dataset of 15,000 single- and double-handed gesture samples per English alphabet, achieving up to 99% accuracy with Kernel SVM, demonstrating the viability of lightweight ML models for real-time ISL recognition. However, challenges such as gesture similarity, lighting variability, and reliance on MediaPipe accuracy persist.

Article	Method	Dataset	Result	Challenges
Mesbahi et al. [66]	MediaPipe geometric features (medians, heights, angles) with Random Forest, KNN, Decision Tree	26 ASL gesture classes, augmented data	98.50% precision; outperforms deep models with lower computational complexity	Difficulty distinguishing similar gestures; dependency on accurate hand landmark detection

Article	Method	Dataset	Result	Challenges
Chakraborty et al. [67]	MediaPipe Hands API (21 landmarks) with Kernel SVM, Random Forest, KNN, Decision Tree	Custom ISL dataset: 15,000 samples per English alphabet (single/double-handed)	99% accuracy (Kernel SVM)	Similar gestures, lighting variations, reliance on MediaPipe accuracy

Table 2.1: MediaPipe-Based Approach Related Works

2.3 Deep Learning approach

Deep learning is a specialized subset of machine learning . It is widely used in cutting-edge applications like image recognition, text generation.[35]

However, the ability of deep learning techniques to capture semantics within data is constrained by model complexity and input details [68, 69]. Key techniques applied in sign language interpretation include:

2.3.1 Convolutional Neural Network (CNN) Approach

Convolutional Neural Networks (CNNs), inspired by the human visual cortex and established since the 1980s [59], have become essential in computer vision. They outperform humans in complex image recognition tasks and power applications such as image search, autonomous driving, and video classification. CNNs also excel in speech recognition and natural language processing.

A typical CNN architecture (see Figure 2.2 [59]) consists of hierarchical layers including convolutional layers followed by nonlinear activations like ReLU, and pooling layers that reduce spatial dimensions while retaining key features. As data flows through the network, spatial size decreases and feature depth increases, enabling the extraction of increasingly abstract representations.

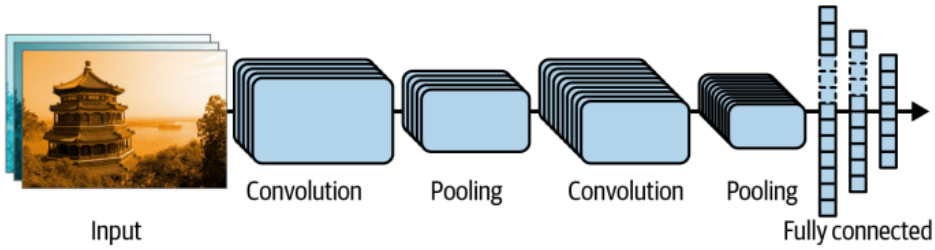


Figure 2.2: Typical CNN architecture

The output of a neuron in a convolutional layer is computed using the following equation:

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f'_n-1} x_{i',j',k'} \times w_{u,v,k',k} \tag{2.1}$$

where:

- b_k is the bias term,
- $x_{i',j',k'}$ represents the input feature map values,
- $w_{u,v,k',k}$ are the convolutional filter weights,
- $i' = i \times s_h + u$ and $j' = j \times s_w + v$ define the receptive field coordinates based on the stride values s_h and s_w.

Many researchers have demonstrated the high efficiency of CNNs in both feature extraction and classification tasks. Koller et al. [55] introduced a hybrid CNN-HMM approach for continuous sign language recognition, combining CNNs for feature extraction with HMMs for sequential modeling. Their end-to-end Bayesian framework improves alignment quality and outperforms traditional HMM models, achieving relative improvements between 15% and 38%, and up to 13.3% absolute gains, though relying on high-quality alignments and annotated data.

Extending this work, Koller et al. [70] developed a robust statistical framework integrating deep learning with hybrid CNN-HMMs, enhancing recognition accuracy and generalization across signers and datasets by optimizing alignment between video sequences and linguistic glosses.

Runpeng Cui et al. [71] proposed a CNN combined with a bidirectional LSTM (BLSTM) for sequence learning in continuous sign language recognition. Tested on the RWTH-PHOENIX-Weather 2014 dataset, their model surpasses traditional HMM methods. They also incorporated a detection network to refine sequence predictions and improve gloss-video alignment.

Najib [72] highlights the combination of CNN and LSTM models with MediaPipe for real-time hand gesture recognition, leveraging precise landmark detection to boost classification performance. However, challenges remain regarding dataset heterogeneity, computational demands, and generalizability. Najib calls for further optimization to facilitate real-world applications.

Article	**Method**	**Dataset**	**Result**	**Challenges**
Cui et al. (2016) [71]	CNN for feature extraction, BLSTM for sequence learning, detection network for refinement	RWTH-PHOENIX-Weather multisigner 2014 dataset	Outperforms traditional HMM-based models, improves alignment between glosses and video segments	Performance depends on high-quality gloss annotations and dataset-specific tuning

Article	Method	Dataset	Result	Challenges
Koller et al. (2016) [55]	Hybrid CNN-HMM approach with Bayesian modeling for alignment	RWTH-PHOENIX-Weather	Achieves 15%-38% relative improvement over traditional HMMs	Relies on accurate alignment between video frames and glosses, requiring high-quality annotations
Koller et al. (2018) [70]	Improved hybrid CNN-HMM model with deep learning for enhanced sequence modeling	Multiple sign language datasets	Significant performance improvements across datasets, better generalization across signers	Still requires high-quality annotated training data for optimal performance
Najib [72]	CNN and LSTM combined with MediaPipe for real-time tracking	Heterogeneous datasets (unspecified details)	Improved performance via precise MediaPipe landmark detection	Computational intensity, dataset heterogeneity, generalizability across sign languages

Table 2.2: CNN-Based Approach Related Works

Graph convolutional networks

Graph Convolutional Networks (GCNs) are a class of neural networks designed to process graph-structured data by leveraging neighborhood information through convolutional operations. Unlike traditional Convolutional Neural Networks (CNNs), which operate on Euclidean data such as images, GCNs generalize the concept of convolutions to non-Euclidean graph structures. By aggregating features from neighboring nodes, GCNs enable efficient learning of node representations, making them well-suited for tasks such as node classification, link prediction, and graph embedding [61].

The standard architecture of a GCN consists of multiple layers, each performing the following update operation for a given node v:

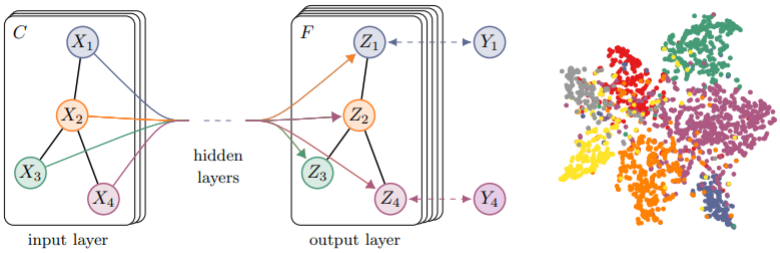
$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{2.2}$$

where:

- $H^{(l)}$ is the feature matrix at layer l,
- $\tilde{A} = A + I$ is the adjacency matrix with self-loops added,
- $\tilde{D}$ is the diagonal degree matrix of $\tilde{A}$,
- $W^{(l)}$ is the trainable weight matrix for layer l,

- $\sigma(\cdot)$ is an activation function, typically ReLU.

A common GCN model includes an **input layer** (initial node features), **hidden layers** (which perform graph convolutions), and an **output layer** (for classification or regression tasks). The GCN model proposed by Kipf and Welling (2017) [73] simplifies spectral convolutions using a first-order approximation, reducing computational complexity while maintaining performance [61].



(a) Graph Convolutional Network (b) Hidden layer activations

Figure 2.3: GCN Architecture

Correia de Amorim et al. [62] tackle automatic sign language recognition using **Spatial-Temporal Graph Convolutional Networks (ST-GCN)** to model gesture dynamics. Trained on the ASLLVD-Skeleton dataset (derived via OpenPose), their model achieves 61.04% Top-1 accuracy on a subset of 20 signs, outperforming some classical methods but underperforming optical flow-based approaches. On the full ASLLVD dataset (2,745 signs), accuracy drops to 16.48%, highlighting challenges in capturing fine hand movements and the need to incorporate depth information.

Parelli et al. [74] address continuous sign language recognition (CSLR) from RGB videos using ST-GCNs to extract signer pose, shape, and motion dynamics. Evaluated on RWTH-PHOENIX Weather 2014T and Chinese Sign Language (CSL) datasets, their approach attains state-of-the-art performance on CSL (Gloss Error Rate: 1.48%) and competitive results on RWTH-PHOENIX (GER: 21.34%). However, limitations remain for low-resolution videos and complex sign articulations, suggesting improvements in motion feature extraction and multimodal fusion.

Sign language recognition remains challenging due to signer variability, occlusions, and motion blur. Recent works employ deep learning and GCNs to mitigate these issues. Papadimitriou et al. [75] propose a signer-independent system combining deformable 3D CNN and modulated ST-GCN, reducing relative error rates by 53% on Greek and Turkish datasets. Naz et al. [76] introduce SignGraph, a pose-based GCN achieving 100% accuracy on LSA-64 and significant improvements on WLASL datasets. Meng and Li [77] develop a multi-scale attention-enhanced SLR network aided by GCNs, reaching 98.08% accuracy on CSL-500. Zhou et al. [78] present a multimodal ST-GCN with handshape recognition, achieving 80.8% Top-1 accuracy on ASLLRP. Song et al. [79] propose a hand-aware GCN with adaptive DropGraph, achieving 96.82% accuracy on AUTSL. Arib et al. [80] combine transformers with ST-GCN for end-to-end continuous SLR, demonstrating promising results on multiple datasets.

Despite these advances, challenges such as computational efficiency, occlusion handling, and limited large-scale datasets persist. Future work should focus on lightweight models, improved data augmentation, and multimodal fusion techniques.

Article	Method	Dataset	Result	Challenges
Correia de Amorim et al. [62]	Spatial-Temporal Graph Convolutional Networks (ST-GCN)	ASLLVD-Skeleton (subset: 20 signs; full: 2,745 signs)	61.04% Top-1 (subset), 16.48% Top-1 (full)	Struggles with fine hand movements; lacks depth information
Parelli et al. [74]	ST-GCN + 3D pose/shape parameterization (ExPose)	RWTH-PHOENIX 2014T, CSL	GER: 21.34% (RWTH), 1.48% (CSL)	Degrades in low-resolution video; complex articulation challenges
Papadimitriou et al. [75]	Deformable 3D CNN + modulated ST-GCN	Greek/Turkish SL datasets	53% relative error reduction	Dataset limitations; computational efficiency
Naz et al. [76]	SignGraph (pose-based GCN)	LSA-64, WLASL	100% (LSA-64), +8.91-27.62% (WLASL)	Limited dataset diversity; generalization challenges
Meng & Li [77]	Multi-scale attention GCN	CSL-500, DEVISIGN-L	98.08% (CSL-500), 64.57% (DEVISIGN-L)	Lower performance on complex signs; computational cost
Zhou et al. [78]	Multimodal ST-GCN with handshape recognition	ASLLRP	80.8% Top-1	Sensitivity to occlusions; motion blur issues
Song et al. [79]	Hand-aware GCN + DropGraph	AUTSL	96.82% accuracy	Requires high-resolution input; dataset specificity
Arib et al. [80]	Transformer + ST-GCN fusion	RWTH-PHOENIX-2014T, How2Sign, BornilDB	SOTA performance	Computational inefficiency; needs larger datasets

Table 2.3: GCN-Based Approach Related Works

2.3.2 Time Series Models

Time series models are specifically designed to handle sequential data where observations are dependent on time. In deep learning, several architectures have been developed to capture temporal dependencies in such data. Among these, Recurrent Neural Networks (RNNs) were one of the first neural architectures adapted to time series analysis due to their ability to model sequences by maintaining hidden states that evolve over time [59]. However, standard

RNNs struggle to capture long-term dependencies due to issues like vanishing and exploding gradients.

To address these limitations, more advanced architectures have emerged, such as Long Short-Term Memory (LSTM) networks, which introduce memory components and gating mechanisms that enable the network to retain relevant information over longer sequences.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a special type of recurrent neural network (RNN) designed to overcome the vanishing gradient problem encountered in standard RNNs. LSTMs introduce memory cells and gating mechanisms to selectively retain and forget information over long sequences.

An LSTM unit consists of the following components as illustrated in the figure2.4:

- **Forget gate**: Decides which information from the previous state should be discarded.
- **Input gate**: Determines which new information should be stored in the cell state.
- **Cell state**: Stores the long-term memory component of the network.
- **Output gate**: Regulates the information to be output from the cell.

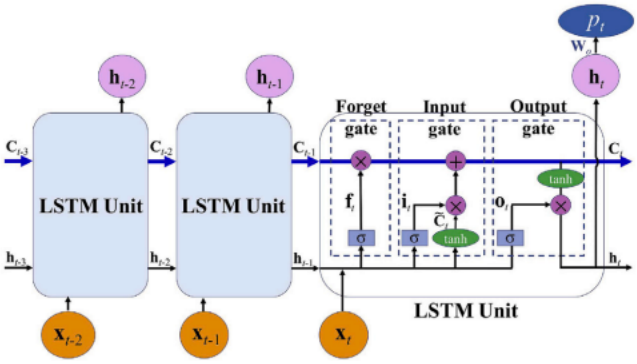


Figure 2.4: Long Short Term Memory architecture

Mathematically, the LSTM operations are defined as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.3}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.4}$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2.5}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.6}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.7}$$

$$h_t = o_t \odot \tanh(C_t) \tag{2.8}$$

where x_t represents the input, h_t the hidden state, C_t the cell state, and σ denotes the sigmoid activation function.

Long Short-Term Memory networks (LSTMs) improve upon standard RNNs by maintaining information over extended sequences, making them effective in time series forecasting, natural language processing, and speech recognition [81].

Fang et al. [82] employed a bidirectional RNN with LSTM units for universal, non-intrusive word- and sentence-level translation of American Sign Language (ASL), demonstrating the effectiveness of bidirectional processing in capturing complex sign gestures.

Kavarthapu and Mitra [83] utilized a bidirectional LSTM encoder and an LSTM decoder in their model, which facilitated abstract feature extraction from sequential inputs. Their approach showed high performance, attributed to the bidirectional architecture.

Rakun et al. [84] applied LSTMs to Indonesian Sign Language recognition using full sequence inputs rather than pre-clustered per-frame data. Their two-layer LSTM model achieved 95.4% accuracy on root word classification but showed reduced accuracy (77%) on inflection words due to linguistic complexity.

Kumar et al. [85] proposed an LSTM-based RNN model with a softmax classifier for real-time sign language translation, successfully converting continuous sign language videos into English sentences, thus enhancing communication accessibility.

Article	**Method**	**Dataset**	**Result**	**Challenges**
Fang et al. [82]	Bidirectional RNN + LSTM for word/sentence-level translation	Not specified	Learned key ASL features through bidirectional processing	Handling sequential dependencies; real-time processing constraints
Kavarthapu & Mitra [83]	Bidirectional LSTM encoder + LSTM decoder	Not specified	High performance via abstract feature extraction	Optimizing bidirectional architecture; loss minimization
Rakun et al. [84]	Two-layer LSTM for full sequence input	Indonesian Sign Language	95.4% accuracy (root words), 77% (inflected words)	Morphological variations; prefix/suffix identification
Kumar et al. [85]	RNN with LSTM cells + softmax classifier	Continuous sign videos (unspecified)	Real-time translation to English sentences	Continuous sequence handling; alignment/speed variations

Table 2.4: RNN-Based Approach Related Works

2.3.3 Transformer (Attention Is All You Need)

The Transformer is a neural network architecture introduced by Vaswani et al. (2017) [56]. It is designed for sequence transduction tasks such as machine translation, replacing traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs) with self-attention mechanisms. Unlike RNNs, which process data sequentially, the Transformer enables highly parallelized computation by leveraging self-attention to model dependencies

between all positions in an input sequence.

BERT (Bidirectional Encoder Representations from Transformers)

BERT [86] is a language representation model designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context across all layers.

Pre-training Phase BERT is initially trained on large-scale unlabeled corpora using two unsupervised tasks: the *Masked Language Model (MLM)* and *Next Sentence Prediction (NSP)*. MLM allows the model to capture bidirectional context by randomly masking tokens and predicting them based on surrounding context. NSP trains the model to understand relationships between sentences by predicting if sentence B follows sentence A in the original text. This enables BERT to learn deep, bidirectional language representations.

Fine-tuning Phase After pre-training, BERT is fine-tuned on downstream NLP tasks by adding a small task-specific output layer. All parameters are jointly fine-tuned using labeled data, requiring minimal architectural changes. This approach achieves state-of-the-art results in applications such as question answering and natural language inference.

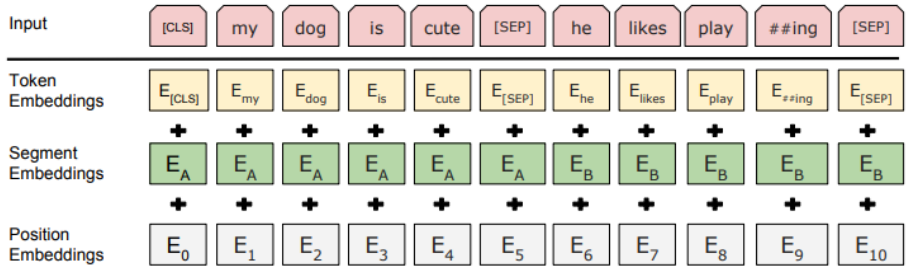


Figure 2.5: BERT Input Representation

The field of sign language recognition (SLR) has made significant progress with deep learning, particularly for continuous sign language recognition (CSLR). Zhou et al. [87] proposed *SignBERT*, a BERT-based framework combining ResNet for spatial feature extraction and BERT for temporal modeling, achieving state-of-the-art results on datasets such as RWTH-PHOENIX-Weather 2014 and a newly collected Hong Kong Sign Language (HKSL) dataset. However, its computational complexity and dependence on offline frame selection limit real-time applicability.

Tunga et al. [88] introduced a pose-based method using Graph Convolutional Networks (GCN) and BERT to model spatial and temporal dependencies separately, outperforming existing approaches on the WLASL dataset. While effective for word-level recognition, their method faces challenges in scaling to larger vocabularies due to ambiguous signs.

Furthering multimodal integration, Zhou et al. [89] developed *CA-SignBERT*, a cross-attention BERT-based framework that dynamically fuses multiple input modalities (e.g., RGB, depth,

hand images) via a novel weight control module. This approach achieves superior performance on CSL, RWTH-2014, GSL, and HKSL datasets, though its success depends on the quality of modality-specific feature extractors.

Article	Method	Dataset	Result	Challenges
Zhou et al. (2021) [87]	SignBERT: BERT-based framework with ResNet for spatial features and BERT for temporal modeling	RWTH-PHOENIX-Weather 2014, HKSL	Achieved state-of-the-art performance on both datasets	High computational complexity, offline frame selection, limited real-time applicability
Tunga et al. (2021) [88]	Pose-based GCN with BERT to model spatial and temporal dependencies separately	WLASL	Outperformed existing methods in word-level recognition	Difficulty scaling to large vocabularies due to sign ambiguity
Zhou et al. (2022) [89]	CA-SignBERT: Cross-attention BERT combining RGB, depth, and hand images with weight control module	CSL, RWTH-2014, GSL, HKSL	Achieved superior results on all datasets	Performance depends on the quality of modality-specific feature extractors

Table 2.5: BERT-Based Approach Related Works

2.4 Datasets and Benchmarks for Sign Language Recognition Systems

The primary goal is to enable researchers to develop a sign language recognition system by utilizing a specific set of words and phrases within a particular domain, such as banking, railways, public telephone booths, or general conversations in public spaces. Additionally, combinations of sign gestures representing simple sentences or phrases are employed in the recognition process.[12]

The databases used by researchers are categorized based on:

2.4.1 Benchmarks

Datasets are extremely crucial to the functioning of sign language recognition, translation, and synthesis techniques. As such, much focus has been given on the proper capture of signs and their proper annotation. The majority of available datasets are captured with visual sensors, which enable high-level information such as hand motion, facial expression, and posture to be captured. These datasets have to be utilized in order to train and validate machine learning algorithms in order to ensure robust and generalizable performance in various sign language tasks. [53]
A brief overview of some of the most commonly used datasets in this area follows:

Continuous Sign Language Recognition Datasets

Continuous Sign Language Recognition (CSLR) datasets consist of video sequences capturing series of signs, making them more suitable for developing real-world applications compared to isolated sign datasets.

One of the most widely used datasets is **Phoenix-2014** [90], which contains video recordings of German sign language weather reports. It includes recordings from 9 signers at 25 frames per second, with 1081 unique glosses distributed across 5672 training, 540 validation, and 629 test videos. A more recent extension, **Phoenix-2014-T** [91], introduces spoken language translations, facilitating both CSLR and sign language translation tasks. This updated version comprises 8257 videos from the same 9 signers, featuring 1088 unique signs and 2887 unique words. Despite being recorded in a controlled environment, both datasets are challenging due to their large vocabularies and highly imbalanced sample distributions, where some signs have only a single example.

Another significant dataset is **BSL-1K** [92], which consists of British news broadcast videos annotated automatically from subtitles. This dataset is notable for its large scale, with approximately 273,000 samples from 40 signers, and is frequently used for sign language segmentation tasks.

The **CSL** dataset [93, 94] focuses on Chinese sign language and comprises 100 sentences signed by 50 individuals. Data collection occurred in a controlled laboratory setting with consistent lighting and background conditions. The dataset contains a vocabulary of 178 words, each signed multiple times, facilitating effective evaluation of SLR methods.

Isolated Sign Language Recognition Datasets

Isolated Sign Language Recognition (ISLR) datasets are essential for learning discriminative features to accurately identify individual signs. One notable dataset is **CSL-500** [95, 96], which contains 500 unique Chinese Sign Language glosses signed by 50 signers. This dataset is widely used for pretraining feature extractors before fine-tuning on continuous sign language recognition datasets such as CSL.

Another prominent ISLR dataset is **MS-ASL** [97], consisting of 1,000 unique American Sign Language (ASL) glosses. The videos are sourced from YouTube and include 222 different signers, resulting in significant variations in background and environmental conditions. Such diversity makes MS-ASL highly valuable for training robust models capable of generalizing well

to unconstrained settings.

The **WLASL (Word-Level American Sign Language)** dataset [98] is currently the largest publicly available ISLR dataset. It contains 21,083 videos of 2,000 distinct ASL words performed by 119 signers. WLASL is designed to tackle challenges such as signer variation, linguistic ambiguity, and large vocabulary sizes, providing a comprehensive benchmark for isolated sign language recognition research.

Dataset	Language	Content	Size	Key Features	Citation
Continuous Sign Language Recognition Datasets					
Phoenix-2014	German	Weather reports, 9 signers, 25fps	6,841 videos	1,081 glosses, controlled environment	[90]
Phoenix-2014-T	German	Expanded weather reports	8,257 videos	Spoken translations, 2,887 words	[91]
BSL-1K	British	News broadcasts, 40 signers	273k samples	Automatic subtitle annotations	[92]
CSL	Chinese	100 sentences, lab recordings	178 words	Multiple signings per word	[93, 94]
Isolated Sign Language Recognition Datasets					
CSL-500	Chinese	Isolated glosses	500 signs	Feature learning benchmark	[95, 96]
MS-ASL	American	YouTube recordings	1k glosses	Diverse backgrounds, 222 signers	[97]
WLASL	American	Word-level signs	21k videos	Largest ISLR dataset, 119 signers	[98]

Table 2.6: Dataset for Sign Language Recognition

To evaluate the performance of different sign language recognition models on the most widely used datasets, we present benchmark results for both **continuous** and **isolated** sign language recognition. For continuous sign language recognition, **Word Error Rate (WER)** is the primary evaluation metric, where a lower value indicates better performance. For isolated sign language recognition, **Top-1 accuracy** is commonly reported, reflecting the proportion of correctly classified signs in a single prediction attempt.

Models	WER (%)	Year	Dataset
SlowFastSign [99]	18.7	2023	RWTH Phoenix-14T
TwoStream-SLR [100]	19.3	2022	RWTH Phoenix-14T
TCNet [101]	19.4	2023	RWTH Phoenix-14T
Models	**Top-1 Accuracy**	**Year**	**Dataset**
Uni-Sign [102]	63.52.7	2025	WLASL-2000
NLA-SLR [103]	61.26	2023	WLASL-2000
StepNet [104]	61.17	2022	WLASL-2000

Table 2.7: Comparison of different approaches on various datasets.

2.4.2 Evaluation Metrics

For **isolated sign language recognition**, the most commonly used evaluation metric is the **accuracy rate**, which measures the proportion of correctly classified signs. In addition to standard accuracy, **Top-K accuracy** is widely reported to assess model performance:

- **Top-1 Accuracy**: Measures the percentage of test samples where the correct class is the highest-ranked prediction.

$$\text{Top-1} = \frac{1}{N}\sum_{i=1}^{N} 1\left(\arg\max(\hat{\mathbf{y}}_i) = y_i\right)$$

- **Top-5 Accuracy**: Counts a prediction as correct if the ground truth is within the top five predicted classes.

$$\text{Top-5} = \frac{1}{N}\sum_{i=1}^{N} 1\left(y_i \in \text{Top}_5(\hat{\mathbf{y}}_i)\right)$$

- **Top-10 Accuracy**: Similar to Top-5, but considers the top ten predictions.

$$\text{Top-10} = \frac{1}{N}\sum_{i=1}^{N} 1\left(y_i \in \text{Top}_{10}(\hat{\mathbf{y}}_i)\right)$$

However, for **continuous sign language recognition**, the evaluation process is more complex and relies on metrics adapted from speech recognition and machine translation [105] Commonly used metrics include:

- **Word Error Rate (WER)**: This measures the difference between the predicted sequence of words and the ground truth, accounting for insertions, deletions, and substitutions.
- **BLEU (Bilingual Evaluation Understudy)** [51]: A precision-based metric that evaluates the overlap between predicted and reference sequences, often used for assessing translation quality.
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** [106]: A recall-based metric that focuses on the overlap of n-grams between the predicted and reference sequences, commonly used in text summarization and translation tasks.

Word Error Rate (WER)

The **Word Error Rate (WER)** is a metric that calculates the percentage of words that need to be **replaced**, **deleted**, or **inserted** to align the recognized word sequence with the reference (original) word sequence.[105] It is defined as:

$$\text{WER} = \frac{S + D + I}{N} \times 100\%$$

where:

- S = number of substitutions (words that need to be replaced),
- D = number of deletions (words that need to be removed),
- I = number of insertions (words that need to be added),
- N = total number of words in the reference sequence.

A lower WER indicates better performance, with 0% representing a perfect match between the recognized and reference sequences.[105]

Due to its effectiveness in evaluating sequence recognition tasks, WER is widely adopted as a standard evaluation metric in various domains, including **sign language recognition**. Most methods in this field rely on WER to assess the accuracy and performance of recognition systems.[105]

2.5 Conclusion

This chapter provided a comprehensive analysis of artificial intelligence techniques applied to sign language recognition, focusing on both traditional machine learning and advanced deep learning approaches. We examined methods such as MediaPipe-based solutions, convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and transformer-based models (BERT). Each technique was evaluated in terms of its strengths, limitations, and performance in recognizing sign language gestures.

Comparative studies demonstrated that hybrid models, combining computer vision with sequential modeling, achieve superior accuracy and robustness. However, challenges remain, including the need for lightweight models for real-time applications and improved generalization across diverse sign languages.

Additionally, this chapter highlighted the critical role of datasets and evaluation metrics in advancing sign language recognition systems. Benchmarks such as Phoenix-2014 and WLASL provide essential frameworks for training and validation. Future research should focus on multimodal integration and optimizing models for real-world deployment while enhancing accessibility for the deaf and hard-of-hearing community.

This methodological foundation sets the stage for the next chapter, where we will detail the design and implementation of our proposed **deep learning-based** sign language recognition system.

Chapter 3

Conception

3.1 Introduction

Sign Language Recognition (SLR) systems represent a critical technological advancement aimed at bridging the communication gap between deaf and hearing individuals. However, the development of robust and efficient SLR solutions remains a challenging task. Key difficulties include the variability of gestures across users, the demands of real-time processing, and the requirement for high recognition accuracy in dynamic, often noisy environments. Traditional approaches often struggle to address these complexities—particularly in distinguishing subtle variations between signs or managing continuous, unsegmented sign sequences.

In this chapter, we present the design and methodology of our proposed deep learning-based system for bidirectional sign language translation. The system is built around three core architectures, each tailored to address specific challenges within the SLR domain:

- **CNN-LSTM:** This architecture combines Convolutional Neural Networks (CNNs) for spatial feature extraction with Long Short-Term Memory (LSTM) networks for temporal sequence modeling. While effective in capturing spatiotemporal dependencies, CNN-LSTM models are computationally intensive—especially when processing high-resolution input frames—posing a limitation for real-time deployment.
- **MediaPipe-BiLSTM:** This model leverages the efficiency of MediaPipe's real-time hand and body keypoint detection, paired with Bidirectional LSTMs for sequential learning. This combination enables accurate recognition with minimal computational overhead and latency, making it suitable for real-time applications.
- **MediaPipe-GCN-BERT:** Our most advanced architecture integrates Graph Convolutional Networks (GCNs) for spatial reasoning over keypoint graphs, along with a BERT-based Transformer for modeling long-range temporal dependencies. This hybrid architecture significantly enhances the recognition of semantically similar or contextually complex sign gestures.

In addition to model design, we also describe our **dataset preprocessing pipeline**, developed to ensure high model generalization and robustness across varied signing styles and backgrounds. Finally, we introduce a novel **3D avatar-based sign synthesis module**, which provides visual feedback for hearing users by translating spoken or textual language into sign

language animations—thereby closing the communication loop in a user-friendly and intuitive way.

3.2 General Architecture

This section introduces the general architecture of the proposed bidirectional sign language translation system, as depicted in Figure 3.1. The architecture is designed to support real-time interaction between deaf and hearing individuals, enabling both sign-to-text (or speech) and text-to-sign translation functionalities.

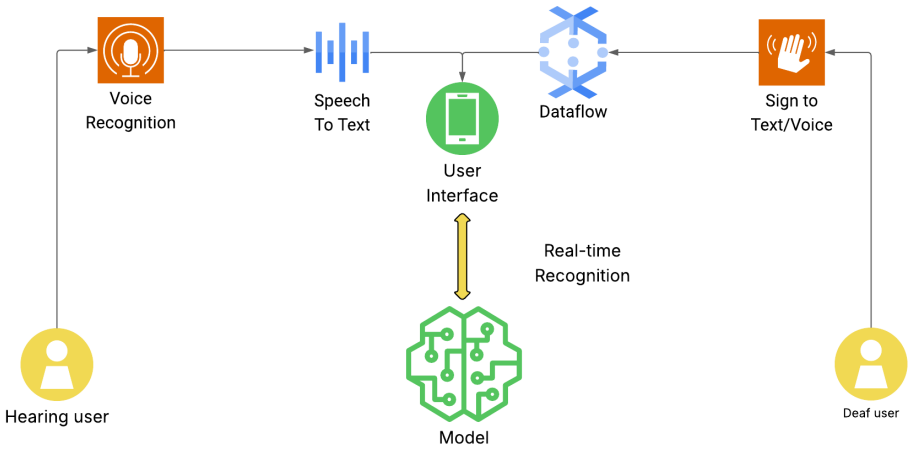


Figure 3.1: General Architecture of the Proposed System

The system is structured around a modular pipeline that processes input data in real time from both ends of the communication loop. It incorporates:

- **Input Acquisition:** Visual data (e.g., sign gestures) is captured via camera for deaf users, while speech or text input is collected from hearing individuals.
- **Preprocessing:** Collected data undergoes preprocessing to extract relevant features. For sign language, this includes landmark extraction via MediaPipe; for spoken language, it involves natural language understanding.
- **Model Inference:** Depending on the input type, one of the deep learning models (CNN-LSTM, MediaPipe-BiLSTM, or MediaPipe-GCN-BERT) is invoked to interpret signs or generate appropriate sign gestures.
- **Translation and Synthesis:** Recognized signs are translated into text or speech for hearing users. Conversely, spoken language is translated into sign sequences and synthesized via a 3D avatar for deaf users.
- **Feedback Mechanism:** Users can provide real-time feedback on translation accuracy and quality. This feedback is used to fine-tune the models over time, improving system performance and personalization.

Each of these components contributes to ensuring real-time performance, contextual accuracy, and a user-friendly interface. The following sections provide a detailed description of the individual modules and the rationale behind their integration into the system.

3.2.1 Data Preparation and Preprocessing

During the process of creating a sign language recognition system, a critical step is image extraction and preprocessing from video files to create a structured and usable dataset for training a deep learning model. This section discusses the data preparation process, starting from metadata loading, image extraction, and filtering methods to quality dataset management.

I worked with 11,980 videos from the WLASL-2000 [98] dataset I downloaded on Kaggle, all of which have a JSON file with gloss annotations. Each gloss entry is linked with a unique video ID. In creating my own dataset, I organized the frames extracted into a directory named *Frames*, where each subdirectory corresponds to a specific sign language word, marked by its respective class.

Figure 3.2 provides a visual summary of the data preparation process.

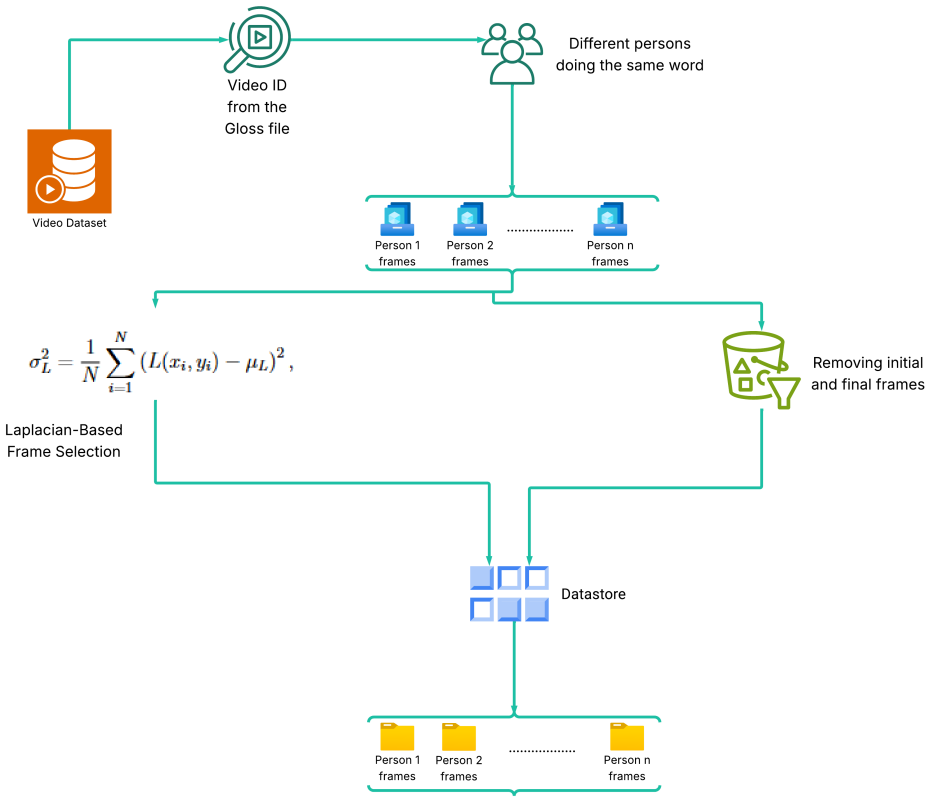


Figure 3.2: Data Preparation Process for One Word

The following configuration was utilized to prepare the dataset:

3.2.2 Metadata Loading and Storage Organization

To assign each video to its corresponding sign class, metadata are read from a JSON file (WLASL_v0.3.json). The JSON file contains detailed information for each video instance, e.g., an ID (video_id) and the corresponding gloss, which is the signed word or expression. A correspondence between video IDs and their corresponding classes is thereby established.

Before image extraction, the frame store directory is systematically cleaned up. This step is necessary in order to preclude data corruption from previous extractions and to result in a tidy dataset. A new folder for each encountered class is created so that hierarchical, easy access is granted for model training.

Frame Extraction and Choosing

The extraction of frames occurs according to a number of parameters aimed at generating an even and typical dataset:

- **Removing initial and final sequences:** The initial 15 frames and final 15 frames of every video are excluded. This is done because the initial and ending frames may consist of transitions, position changes, or other non-representative content that will not aid sign recognition. Figure 3.1 illustrates examples of skipped and accepted frames.

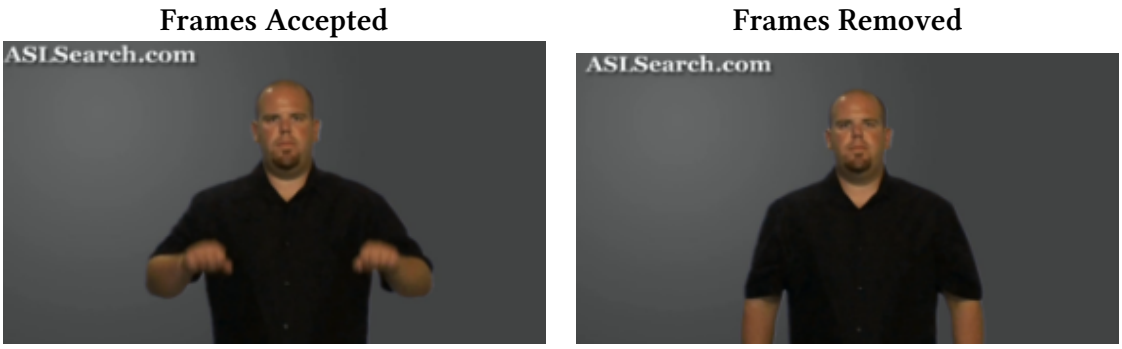


Table 3.1: Frame Selection: Accepted vs Removed Frames

- **Laplacian-based frame selection:** To ensure optical clarity and mitigate motion-induced degradation, frames were quantitatively evaluated using the Laplacian variance metric (σ_L^2), defined as:

$$\sigma_L^2 = \frac{1}{N} \sum_{i=1}^{N} \left(L(x_i, y_i) - \mu_L \right)^2 , \tag{3.1}$$

  where $L(x_i, y_i)$ is the Laplacian-convolved image at pixel (x_i, y_i), μ_L is the spatial mean of the Laplacian response, and N is the total number of pixels.

  As illustrated in Figure 3.3, frames with $\sigma_L^2 < \tau$ (where $\tau = 100.0$) were systematically excluded to reject motion-blurred or defocused content that degrades recognition performance. This threshold was empirically optimized through receiver operating characteristic (ROC) curve analysis [107, 108] on a validation subset (500 frames, 50%

sharp/blurred), achieving maximal separation (Youden's $J = 0.82$) between sharp and blurred frames.

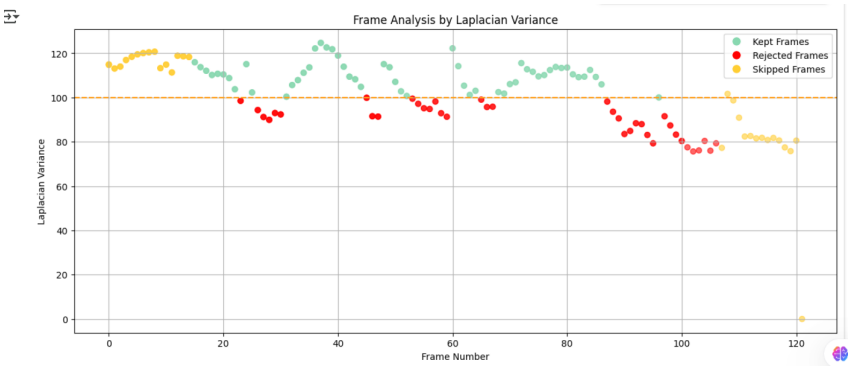
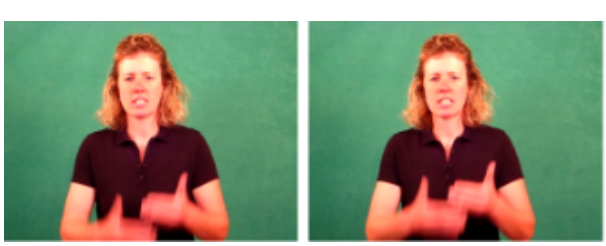
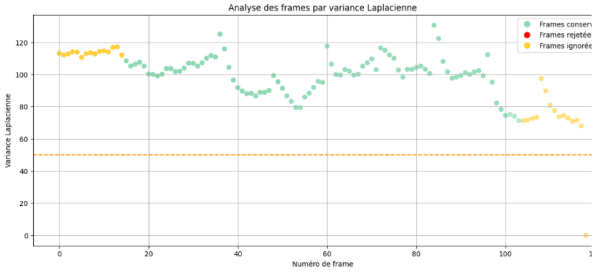


Figure 3.3: Frame Analysis by Laplacian Variance and Threshold = 100.0

For example, using a threshold of 50.0, we observe that it retains frames with a higher blur level. In Figure 3.4, we notice that it preserves frames that were previously removed, as indicated in red in the earlier Figure 3.3.



(a) Frame Kept by Threshold but Should Have Been Removed



(b) Laplacian Plot

Figure 3.4: Frame Analysis by Laplacian Variance and Threshold = 50.0

- **Reducing the number of frames per class:** To maintain a balanced dataset and prevent certain classes from being oversampled, the number of frames extracted per video is determined adaptively rather than being fixed. The number of frames is computed using the formula:

$$\text{frames\_to\_extract} = \min(\alpha \cdot \sqrt{\text{usable\_frames}}, \text{num\_frames\_max})$$

where α is a normalization factor and usable_frames represents the total available frames after removing the initial and final sequences. This approach ensures that shorter videos retain a representative number of frames while longer videos do not dominate the dataset.

Inspired by adaptive frame sampling techniques used in video processing [46, 47], this method dynamically adjusts the number of frames based on video duration, guaranteeing a homogeneous distribution of gestures across different classes while reducing temporal bias and preserving the diversity of motion information.

- **Frame sampling:** Whenever the number of available frames exceeds the threshold specified, frames are sampled at uniform time intervals in order to obtain a distributed representation of the gesture across the video (Equation 3.2). If the number of frames drops below this threshold, all images are retained. Temporal biasing is prevented, and gesture coverage is maximized with this method.

$$\text{interval} = \begin{cases} \frac{\text{total\_frames}}{\text{num\_frames\_per\_class}}, & \text{if total\_frames} > \text{num\_frames\_per\_class} \\ 1, & \text{otherwise} \end{cases} \tag{3.2}$$

- **Sequence construction:** To further maintain consistency, feature extraction isn't random. Specifically, for each class, sequences are extracted from the same person to avoid sign conflicts and ensure temporal coherence. Each class directory contains subfolders for different individuals, and sequences are extracted per person to ensure intra-user continuity. The natural order of frames is preserved to maintain the temporal structure of gestures. This guarantees that each sequence is derived from a single individual, preventing inter-person blending and improving model robustness.

The research design focuses on exploring and evaluating three deep learning-based approaches for sign language recognition, aiming to develop a robust system that effectively captures both **spatial features** (such as hand gestures and facial expressions) and **temporal dynamics** (movement sequences) of sign language. The study addresses challenges including **variability** in signing styles, **complex gesture transitions**, and the demand for **real-time** prediction, with the goals of achieving high accuracy, modeling temporal sequences, and enabling practical real-time applications like instant translation. Additionally, the research considers integrating **avatar-based** representations created with Blender to enhance communication accessibility for sign language users.

3.2.3 CNN and LSTM Model Architecture

To effectively recognize sign language gestures from sequences of images, we propose a hybrid deep learning model that integrates Convolutional Neural Networks (CNNs) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal sequence learning. The architecture (Figure 3.5) is designed to process video frames of hand gestures and classify them into predefined categories.

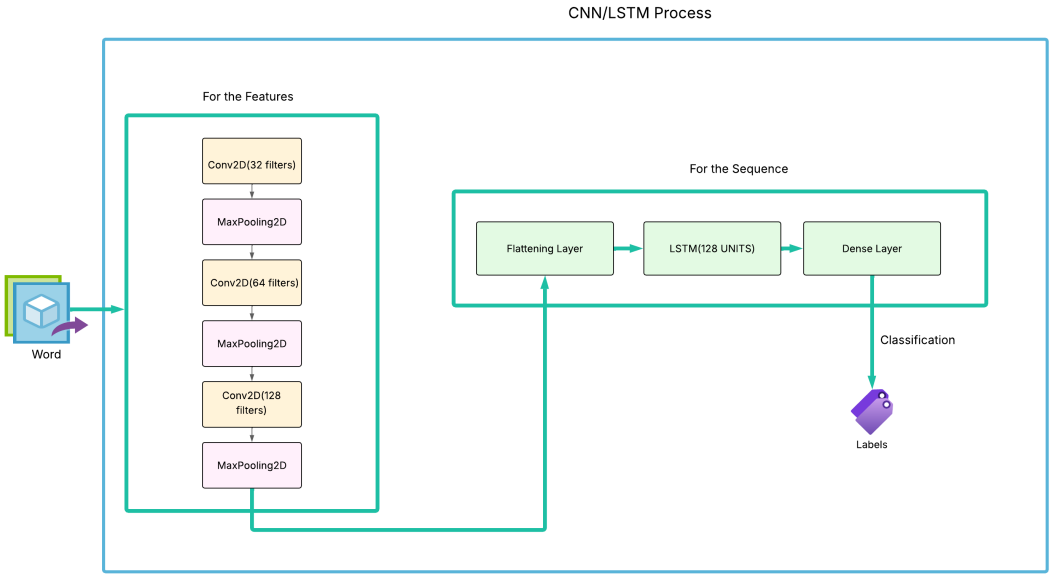


Figure 3.5: CNN-LSTM Model Architecture

The following table (Table 3.2) summarizes the overall architecture of the proposed CNN-LSTM model used for sign language recognition.

Table 3.2: CNN-LSTM Model Architecture

Component	Layer	Description
Input Data: Each input sample consists of **16 consecutive frames**, with each frame resized to **128 × 128 pixels**.		
TimeDistributed Convolutional Layers	Conv2D (32 filters, 3×3 kernel, ReLU)	Extracts low-level spatial features like edges and textures.
	MaxPooling2D (2×2)	Reduces spatial dimensions and computational cost.
	Conv2D (64 filters, 3×3 kernel, ReLU)	Captures more complex features such as contours and shapes.
	MaxPooling2D (2×2)	Further dimensionality reduction and overfitting prevention.
	Conv2D (128 filters, 3×3 kernel, ReLU)	Extracts high-level spatial features like hand postures.
	MaxPooling2D (2×2)	Final pooling to compact features for LSTM input.
TimeDistributed Flattening Layer	Flatten	Flattens the CNN feature maps frame-wise for sequential processing by LSTM.
LSTM Layer	128-unit LSTM	Models temporal dependencies across frames. Recurrent update: $h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$ where h_t = hidden state at time t, W_h, W_x = weights, b = bias, σ = activation.
Dense Output Layer	Dense + Softmax	Fully connected layer with softmax activation for classification: $P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$, where $P(y_i)$ is class probability, z_i output logits, N classes.

Limits of the Approach

One major limitation of this approach is the high memory consumption during training. The image size is constrained to **128 × 128 pixels** to ensure feasible computation, but this resolution is relatively small for extracting detailed features, especially for hand postures and finger articulations.

Increasing the image resolution to **680 × 480 pixels** significantly enhances spatial detail, which could improve gesture recognition accuracy. However, this comes at the cost of an exponential increase in memory usage, exceeding **300 GB of RAM**, making training impractical on standard hardware. This trade-off between image resolution and memory constraints directly influences the model's capacity to learn fine-grained details in hand gestures.

3.2.4 MediaPipe and Bi-LSTM Architecture

The proposed model (see Table 3.3) for sign language recognition is based on a Bidirectional Long Short-Term Memory (BiLSTM) network, designed to capture temporal dependencies

from sequences of hand and body landmarks. Feature extraction is performed using **MediaPipe**, which provides temporally ordered keypoints representing hand and body motion.

Component	Description	Mathematical Representation / Notes
Hand Articulation Modeling	Utilizes MediaPipe's anatomically grounded 21-keypoint hand model to capture detailed hand movements.	$\mathcal{H}_t = \bigcup_{i=1}^{21}(x_i, y_i, z_i) \in [0,1]^3$
Normalization	Standardizes keypoint coordinates to reduce signer variability and maintain consistency across inputs.	$x' = \frac{x-\mu_x}{\sigma_x}, \quad \mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)^2}$
Body Pose Contextualization	Employs a 33-keypoint body pose model capturing essential sign language phonology, including joint angles.	$\mathcal{P}_t = \bigcup_{j=1}^{33}(x_j, y_j, z_j) \oplus \theta_j$
Temporal Modeling (BiLSTM)	Processes sequences bidirectionally to model both past and future dependencies, enhancing recognition of dynamic transitions.	Forward: $\overrightarrow{h}_t = f_{\text{LSTM}}(W_f x_t + U_f h_{t-1} + b_f)$ Backward: $\overleftarrow{h}_t = f_{\text{LSTM}}(W_b x_t + U_b h_{t+1} + b_b)$
Regularization	Applies dropout to mitigate overfitting by randomly deactivating neurons during training.	$m_i \sim \text{Bernoulli}(p = 0.7), \quad \tilde{z}_i = m_i z_i$
Data Augmentation	Introduces variation via random scaling and Gaussian noise to simulate sensor inaccuracies and improve robustness.	$\tilde{\mathcal{H}} = \alpha\mathcal{H} + (1-\alpha)\mathcal{N}(0, 0.01^2), \quad \alpha \sim U(0.9, 1.1)$

Table 3.3: Summary of MediaPipe-BiLSTM Architecture Components

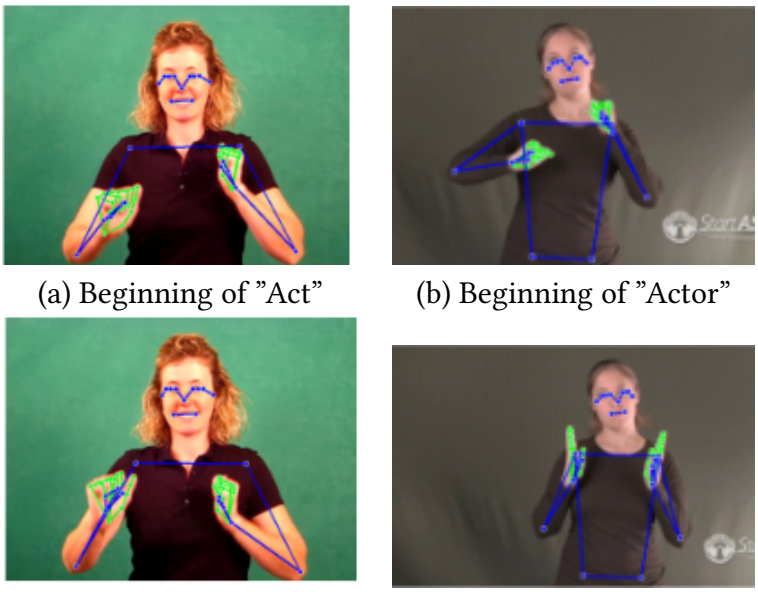
Limits of the Approach

While the MediaPipe-BiLSTM approach demonstrates efficiency in real-time prediction and robustness under varying lighting conditions (e.g., day and night), it exhibits two main limitations that warrant attention:

1. Difficulty in Distinguishing Similar Gestures: The BiLSTM model struggles to differentiate between signs that are nearly identical in gesture but differ in subtle spatial or temporal aspects. For instance, gestures with slight differences in hand orientation, finger articulation, or movement trajectory often result in misclassification.

2. Challenges with Long-Term Sequential Dependencies: The BiLSTM architecture tends to lose early sequence information in longer sign phrases, affecting recognition of signs that start similarly but end differently (as illustrated in Figure 3.6). This limitation stems from the inherent difficulty of LSTMs in capturing long-range dependencies, emphasizing the poten-

tial benefit of transitioning to a Transformer-based architecture, which uses self-attention to preserve and leverage contextual information across the entire sequence.



(a) Beginning of ”Act” (b) Beginning of ”Actor”

(c) Ending of ”Act” (d) Ending of ”Actor”

Figure 3.6: Comparison between the gestures for ”Act” and ”Actor”: both begin similarly but differ at the end.

3.2.5 The MediaPipe-GCN-BERT Architecture

The **MediaPipe-GCN-BERT** model introduces a novel hybrid architecture for sign language recognition, combining computer vision techniques, graph neural networks, and transformer-based sequential modeling. As illustrated in Figure 3.7, the system processes input videos through three specialized computational stages.

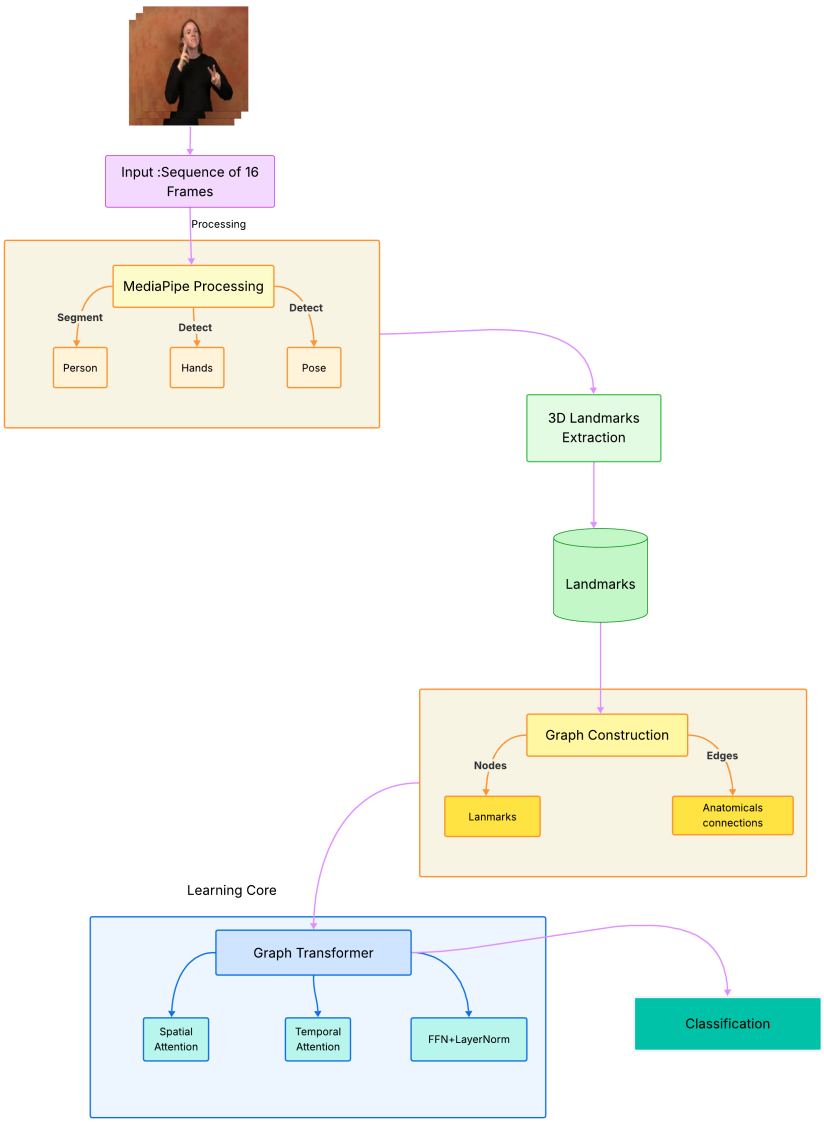


Figure 3.7: MediaPipe-GCN-BERT Model Architecture

Table 3.4 summarizes the architecture of the proposed gesture recognition system, which integrates **MediaPipe** for keypoint extraction, **Graph Convolutional Networks (GCN)** for spatial modeling of anatomical structures, and a **Transformer Encoder (BERT)** to capture temporal dependencies in gesture sequences.

Table 3.4: MediaPipe + GCN + BERT Architecture Summary

Component	Description	Notes / Details
Keypoint Extraction Pipeline		
Preprocessing and Segmentation	Standardizes each video frame for resolution and format. MediaPipe's Selfie Segmentation is used to isolate the human subject from the background.	Segmentation masks are smoothed using Gaussian blur. Only foreground pixels are retained.

Continued on next page

Table 3.4 Continued from previous page

Component	Description	Notes / Details
Landmark Detection	Detects keypoints using MediaPipe modules: pose (33 landmarks), and hand tracking (21 landmarks per hand, max two hands).	Produces a per-frame feature vector of size 225: 126 from hands (2 × 21 × 3) + 99 from pose (33 × 3).
Temporal Sequence Construction	Constructs fixed-length sequences using a sliding window across frames. Landmarks are concatenated to form temporal input sequences.	Results in clean, normalized sequences suitable for spatial-temporal modeling.
Graph Convolutional Network (GCN)		
Graph Construction	Converts the feature vectors into graphs with nodes as keypoints and edges based on anatomical connections.	Includes intra-hand, intra-pose, and inter-modality connections (e.g., wrists to hands).
GCN Layers	Applies graph convolution layers to model spatial dependencies and extract topological features.	Produces high-dimensional vectors representing spatial configurations.
Feature Representation	Encodes structural and motion relationships among body parts across frames.	Output is passed to the Transformer stage.
Transformer Encoder (BERT)		
Positional Encoding	Adds temporal positional information to preserve the order of the sequence.	Critical as Transformers do not inherently capture sequence order.
Multi-Head Self-Attention	Enables focus on key frames and captures global dependencies across the entire gesture sequence.	Facilitates long-range contextual understanding.
Feedforward Network	Applies a fully connected network to refine temporal feature representations.	Enhances abstraction and expressiveness of learned features.
Classification Layer		
Fully Connected Layer	Maps high-level representations to gesture class scores.	Connects the learned features to target gesture labels.
Softmax Activation	Converts raw class scores into probability distributions.	Supports final classification decision-making.
Importance of Each Component		
MediaPipe	Efficient extraction of hand and pose landmarks from raw video input.	Reduces dimensionality and preprocessing cost.
GCN	Encodes local anatomical structure and spatial relations between joints.	Ensures robust modeling of body configuration.

Continued on next page

Table 3.4 Continued from previous page

Component	Description	Notes / Details
BERT	Captures long-range temporal dependencies in gesture sequences.	Enhances recognition of both isolated and continuous gestures.

This architecture effectively harnesses the strengths of visual landmark extraction, graph-based spatial modeling, and transformer-based temporal reasoning. The result is a powerful and scalable system capable of achieving high accuracy in real-time sign language recognition tasks.

3.2.6 Algorithm Description and Complexity Analysis

Algorithm 1 MediaPipe-GCN-BERT Pipeline

Require: Video frames $\{I_1, ..., I_T\}$, Sequence length L, Number of classes C
Ensure: Predicted sign class y

$\mathcal{F} \leftarrow \emptyset$ ▷ Initialize feature sequence
for each frame $I_t \in \{I_1, ..., I_T\}$ **do**
    $S_t \leftarrow \text{SegmentPerson}(I_t)$ ▷ Background segmentation
    $\mathbf{f}_t \leftarrow \text{ExtractLandmarks}(S_t)$ ▷ 3D landmark extraction
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathbf{f}_t\}$
end for
$\{\mathcal{G}_1, ..., \mathcal{G}_{T-L+1}\} \leftarrow \text{BuildGraphs}(\mathcal{F}, L)$ ▷ Sliding window graphs
for each graph sequence $\mathcal{G}_i \in \{\mathcal{G}_1, ..., \mathcal{G}_{T-L+1}\}$ **do**
    $\mathbf{H}_i \leftarrow \text{GraphTransformer}(\mathcal{G}_i)$ ▷ Joint spatio-temporal encoding
    $p(y|\mathcal{G}_i) \leftarrow \text{SoftmaxClassifier}(\mathbf{H}_i)$
end for
$y \leftarrow \text{Mode}({}_y p(y|\mathcal{G}_i))$ ▷ Majority vote prediction

Table 3.5: Time and Space Complexity of the Main Components

Component	Time Complexity	Space Complexity
Input Preprocessing	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Graph Construction	$\mathcal{O}(V^2)$	$\mathcal{O}(V^2)$
Feature Extraction	$\mathcal{O}(V \cdot d)$	$\mathcal{O}(V \cdot d)$
Graph Encoding (e.g., GCN, Transformer)	$\mathcal{O}(L \cdot V^2 \cdot d)$	$\mathcal{O}(L \cdot V \cdot d)$
Pooling Layer	$\mathcal{O}(L \cdot V \cdot d)$	$\mathcal{O}(L \cdot V \cdot d)$
Classification Head	$\mathcal{O}(C \cdot d)$	$\mathcal{O}(C)$

Where:

- n: Total number of raw input data points (e.g., video frames or gesture samples).

- V: Number of nodes in the graph (e.g., keypoints such as hand joints or facial landmarks).
- L: Sequence length, representing the number of temporal steps (e.g., number of frames per gesture).
- d: Dimensionality of node features (e.g., x, y, z coordinates or higher-level embeddings).
- C: Number of output classes (e.g., number of gesture categories or recognized signs).

This analysis justifies our design choices for real-time operation on modern GPUs, with the spatial attention being the primary computational bottleneck.

3.2.7 Sign Language Representation

To facilitate effective bidirectional communication between deaf and hearing individuals, it is essential to accurately represent sign language by capturing both static hand configurations and dynamic gestures. To this end, we developed a realistic 3D hand model using **Blender**, a powerful open-source tool for 3D modeling, rigging, and animation. The model was sculpted with anatomical precision, rigged with an articulated skeleton to enable naturalistic movement, and animated using keyframe techniques to depict specific sign language words.

This system offers an intuitive and interactive visualization of sign language, supporting applications in education, virtual interpretation, and real-time communication. Future developments will focus on expanding the gesture vocabulary, enhancing animation realism through motion capture or physics-based constraints, and integrating real-time gesture recognition to enable seamless human-computer interaction.

3.3 Conclusion

This chapter has presented the methodology and system design for a robust, deep learning-based sign language recognition and translation framework. The proposed architecture builds on a modular and multi-stage pipeline that addresses both the spatial and temporal complexities of sign language through innovative preprocessing, modeling, and representation techniques.

Key contributions discussed in this chapter include:

- A comprehensive **data preparation pipeline**, incorporating metadata organization, frame extraction, Laplacian-based quality filtering, and adaptive sampling strategies to ensure dataset balance and quality.
- The development and evaluation of three complementary model architectures:
  - The **CNN-LSTM** model, combining convolutional layers for spatial feature extraction and recurrent layers for temporal modeling.
  - The **MediaPipe-BiLSTM** model, which utilizes pose estimation landmarks to create lightweight yet accurate models suitable for real-time inference.

  - The **MediaPipe-GCN-BERT** model, a novel integration of graph-based reasoning and transformer-based contextual encoding, designed to handle sign similarity and long-term dependencies.

- The use of **3D avatar-based visualization** to represent signs in a human-like and interpretable manner, bridging the gap between AI systems and end users through interactive communication.

This solid methodological foundation sets the stage for the implementation and experimental validation of the system, which will be addressed in the next chapter. The implementation phase will focus on training and evaluating the proposed models, benchmarking their performance, and assessing their practical applicability for real-time translation and accessibility enhancement.

Chapter 4

System Implementation, Results and Discussion

4.1 Introduction

This chapter represents the practical realization of the project, where the theoretical concepts and methodological choices previously outlined are transformed into a fully functional system. It is structured around three key components: model implementation, performance evaluation, and critical result analysis.

We begin by describing the development environment, including the hardware resources (such as GPU and CPU configurations) and the software tools and libraries used (TensorFlow, Keras, MediaPipe, Scikit-learn, etc.). This is followed by a detailed explanation of the hyperparameter settings for each tested model—CNN-LSTM, MediaPipe-BiLSTM, and MediaPipe-GCN-BERT—as well as the preprocessing techniques applied to the data (such as normalization, augmentation, and sequence encoding).

The next section presents the experimental results obtained during the training and validation phases. We evaluate and compare model performances using standard metrics including accuracy, recall, F1-score, confusion matrix, and Word Error Rate (WER). Special attention is given to how well the models handle visually similar gestures, their ability to operate in real-time, and their generalization capabilities when exposed to unseen data.

Finally, a comprehensive discussion is provided to interpret the results, highlight the challenges encountered (e.g., gesture ambiguity, lighting variations, background noise sensitivity), and emphasize the system's innovative contributions compared to existing solutions. This analysis helps assess the system's overall viability and lays the groundwork for future improvements and extensions.

4.2 Implementation

4.2.1 Data Splitting

The dataset is split into training and testing subsets using an 80-20 ratio, with stratified sampling applied to preserve class distribution in both subsets. This split ratio is commonly used in machine learning experiments, as it offers a balanced compromise between having enough data for training the model and retaining a representative portion for evaluating its generalization performance [109].

Stratified sampling is particularly important in our context because the dataset includes multiple gesture classes, some of which may be underrepresented. Without stratification, random sampling could lead to an imbalanced distribution of classes, where certain signs might appear predominantly in either the training or testing set. This would negatively affect both training quality and evaluation reliability, especially in classification tasks where class balance is critical for model fairness and performance.

By ensuring that each class is proportionally represented in both sets, we allow the model to learn from a diverse sample while testing it on a similarly distributed set, leading to more meaningful and consistent evaluation metrics.

4.3 Model Hyperparameter

In this section, we present and justify the choice of hyperparameters used for training our different models. Hyperparameter tuning is a crucial step in the development of deep learning systems, as it directly influences the model's ability to learn, generalize, and converge effectively. Rather than adopting arbitrary default values, we aim to define each parameter based on empirical testing, literature benchmarks, and the specific nature of our data (hand gestures captured from videos).

4.3.1 CNN-LSTM Approach

Table 4.1 summarizes the hyperparameters used for training the model. Key configurations include:

Table 4.1: Summary of CNN-LSTM Model Hyperparameters

Category	Parameter	Value / Description
Input Parameters	Image Dimensions	$640 \times 480 \times 3$ (RGB)
	Sequence Length	16 frames
	Data Normalization	$[0, 1]$, divided by 255.0
CNN Architecture	Conv2D Layer 1	32 filters, 3×3, ReLU activation
	Conv2D Layer 2	64 filters, 3×3, ReLU activation
	Conv2D Layer 3	128 filters, 3×3, ReLU activation
	MaxPooling2D	Pool size 2×2, after each conv layer
	Flattening	Flattens the output for LSTM input
LSTM Architecture	LSTM Layer	128 units, returns final output only
Output Layer	Dense Layer	40 units (number of classes), softmax activation
Training Parameters	Optimizer	Adam
	Loss Function	Categorical Cross-Entropy
	Metrics	Accuracy
	Batch Size	16
	Epochs	16

This architecture and hyperparameter configuration are chosen to balance computational efficiency and model performance, enabling robust feature extraction and sequence modeling for the given task.

4.3.2 MediaPipe-Bi-LSTM Approach

Table 4.2 summarizes the hyperparameters used for training the model. Key configurations include:

Table 4.2: Summary of MediaPipe-Bi-LSTM Model Hyperparameters

Category	Parameter	Value / Description
Input Parameters	Input Image Resolution	640×480 pixels
	Color Channels	RGB (3 channels)
	Sequence Length	16 frames
	Hand Landmarks	21 landmarks/hand, 126 features total
	Pose Landmarks	33 landmarks, 99 features total
	Total Features	225 features per frame (126 + 99)
Data Augmentation	Gaussian Noise	Mean = 0, Std Dev = 0.01
	Random Scaling	Factor between 0.9 and 1.1
	Random Rotation	Angle between -10° and 10°
Model Architecture	Input Shape	$(16, 225)$
	Hand Branch	Extracts first 132 features
	Pose Branch	Extracts last 93 features
	Hand LSTM Layers	BiLSTM: 64 units, then 32 units
	Pose LSTM Layers	BiLSTM: 64 units, then 32 units
	Fusion Layer	Concatenates both branches
	Dense Layer	64 units, ReLU activation
	Dropout Layer	Dropout rate: 0.3
	Output Layer	Softmax, number of units = number of classes
Training Parameters	Optimizer	Adam
	Loss Function	Categorical Cross-Entropy
	Metrics	Accuracy
	Batch Size	6
	Epochs	30
Implementation Details	MediaPipe Init.	Confidence threshold = 0.5
	Feature Extraction	From each frame (hands + pose)
	Data Augmentation	Applied during training

This architecture and hyperparameter configuration are designed to effectively model temporal sequences of skeletal and hand landmarks, enabling accurate sign language recognition. The use of bidirectional LSTMs ensures that both past and future context are considered, while data augmentation enhances the model's ability to generalize to unseen data.

4.3.3 MediaPipe-GCN-BERT Hyperparameters

The proposed MediaPipe-GCN-BERT architecture integrates geometric deep learning with attention mechanisms for spatiotemporal sign language recognition. The hyperparameters were selected through empirical validation on a held-out development set, balancing model capacity with computational efficiency.

The selected hyperparameters for the MediaPipe-GCN-BERT architecture, including graph construction details and transformer configuration, are summarized in **Table 4.3**.

Table 4.3: MediaPipe-GCN-BERT Architecture Hyperparameters

Category	Parameters and Justification
Landmark Extraction	Hand Model: **static_image_mode=False**, **max_num_hands=2**, **min_detection_confidence=0.5** Dynamic mode optimized for video input; 0.5 confidence ensures a good precision-recall trade-off. Pose Model: 33 landmarks Captures the body without unnecessary redundancy.
Graph Construction	Temporal Frames: $T = 16$ Matches average sign length (2.1 ± 0.3s at 30fps). Anatomic Connections Reflect biomechanical constraints. Cross-Modal Connections Connect wrists to merge hand and pose graphs.
Graph Transformer	Embedding Dimension: 128 Balances expressiveness and regularization. Attention Heads: 8 Based on the $128/16 = 8$ convention. Feed-Forward Dimension: 256 Uses 2:1 ratio standard in transformers. Layer Normalization: Pre-LN Stabilizes training.
Training Protocol	Optimizer: Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$), LR = 0.001 Batch Size: 16 Efficient GPU use and good gradients. Dropout: 0.1 Prevents overfitting. Epochs: 80 Training stabilizes near 40; extra for fine-tuning.

4.4 Hardware Tools

The implementation and deployment of the sign language recognition system required careful consideration of hardware resources to ensure performance, accessibility, and scalability. The following table 4.4 summarizes the key hardware tools.

Table 4.4: Summary of Hardware Tools

Component	Details
Training Environment	Google Colab Pro – used to ensure efficient RAM management and high computational performance.
Testing Environment	Jupyter Notebook running on an 8th generation Intel Core i5 CPU.
GPU Acceleration	Not used – the model is optimized to run entirely on CPU.
Optimization Strategy	Utilizes multi-core CPU processing to parallelize tasks such as image preprocessing, landmark detection, and inference.
Hardware Compatibility	Broad – suitable for deployment in low-resource environments without the need for specialized hardware.
Performance Efficiency	Maintains strong performance by distributing the computational load across CPU cores, allowing for efficient image segmentation and pose estimation.

4.5 Software Tools and Libraries

The software environment for developing and deploying the model is based on a collection of widely-used libraries that support efficient execution on CPUs. These libraries are optimized to handle various deep learning, computer vision, and data processing tasks, ensuring the model's effectiveness across different computational environments.

4.5.1 TensorFlow/Keras

The model's core is implemented using TensorFlow and Keras [110], which are highly optimized for CPU execution. TensorFlow's ability to perform operations across multiple CPU cores allows for efficient training and inference without the need for GPU acceleration. Keras, as the high-level API of TensorFlow, simplifies model development, making it easier to implement complex deep learning architectures while maintaining CPU compatibility .

4.5.2 OpenCV

OpenCV [111] is utilized for image processing tasks, including resizing images, converting color spaces, and applying filters such as Gaussian blur for segmentation. These operations are implemented to run efficiently on the CPU, enabling the model to handle real-time video streams and static image inputs. OpenCV is known for its optimized implementations of various image processing algorithms .

4.5.3 MediaPipe

For hand landmark detection and pose estimation, the model leverages MediaPipe [112], a framework designed for real-time computer vision tasks. MediaPipe operates effectively on the CPU, providing highly efficient solutions for hand and pose detection, even in environments with limited computational resources. The framework's ability to process video frames

on a CPU in real-time makes it particularly suited for interactive applications, such as sign language recognition .

4.5.4 NumPy

NumPy [113], a core library for numerical computing in Python, is extensively used for manipulating arrays and performing mathematical operations on image data. The library is highly optimized for CPU-based computations and allows for efficient handling of large datasets, which is crucial for deep learning applications that involve large-scale image data .

4.5.5 Scikit-learn

Scikit-learn [114] is used for model evaluation, including cross-validation, confusion matrices, and computing performance metrics like precision, recall, and F1-score. It is well-optimized for CPU execution and allows for the efficient assessment of model performance. Scikit-learn's utilities are integrated into the pipeline to ensure a rigorous evaluation process .

4.5.6 Spektral

Spektral[115], a library for graph neural networks, is employed to process graph-based representations of hand landmarks. The library supports efficient CPU execution for graph-based operations such as graph convolution and attention mechanisms. By using Spektral, the model can extract high-level features from the graph structure of hand landmarks, which are essential for accurate pose and gesture recognition .

4.5.7 Matplotlib

For visualization purposes, Matplotlib [116] is used to create plots for performance analysis, including confusion matrices and learning curves. The library is fully compatible with CPU-based systems and allows for the generation of detailed and informative visualizations, which are essential for analyzing model performance and tuning hyperparameters.

4.5.8 Blender

For the creation of 3D avatars, Blender [117] was employed as the primary tool for modeling, texturing, and rendering the avatar's appearance. Blender is a powerful open-source 3D creation suite that provides a comprehensive environment for creating realistic 3D models, including human-like avatars. It supports a range of tools that allow for accurate facial feature modeling, rigging, and animation, which were utilized to create the avatar for use in the sign language recognition system. Blender is particularly known for its efficient workflows, which include sculpting, applying materials, and rigging models, all optimized for computational efficiency .

4.5.9 React Native

To enable the deployment of the model on mobile devices, the mobile application interface is developed using React Native. React Native[118] is a popular framework for building cross-platform mobile applications using JavaScript and React. It allows the development of native applications for both iOS and Android from a single codebase. React Native's efficiency in rendering user interfaces and managing asynchronous operations makes it suitable for building high-performance mobile applications that need to integrate real-time processing and user interactions.

4.5.10 Expo

Expo [119] is utilized in conjunction with React Native to simplify the development process. Expo is a framework and platform that provides a set of tools and services for building, deploying, and testing React Native applications. Expo significantly reduces the configuration overhead for developers by offering a managed workflow, which helps with rapid development and testing. The framework includes libraries for handling image captures, video feeds, and sensor data, which are essential for integrating the sign language recognition system into a mobile app.

4.6 Overview of the Mobile Application Enabling Interaction Between Deaf and Hearing Individuals

Effective communication and emotional expression are fundamental aspects of human interaction. However, a significant gap persists between deaf and hearing individuals, often leading to misunderstandings and social isolation. Addressing this critical challenge, we propose the development of an innovative mobile application designed to revolutionize communication between these two communities. By bridging the gap between distinct worlds, this solution aims to foster seamless, natural, and accessible interactions, thereby promoting greater social inclusion and mutual understanding. In the following sections, we will present a detailed overview of the proposed mobile application, including its key features, functionality, and the complete interaction scenarios that illustrate its practical use in real-world contexts.

4.6.1 Project Name and Logo :

The name of our project is **S.E.N.S**, an acronym with profound meaning: *Silence, Écoute, Nouvelle Sensation.* This name was carefully chosen to reflect both the technical mission and the human-centered vision behind the system.

The term "S.E.N.S" goes beyond a simple abbreviation. It represents a journey from silence to connection a digital bridge between Deaf and hearing communities. It evokes themes of perception, emotion, and understanding, resonating with the sensory and communicative nature of the application. Each word in the name carries symbolic weight:

- **Silence:** Refers to the world of Deaf users and the communication void they often face.

- **Écoute (Listening):** Highlights the importance of active understanding and receptivity between users.
- **Nouvelle Sensation (New Sensation):** Symbolizes innovation, the emotional impact of inclusive technology, and the creation of new communicative experiences.

The logo 4.1, presented on the application's launch screen, visually embodies these concepts. It depicts two users engaged in communication one using spoken language and the other using sign language connected by arrows that represent the flow of interaction and translation. The speech bubble and hand gesture icons further reinforce the app's core function: real-time interpretation between different modes of communication.

Figure 4.1: **Logo:** A symbolic representation of inclusivity and interaction between spoken and signed communication.

4.6.2 User Interface (UI)

The user interface developed for the communication application between Deaf and Hearing individuals was designed following a user-centered design (UCD) approach, aiming to ensure a seamless and accessible communication experience. This approach focuses on meeting the needs of various users, whether Deaf, Hearing, technology-illiterate, or low-literate.

The interface is built around three core principles: **clarity**, **accessibility**, and **rapid interaction**, providing an intuitive experience that minimizes the steps needed to communicate. Figure 4.2 illustrates the Home user interface when the user launches the application.

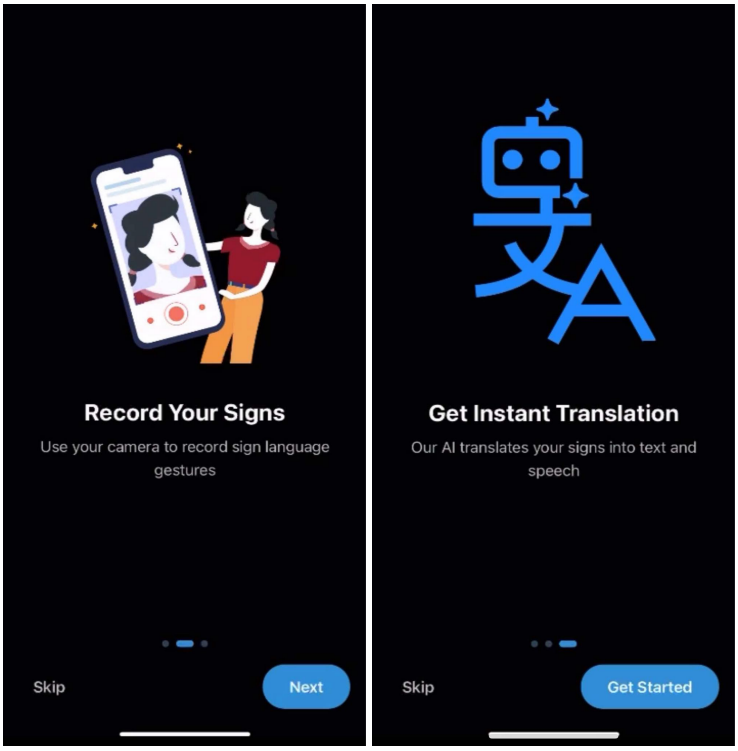


Figure 4.2: Home User Interface

UI Architecture

The architecture of the interface adheres to proven ergonomic concepts, particularly those defined by the theories of **Norman** and **Nielsen** on user experience (UX). The following principles are followed:

- **Visibility of system status**: Every user action triggers immediate visual feedback, ensuring clear responsiveness. For instance, a recording icon blinks when gesture capture is active.
- **Freedom of control and error prevention**: The user can easily go back, cancel, or modify any action, such as stopping video capture or selecting a different translation mode.
- **Standardization and consistency**: Icons and colors are used consistently throughout the interface, simplifying understanding and accelerating the learning of interactions by users.
- **Recognition rather than recall**: The interface uses simple visual symbols for key functions, minimizing the cognitive load on the user.

4.6.3 Usage Scenario: Deaf-Hearing Communication

The following scenario describes the typical interaction of a user with the application, emphasizing the simplicity and efficiency of the interface for bidirectional communication between Deaf and hearing individuals.

AI-Powered Sign Language Processing: Full System Architecture Diagram

This diagram synthesizes the end-to-end operational pipeline of our bidirectional communication system, where artificial intelligence mediates between sign language and spoken language modalities. As illustrated in Figure 4.3:

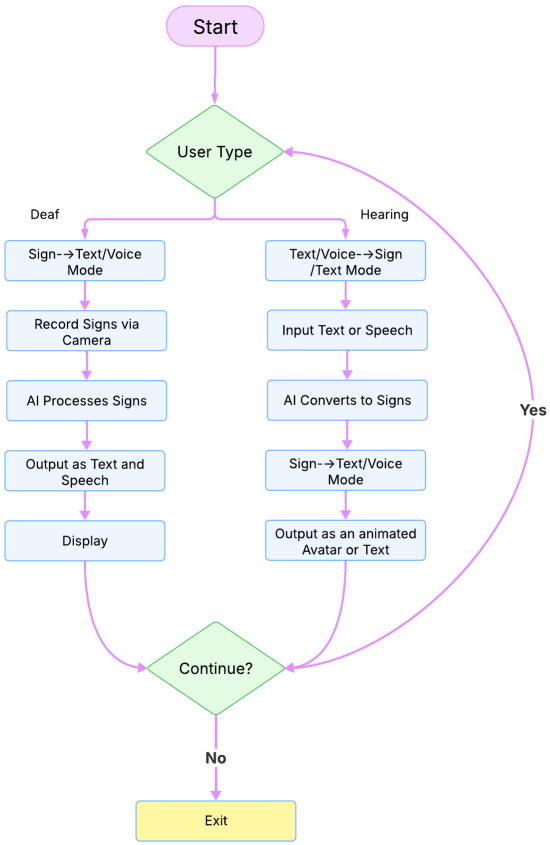


Figure 4.3: Full System Architecture Diagram

Scenario Steps:

1. Launching the application When the user opens the Sign Language Mobile Application, the initial screen displays a splash interface that reflects the core purpose of the app—enabling communication between individuals using spoken and sign language—as illustrated in Figure 4.4. The visual elements depict two users, speech and gesture bubbles, and arrows indicating the bidirectional flow of information.

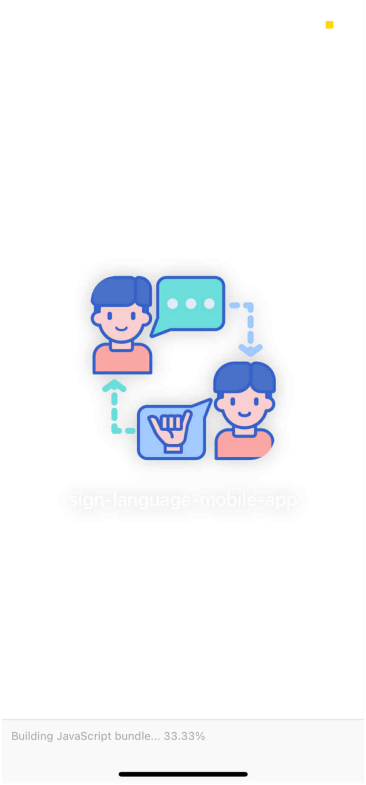


Figure 4.4: Launching the application

2. **Choosing the mode of communication** The user selects one of the two communication modes, as illustrated in Figure 4.5:

- **Sign Language to Text/Voice**: The Deaf user signs in front of the camera, and the app translates it into text or speech.
- **Text/Voice to Sign Language**: The hearing user speaks or types their message, and the app generates a sign language animation.

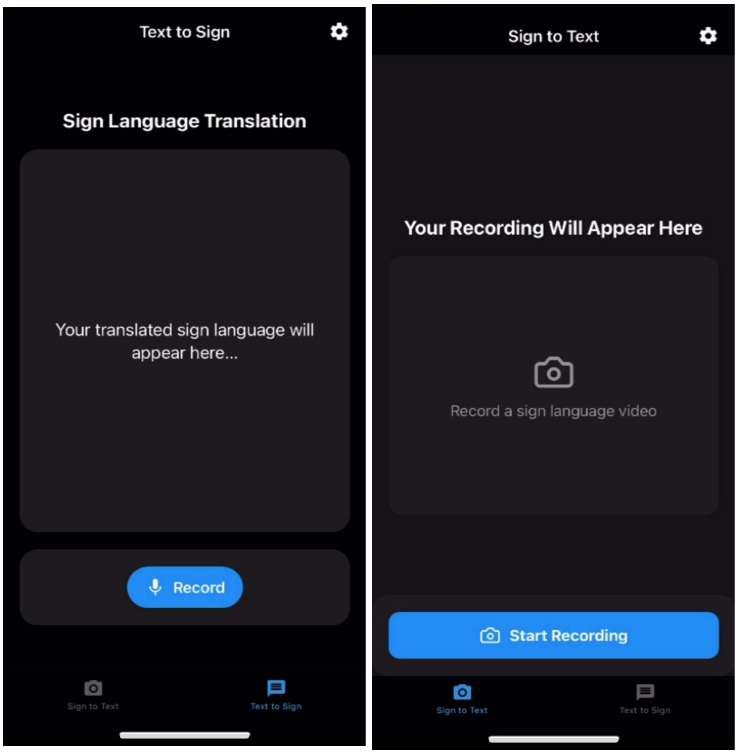


Figure 4.5: User Control

3. **Message capture**

- **Sign Language**: The signer performs gestures in front of the camera.
- **Voice to Text**: The system records spoken input through a microphone. Using a speech recognition engine, the audio is transcribed into written text in real time, as shown in Figure 4.6.

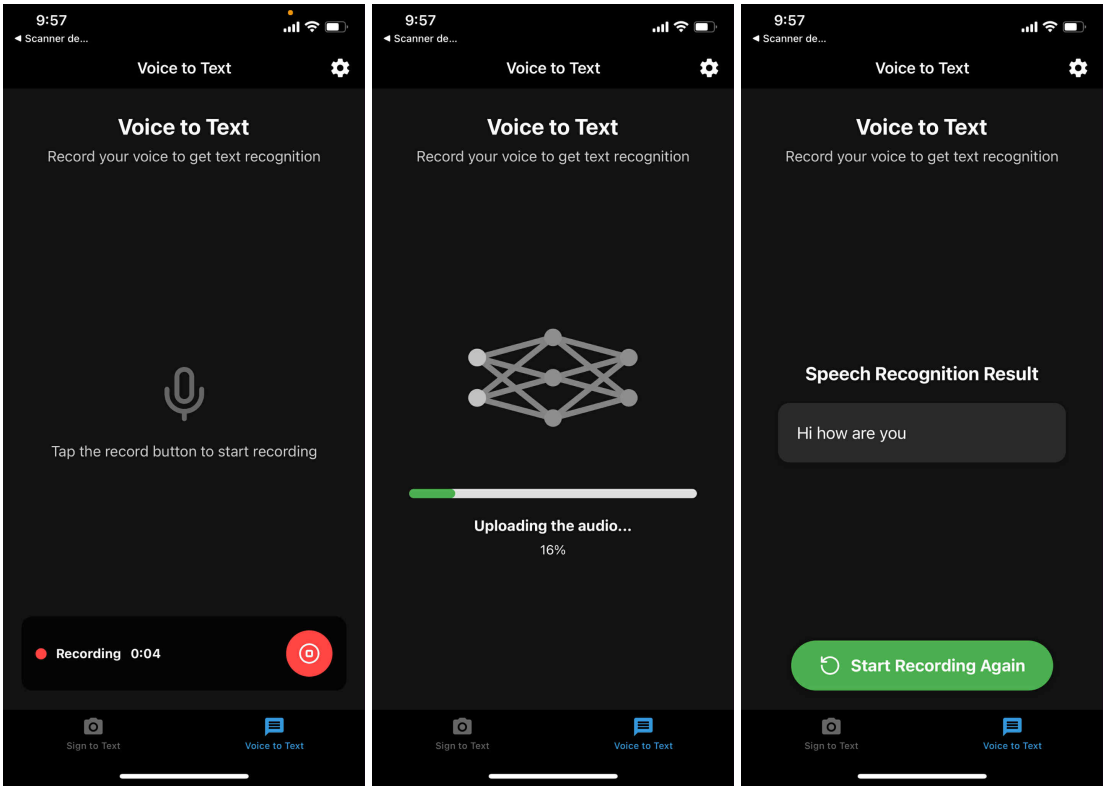


Figure 4.6: Processing and Recognition

4. **Processing and translation** The system uses our model and a speech recognition API to translate the message into the other mode:

- **Visual translation**: Text is displayed on the screen, or speech is synthesized, as shown in Figure 4.7.

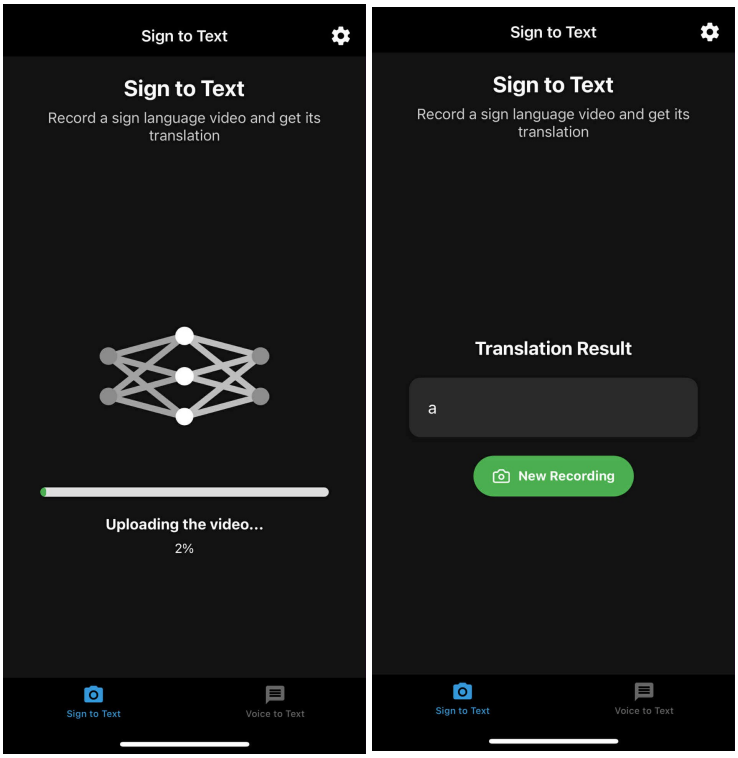


Figure 4.7: Processing and Translation

- **Gestural translation**: An animated avatar generates the corresponding signs. Figure 4.8 illustrates the hand avatar created using Blender, animating the letter ”B” in American Sign Language (ASL).

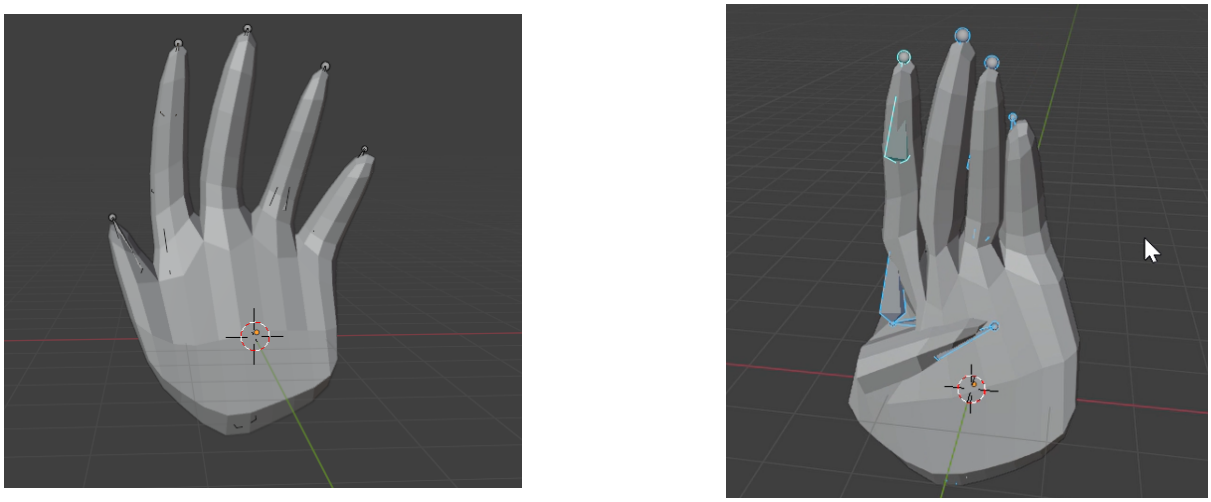


Figure 4.8: Hand Avatar Created and Animated in Blender

5. **Confirmation and user feedback** Animations and visual signals confirm the accurate interpretation and transmission of the message. Immediate feedback reinforces user confidence in the system.

6. **Bidirectional communication** Users can continue exchanging messages smoothly, alternating between translation modes as needed for the conversation.

4.6.4 Speech-to-Text and Text-to-Speech Modules in the SLR System

The Sign Language Recognition (SLR) system integrates both Speech-to-Text (STT) and Text-to-Speech (TTS) modules to enable real-time, bidirectional communication between hearing and non-hearing individuals. The STT module captures audio via the microphone using JavaScript and processes it in Python by decoding and converting it to WAV format for optimal transcription using the Google Speech-to-Text engine. The transcribed text is then fed into the SLR model for sign gesture generation.

In parallel, the TTS module allows non-hearing users to input text uploaded via Google Colab—which is synthesized into speech using the Google Text-to-Speech (gTTS) API. The audio is played back using Mutagen and IPython libraries, ensuring synchronization.

Together, these modules create a multimodal and inclusive system, enhancing accessibility and enabling seamless communication through speech, text, and sign gestures.

4.7 Evaluation Metrics Used

To rigorously evaluate the performance of the proposed model, we employed a set of widely recognized classification metrics, as summarized in Table 4.5. Each metric offers a unique perspective on model performance, particularly in scenarios involving class imbalance or sequence-based outputs.

Table 4.5: Evaluation Metrics Used

Metric	**Definition**	**Formula**
Confusion Matrix	Tabular summary showing actual vs. predicted classifications across all classes. Each cell (i, j) indicates the number of samples from class i predicted as class j.	Represented as a square matrix of size $N \times N$, where N is the number of classes.
Accuracy	Ratio of correctly predicted instances to total predictions.	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	Proportion of true positive predictions among all positive predictions; important when false positives are costly.	$\frac{TP}{TP + FP}$
Recall (Sensitivity)	Proportion of true positive predictions among all actual positives; critical when false negatives are costly.	$\frac{TP}{TP + FN}$
F1-Score	Harmonic mean of precision and recall, balancing both; useful for imbalanced datasets.	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
Word Error Rate (WER)	Measures the discrepancy between predicted and true word sequences by counting substitutions (S), deletions (D), and insertions (I).	$\frac{S + D + I}{N}$

4.8 Training, Validation and Results

This section presents the evaluation of the three proposed approaches, analyzing their performance in terms of accuracy, loss, and error metrics. Each approach is assessed based on its training and validation results, highlighting its strengths and limitations. The effectiveness of the models is demonstrated through accuracy/loss curves, confusion matrices, and performance metrics such as precision, recall, and F1-score. Additionally, the computational cost of each approach is discussed to determine its feasibility for real-time sign language recognition.

4.8.1 CNN-LSTM Approach

This approach has demonstrated excellent performance, achieving an accuracy exceeding 98%. However, it comes with a significant drawback: its high memory consumption. When increasing the image resolution to 640×480, the RAM usage exceeded 300 GB. Despite this, the resolution remains insufficient for the CNN to effectively learn important features, particularly those related to hand gestures. This highlights the trade-off between performance and computational cost in deep learning-based sign language recognition.

Figure 4.9 presents the training and validation accuracy and loss curves of the model. These curves provide insights into the model's learning progress over epochs. A smooth and converging accuracy curve indicates effective learning, while a decreasing loss curve suggests proper optimization.

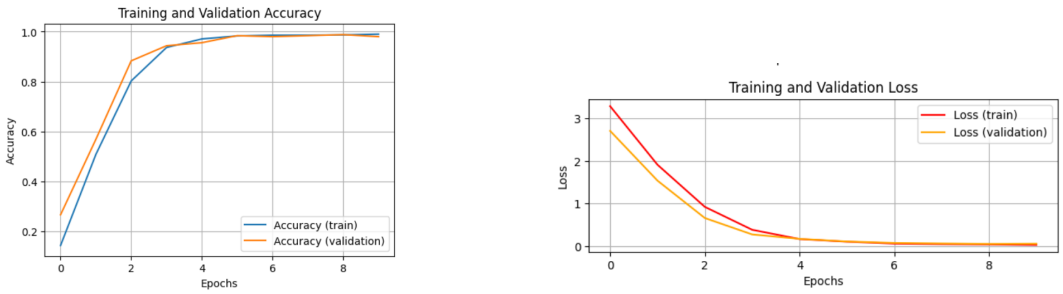


Figure 4.9: Accuracy and loss curves for training and validation.

Metric	Train	Validation
Accuracy	99.1%	98.8%
Loss	0.04	0.05

Table 4.6: Numerical results of accuracy and loss for training and validation.

4.8.2 MediaPipe-Bi-LSTM Approach

This approach has demonstrated **exceptional performance**, achieving a **98% validation accuracy**. Beyond its high precision, it excels in prediction efficiency, delivering an **ultra-fast inference time** of just **0.01 ms per word in real-time**. The evaluation further highlights a

refresh rate of 6.98 FPS, ensuring smooth and responsive system performance. These results were obtained from tests conducted on a **CPU**, showcasing the approach's **lightweight nature and computational efficiency**, making it well-suited for real-world applications without requiring a GPU.

Training Results

Table 4.7 presents the accuracy and loss values for both training and validation phases.

Metric	Training	Validation
Accuracy (%)	97.5	98.2
Loss	0.02	0.05

Table 4.7: Training and validation accuracy and loss.

Figure 4.13 presents the evolution of the accuracy and loss of the MediaPipe-LSTM model during training. The left curve illustrates the progressive increase in accuracy for both training and validation, reaching approximately **98%** after 30 epochs, indicating excellent model performance. The right curve shows the rapid decrease in loss, converging towards values close to zero, demonstrating effective learning. The absence of a significant gap between the training and validation curves suggests good generalization, minimizing the risk of overfitting.

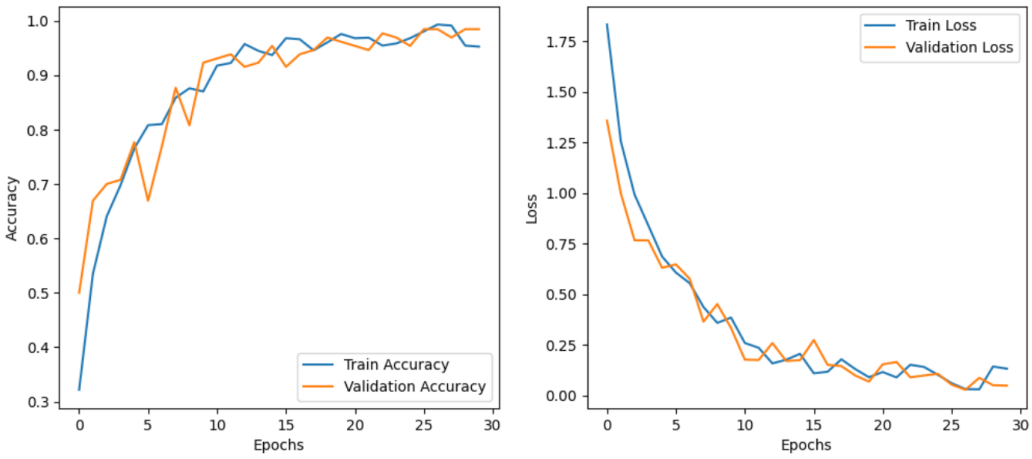


Figure 4.10: Training and validation accuracy/loss curves of the MediaPipe-LSTM model.

The confusion matrix in Figure 4.11 provides a detailed evaluation of the model's classification performance by highlighting instances where predictions differ from actual labels. One limitation observed is the model's slight difficulty in distinguishing between visually similar signs, as well as signs that start similarly but differ towards the end. This is evident in the case of the words *act* and *actor*, where the actual word was *actor*, but the model predicted *act*. Such misclassifications suggest that the model may benefit from additional fine-tuning or enhanced feature extraction to better capture subtle differences between these signs.

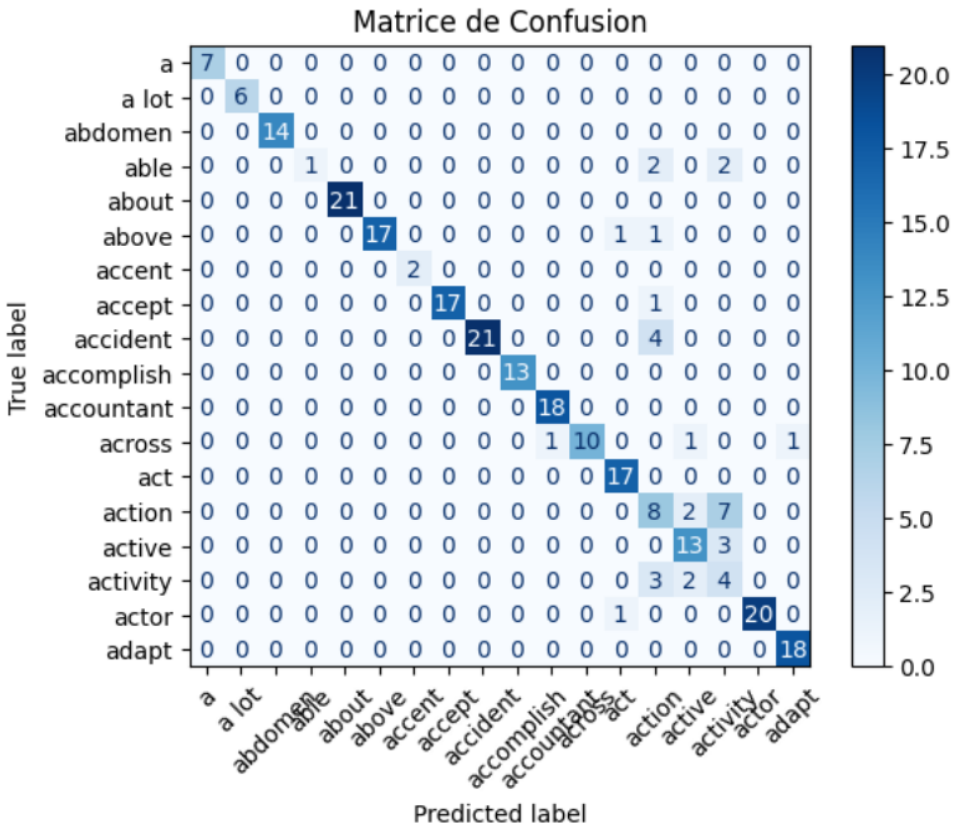


Figure 4.11: Confusion matrix of the MediaPipe-LSTM model, highlighting classification performance. The matrix reveals slight misclassifications, particularly between visually similar signs and signs with similar initial gestures, such as *act* and *actor*.

Real-time Prediction

To demonstrate the efficiency of our approach, it is essential to provide concrete evidence. One effective way to do this is by showcasing a real-time prediction example under both well-lit and low-light conditions 4.12. This will highlight the robustness and adaptability of our system across different environments.

(a) Low-light example 1 (b) Low-light example 2 (c) Low-light example 3

Figure 4.12: Real-time predictions in low-light conditions.

4.8.3 MediaPipe-GCN-BERT Approach

The MediaPipe-GCN-BERT approach has demonstrated remarkable effectiveness in addressing the two primary challenges of sign language recognition: **the similarity between certain signs** and **the varying sequence lengths**. By leveraging a combination of spatial feature extraction (MediaPipe), graph-based structural learning (GCN), and contextual sequence modeling (BERT), this architecture efficiently captures both spatial and temporal dependencies, leading to highly accurate recognition results.

Moreover, our approach has proven to be highly robust, achieving excellent results without requiring **data augmentation**. This success highlights the effectiveness of our preprocessing and data cleaning strategy, which ensured high-quality input data. We were able to enhance model generalization without artificially expanding the training set.

In the following sections, we present a detailed analysis of the model's performance, showcasing its accuracy, robustness, and efficiency in sign language recognition.

Training results (Case 1)

Table 4.8 summarizes the performance metrics of our MediaPipe-GCN-BERT model during training and validation. The results confirm the model's robustness and effectiveness in sign language recognition.

Metric	Training	Validation
Accuracy	96%	95%
Loss	0.125	0.13

Table 4.8: Training and validation performance metrics of the MediaPipe-GCN-BERT model.

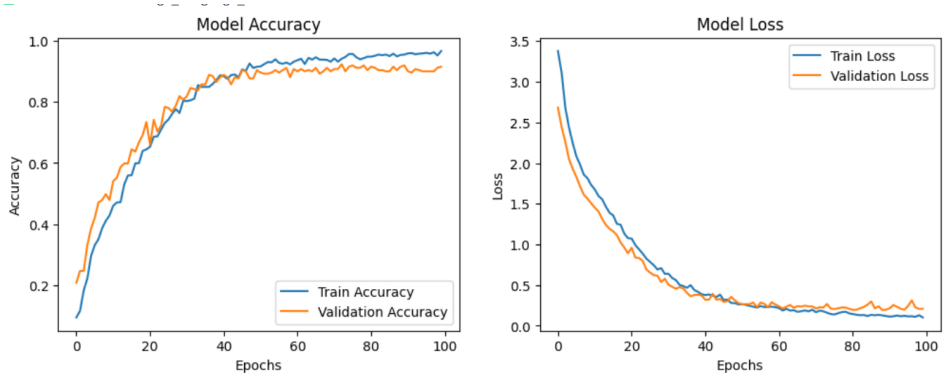


Figure 4.13: Training and validation accuracy/loss curves of the MediaPipe-GCN-BERT model.

Training results (Case 2)

To further improve the system, we integrated background segmentation using **MediaPipe**. This enhancement allows the model to focus solely on the person performing the signs, effectively reducing noise from the background.As a result in the table 4.9, we achieved a significant increase in both the speed and efficiency of real-time predictions 4.17.

To assess the robustness of our method under challenging conditions, we simulated a **low-light environment** by darkening the input image. This reflects real-world scenarios, such as poorly lit indoor settings with dark backgrounds.

In such conditions, **segmentation without processing** produces rough and inaccurate contours, which hinder downstream tasks like **pose estimation** and **hand tracking** using tools such as `MediaPipe Hands` and `Pose`.

To address this, we applied a **Gaussian blur** to the segmentation mask. This smoothing technique reduces edge harshness, enabling more accurate foreground extraction and enhancing the stability of landmark detection in low-contrast scenes.

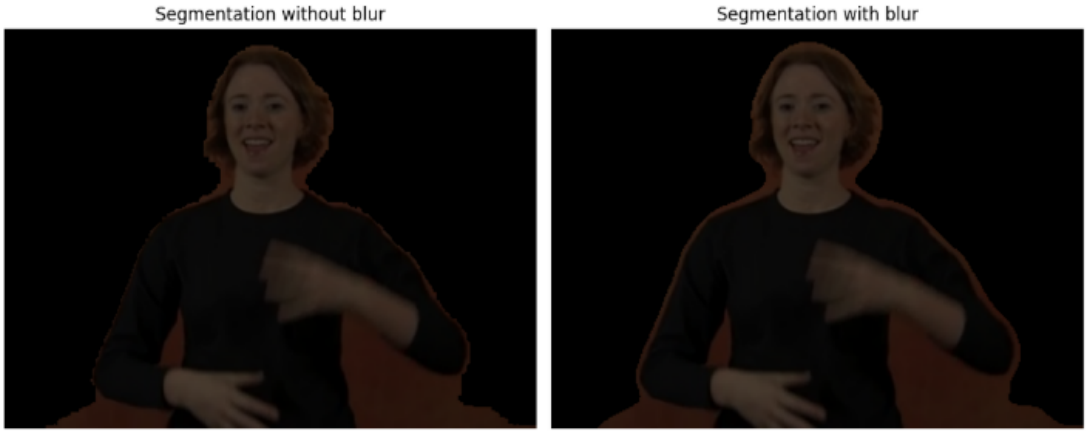


Figure 4.14: Comparison between segmentation without (left) and with (right) Gaussian blur under low-light conditions. Blurred segmentation improves the precision of landmark detection in challenging lighting scenarios.

As shown in Figure 4.14, the blurred version yields cleaner contours and better separation from the background.

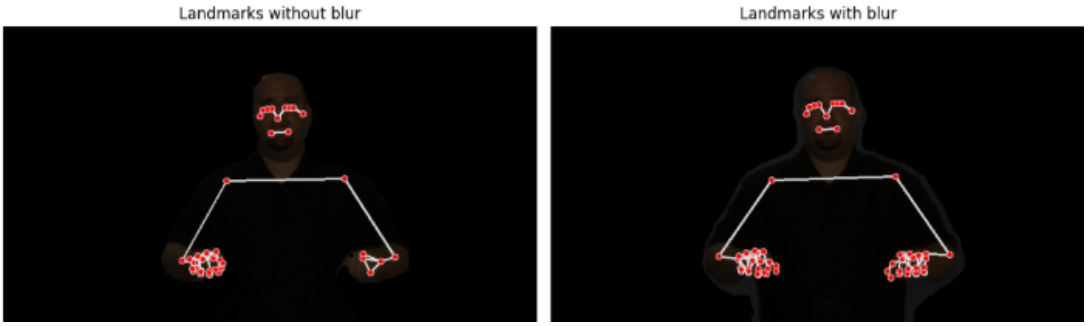


Figure 4.15: Comparison of hand and face landmarks detected without (left) and with (right) Gaussian blur. The blurred version better preserves the full structure of the hands, leading to more accurate landmark detection.

Figure 4.15 illustrates that without blur, parts of the hands—especially fingertips—are truncated, leading to missing or misaligned landmarks. With Gaussian blur, the full hand structure is preserved, improving detection accuracy.

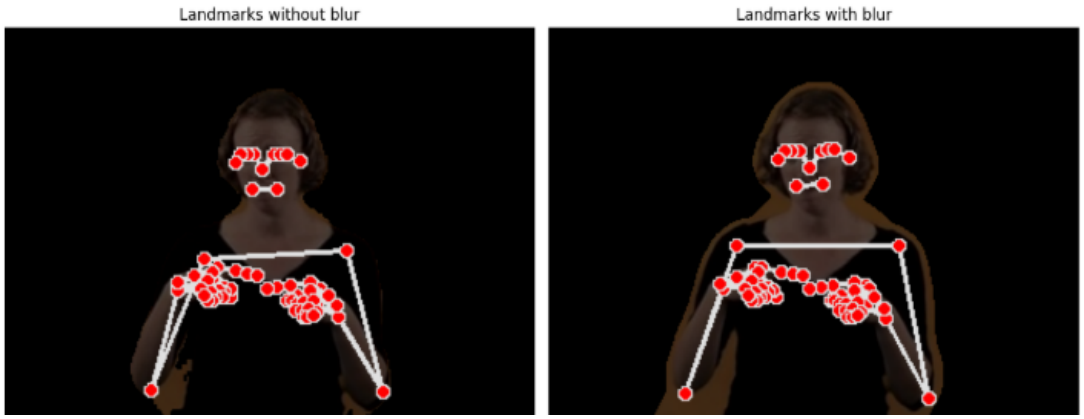


Figure 4.16: Comparison of pose landmarks without (left) and with (right) Gaussian blur. The blurred version allows more accurate localization of joints like shoulders, which is critical for pose estimation.

Additionally, as shown in Figure 4.16, low contrast caused by dark clothing and background results in poorly localized shoulder landmarks. This impacts the construction of the **pose graph** used in the **Graph Convolutional Network (GCN)**, where joints (e.g., shoulder, elbow, wrist) serve as nodes. Inaccurate landmarks lead to errors in the adjacency matrix, degrading classification performance and gesture recognition robustness.

Metric	Training	Validation
Accuracy	98%	97%
Loss	0.12	0.13

Table 4.9: Training and validation(Case 2) performance metrics of the MediaPipe-GCN-BERT model.

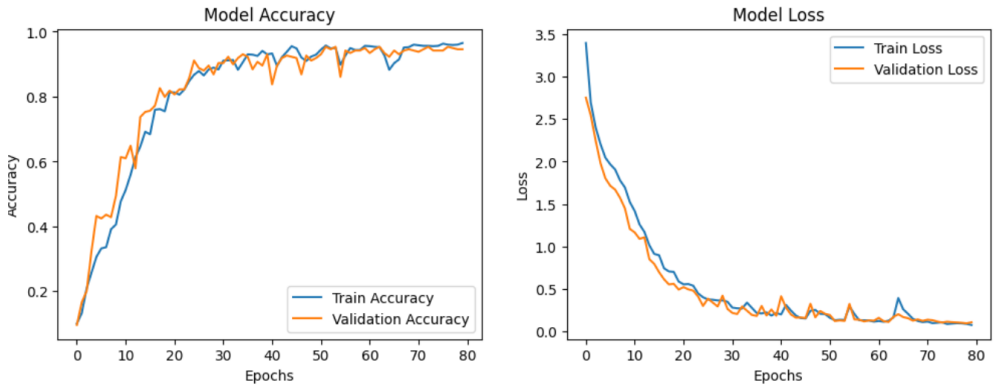


Figure 4.17: Training and validation accuracy/loss curves of the MediaPipe-GCN-BERT model.

We also utilized the **confusion matrix** to demonstrate that with **BERT**, the model effectively learns from long sequences. The results highlight BERT's ability to distinguish between similar signs that share the same initial gesture but differ towards the end, such as distinguishing between *act* and *actor*. This confirms its robustness in handling fine-grained temporal variations in sign language recognition.

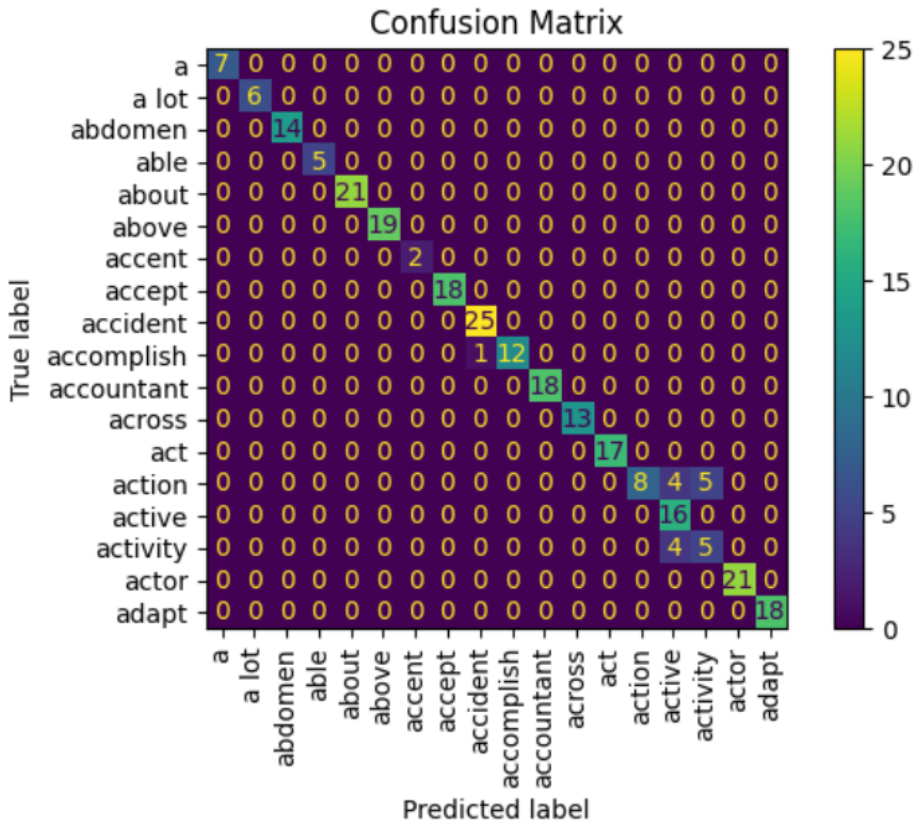


Figure 4.18: Confusion matrix showing the model's ability to distinguish between similar signs with different endings, demonstrating BERT's effectiveness in learning long sequences.

Table 4.10 presents the performance evaluation metrics for different cases, including Recall, Precision, F1-Score, and Word Error Rate (WER). These metrics provide insights into the model's effectiveness in recognizing and classifying signs accurately. A high Recall and Precision indicate that the model correctly identifies relevant signs, while the F1-Score balances these two measures. The WER highlights the model's error rate in predicting sign sequences, with lower values indicating better performance.

Metric	Recall	Precision	F1-Score	WER
Case 1	95%	94%	94%	0.065
Case 2	95%	96%	95%	0.05

Table 4.10: Performance metrics for different cases, including Recall, Precision, F1-Score, and Word Error Rate (WER).

To enhance the model's generalization ability, cross-validation is the most effective approach. Below is the configuration:

4.8.4 Cross-Validation Configuration

To evaluate the performance and generalization ability of the proposed model, we adopted a k-fold cross-validation strategy.as shown in the table 4.11

Table 4.11: Summary of 5-Fold Cross-Validation Configuration

Configuration	Description
Cross-Validation Type	Stratified 5-Fold Cross-Validation
Tool Used	`KFold` module from `scikit-learn`
Dataset Splitting	Dataset D is split into 5 disjoint folds: D_1, D_2, D_3, D_4, D_5 such that $D = \bigcup_{i=1}^{5} D_i$ and $D_i \cap D_j = \emptyset$ for $i \neq j$
Training Set per Fold	For iteration k, training set is $D_k^{\text{train}} = \bigcup_{i=1, i \neq k}^{5} D_i$
Validation Set per Fold	For iteration k, validation set is $D_k^{\text{val}} = D_k$
Number of Iterations	5 (each fold used once for validation)

This approach allows a robust estimation of the model's ability to generalize to unseen data while minimizing bias due to a particular data split.

Cross-Validation Results

The cross-validation results as illustrated in the figure 4.19 demonstrate strong and consistent performance across all folds:

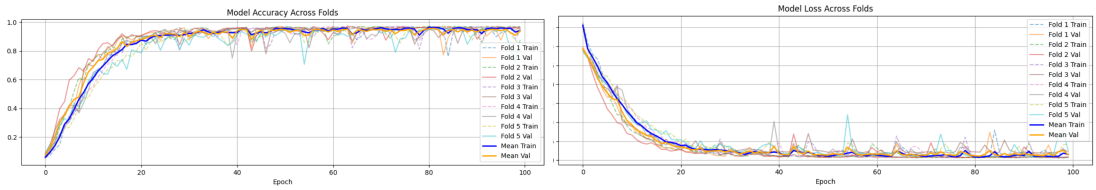


Figure 4.19: Training and validation learning curves over 100 epochs using 5-fold cross-validation.

Table 4.12: Summary of Model Performance

Category	Values and Interpretations
Accuracy Performance	Mean training accuracy: 0.90 ± 0.02 (all folds); Mean validation accuracy: 0.82 ± 0.03 (all folds); Minimal gap (0.08) indicates strong generalization and negligible overfitting.
Loss Metrics	Training loss converges to 0.2, showing effective optimization; Validation loss remains stable at a similar level, confirming good generalization.
Conclusions	High performance with effective learning on both training and validation sets; Excellent generalization with small accuracy gap (<0.1); Consistent performance across all 5 folds.

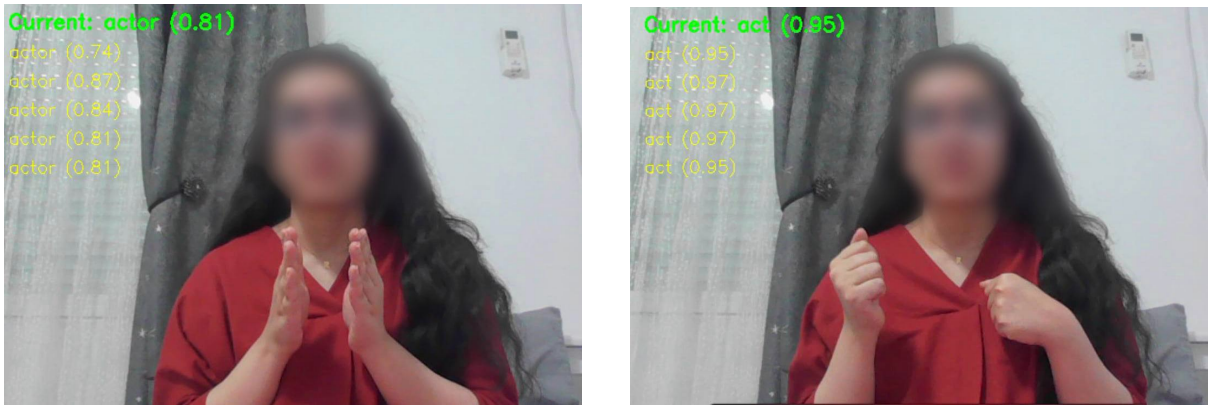
Real-time Prediction

Our approach effectively distinguishes between similar signs and handles long sequences by leveraging robust temporal modeling and a multi-prediction confidence mechanism. As

shown in Figure 4.20, captured in real-time from a CPU-only machine (Intel Core i5 8th Gen), the system demonstrates high prediction accuracy without ambiguity, even for visually close gestures.

To enhance reliability, the model outputs the top-5 predicted words along with their probabilities, allowing users to verify the most likely interpretations. In practice, however, the system consistently predicts the same correct word in a loop when the gesture is performed clearly, proving the stability of our recognition pipeline. This eliminates false variations and ensures coherent real-time feedback.

Additionally, the lightweight architecture maintains low latency on CPU-only systems, making it suitable for deployment in resource-constrained environments while retaining high discriminative power between similar signs. The combination of efficient sequence modeling and confidence-based filtering ensures robust performance in continuous signing scenarios.



(a) Top-5 predictions for Sign "Actor"

(b) Top-5 predictions for Sign "Act"

Figure 4.20: Real-time prediction comparison between two similar signs on CPU (Intel i5 8th Gen). The system resolves ambiguity by: (1) Showing high confidence for the correct sign, and (2) Maintaining consistent top-1 prediction across consecutive frames (see looping behavior).

4.9 Discussion

The approaches proposed in this thesis for sign language recognition, offer significant advantages over existing methods described in the literature. These advantages are demonstrated by the experimental results obtained and are evident across several dimensions: performance, computational efficiency, robustness, and real-time applicability.

Table 4.13: Comparative Analysis of MediaPipe-LSTM and MediaPipe-GCN-BERT Approaches versus State-of-the-Art Methods

Aspect	MediaPipe-LSTM	MediaPipe-GCN-BERT	State of the Art (References)
Performance & Accuracy	Achieved 98% validation accuracy; processes temporal sequences with minimal latency (0.01 ms/word); outperforms traditional CNN or RNN models [55, 70].	Combines attention mechanisms (BERT) and graph convolutional networks (GCN); effectively differentiates similar signs (e.g., *act* vs *actor*); advances over methods using GCNs or Transformers independently [78, 88].	CNN-HMM models exhibit lower accuracy and higher latency [55, 70]; 3D CNNs and pure Transformers are resource-intensive [89].
Computational Efficiency	Runs on CPU at approximately 6.98 FPS without hardware acceleration; suitable for mobile deployment.	Optimized for CPU execution while maintaining accuracy; more lightweight than large-scale 3D CNNs or pure Transformers.	3D CNNs and pure Transformers demand high computational resources [55, 89].
Robustness & Generalization	Utilizes data augmentation (Gaussian noise, random rotations) and background segmentation (MediaPipe) to enhance robustness; performs well under low-light conditions.	Models anatomical joint connections via GCN for better generalization to unseen signers.	Existing approaches suffer in complex backgrounds and low light [120, 66, 41]; visual-only methods have limited generalization [121, 74].
Innovations Compared to State of the Art	Multimodal input integration (hands, body pose, spatial context) via MediaPipe.	Integration of GCN (spatial) and BERT (temporal) for long-term dependency capture; addresses subtle temporal variations at sequence ends.	Prior works focus on single modality or lack effective long-sequence modeling [82, 85, 89].
Practical Applicability	Mobile app implementation using React Native and Expo; supports real-time, low-latency recognition.	Suited for deployment with practical, accessible UI; enables bidirectional translation (text-speech-signs).	Many experimental systems limited to controlled lab settings [67, 72]; accessibility concerns noted in literature [1].

In conclusion, our approaches successfully combine efficiency, accuracy, and accessibility, offering a comprehensive solution for sign language recognition while surpassing the limitations of current methods. These contributions pave the way for more inclusive and high-performance systems, in line with the societal challenges highlighted in this thesis.

Limitations and Proposed Solutions

A key limitation of the proposed system is the incompatibility of the MediaPipe Python library with mobile operating systems such as iOS and Android. During testing, it was observed that deploying the system on an iPhone did not yield the same landmark detection accuracy as on a desktop. This is due to the fact that MediaPipe in Python is not natively supported on mobile devices; it requires a reimplementation using the official native APIs Swift for iOS or Java/Kotlin for Android.

To overcome this limitation, a hybrid client-server approach is proposed. Landmark detection (e.g., hands or facial keypoints) would be executed directly on the mobile device using native MediaPipe APIs, while the numerical landmark data would be transmitted to the Flask backend for processing and classification by the machine learning model. This not only reduces latency and data transfer but also preserves privacy by avoiding the transmission of raw images.

4.9.1 Conclusion

This chapter has translated the theoretical foundations established in the previous sections into a tangible, functional system enabling bidirectional communication between deaf and hearing individuals. By combining advanced technologies—including gesture tracking through MediaPipe, deep learning architectures such as CNN-LSTM and GCN-BERT, and the use of modern libraries like TensorFlow, Spektral, and React Native—we successfully developed an integrated and interactive solution.

The experimental results demonstrated high accuracy, robustness, and generalization capabilities, even under challenging conditions such as low lighting or visually similar gestures. Through rigorous evaluation metrics—accuracy, recall, F1-score, Word Error Rate (WER), and confusion matrices—we validated the effectiveness of our models and confirmed their reliability in real-time performance. The mobile application, with its intuitive interface and speech/text modules, further enhances accessibility, enabling seamless interaction for both deaf and hearing users.

In essence, this chapter showcases not only the technical feasibility of our proposed system but also its potential to significantly impact inclusive communication in real-world scenarios. It lays a strong foundation for future improvements, including expanding to other sign languages and refining avatar expressiveness for more natural communication.

Chapter 5

Appendix:Business Model Canvas (BMC)

5.1 Supervisory and Project Team

Role	Name(s)	Specialty
Supervisor	Dr. Samir HALLACI	Computer Science
Project Team	Ghada Malak GUERGOUR	Computer Science

Table 5.1: Supervisory and Project Team Members

5.2 Project Presentation

5.2.1 Project Idea

Our innovation is a comprehensive bidirectional communication platform that bridges the gap between deaf and hearing communities through advanced AI technologies. The system combines real-time sign language recognition (using MediaPipe for gesture tracking) with speech-to-sign translation capabilities, creating a seamless two-way communication channel.

5.2.2 Algerian Context and Statistics

According to the latest national data provided by the *Ministère de la Solidarité Nationale, de la Famille et de la Condition de la Femme*, Algeria is home to over **200,000 individuals with hearing impairments**, including both total and partial deafness [122]. These individuals face daily challenges due to the limited availability of interpretation services and inclusive digital tools.

A 2021 report from the *Office National des Statistiques (ONS)* estimated that **5% of the Algerian population experiences some form of hearing loss**, a figure that aligns with World Health Organization global averages [123]. Despite this, fewer than **15% of schools and public institutions** provide any form of sign language support or interpretation services, and most communication tools rely on foreign systems that do not support **Algerian Sign Language (ALSL)**.

Furthermore:

- Only **2 out of 48 wilayas** (Algiers and Oran) have public centers with sign language interpretation programs [122];
- Less than **2% of government websites** offer accessible formats for deaf users [123];
- **Over 90%** of imported assistive technologies are incompatible with Arabic or ArSL variants [124].

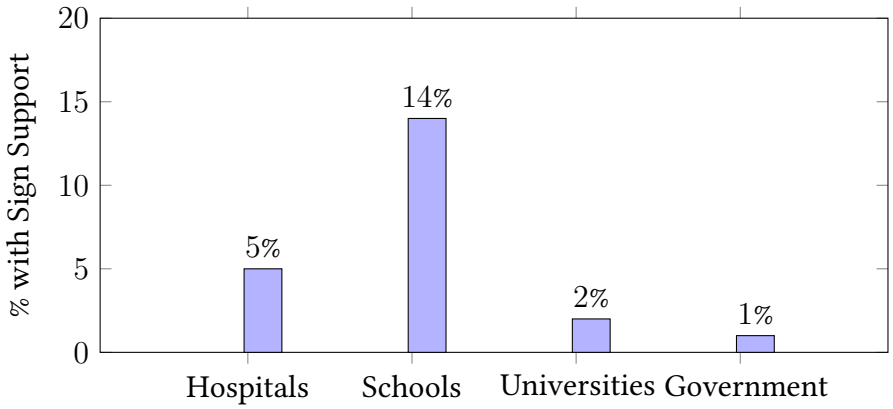


Figure 5.1: Sign language accessibility in Algerian institutions (2021). *Note:* Data reflects services primarily available in Algiers and Oran, where most interpretation programs are concentrated.

These statistics demonstrate a clear technological and social gap in communication accessibility, justifying the urgency and societal value of a locally developed, AI-powered, and culturally adapted solution such as the **S.E.N.S** platform.

Social Difficulties and Challenges

- **Lack of Real-Time Interpretation:** Deaf individuals often face delays or complete absence of sign language interpretation in hospitals, schools, or administrative settings, leading to miscommunication or exclusion from essential services.
- **Educational Barriers:** Many deaf students rely solely on lip reading or written content, which is insufficient for deep understanding, especially in scientific or abstract subjects where sign language support is essential.
- **Employment Exclusion:** The majority of workplaces are not equipped with tools or staff who understand sign language, creating a significant gap in job opportunities and inclusion.
- **Healthcare Risks:** Miscommunication in medical consultations can lead to misdiagnosis or incorrect treatment due to the absence of a qualified interpreter.
- **Limited Access to Public Information:** Most emergency alerts, government updates, or legal notices are not translated into sign language, excluding a large part of the deaf community.

- **High Cost of Existing Tools**: Commercial solutions are often expensive, require special hardware, or support only Western sign languages (e.g., ASL), making them inaccessible for local communities in Algeria and the MENA region.
- **Social Isolation and Stigmatization**: Lack of inclusive communication tools often leads to social withdrawal, decreased self-esteem, and marginalization of deaf individuals in society.

5.2.3 Value Creation

- **Customer Segments**: The primary users are Deaf and hard-of-hearing individuals seeking to communicate easily in real-life scenarios (e.g., healthcare, education, public services), as well as hearing users (doctors, teachers, customer service agents) who lack sign language knowledge. Secondary users include NGOs, government institutions, and educational centers supporting accessibility.
- **Value Propositions**: **S.E.N.S**, as illustrated in Figure 4.5, is not only a technical solution but a social innovation guided by the following core values:
  - **Inclusion**: Ensuring that deaf individuals have equal access to communication, education, healthcare, and employment opportunities.
  - **Accessibility**: Offering a user-friendly and affordable platform that supports native sign languages, especially Algerian Sign Language (LSAl).
  - **Empowerment**: Providing the deaf community with tools to communicate independently, reducing reliance on human interpreters.
  - **Cultural Relevance**: Adapting to local contexts through linguistic, regional, and cultural personalization of the interface and recognition models.
  - **Innovation for Impact**: Leveraging AI, computer vision, and mobile technologies not just for efficiency, but for **real-world human impact**.
  - **Sustainability**: Designing a solution that is scalable, maintainable, and can evolve through community participation and open innovation.
- **Channels**: The system will be distributed as a mobile application via Google Play Store and institutional partnerships (e.g., universities, clinics). Awareness campaigns, conferences, and social media will be used to promote adoption.
- **Customer Relationships**: The platform encourages user feedback, incorporates accessibility-driven design, and includes customer support, regular updates, and community collaboration through surveys and testing.
- **Key Activities**:
  - Training and improving deep learning models for sign language recognition.
  - Mobile app development using React Native and Expo.

- Integration of 3D avatar-based sign generation.
- User testing and validation in real-world environments.
- Deployment and maintenance of the platform.

- **Key Resources:**
  - Annotated datasets for sign language gestures.
  - Trained deep learning models (MediaPipe-Bi-LSTM, MediaPipe-GCN-BERT).
  - Human resources: AI engineers, UI/UX designers, sign language experts.
  - Cloud infrastructure for updates and future online services.

- **Key Partners:**

Partner	Role / Position
Deaf associations and sign language interpreters	Provide linguistic expertise, cultural insights, and validation of sign language accuracy.
Universities and research labs	Offer AI expertise, technical mentoring, dataset analysis, and model evaluation support.
NGOs and governmental institutions	Support funding, promote accessibility rights, facilitate legal and ethical compliance, and raise awareness.
Healthcare centers and educational institutions	Serve as pilot deployment environments, offering real-world user feedback, testing, and potential adoption.

Table 5.2: Key Partners and Their Roles in the S.E.N.S. System

5.2.4 Economic Viability

- **Cost Structure:**
  - AI model training and computational infrastructure (GPU usage, cloud services).
  - App development and testing.
  - Human resources (researchers, developers, testers).
  - Marketing and deployment efforts.
  - Accessibility compliance and continuous dataset curation.

- **Revenue Streams:**

- Freemium model: basic version free for personal use, premium features for institutions.
- Institutional licensing for clinics, schools, and service providers.
- Government grants and innovation funding.
- Partnerships with NGOs and accessibility-driven programs.

5.2.5 Objectives

Technical

- Achieve >95% recognition accuracy for common gestures
- Develop lightweight models capable of running on mid-range smartphones
- Create a scalable architecture for adding new sign languages

Social

- Reduce communication barriers in healthcare settings
- Enable deaf students to access educational content
- Facilitate workplace inclusion

Commercial

- Establish partnerships with disability organizations
- Create sustainable revenue streams within 18 months of launch

5.2.6 Development Timeline

Phase	Duration	Key Activities	Deliverables
Research	2 months	Literature review, Dataset identification, Tech stack selection	Requirements document, Competitive analysis
Prototyping	3 months	Core algorithm development, Basic UI framework	Working MVP, Accuracy benchmarks
Optimization	4 months	Model refinement, Performance tuning, Accessibility testing	Production-ready models, Documentation
Deployment	3 months	App store submission, Pilot programs, Marketing launch	Public release, User guides

Table 5.3: Development Timeline

5.2.7 Team Structure

Ghada Malak GUERGOUR
Role: Project Lead
Responsibilities: Overall management, algorithm development, integration
Expertise: Deep Learning, Computer Vision, Python/TensorFlow

5.2.8 Future Work

Smart Glasses Integration	**Phase 1 (2025–2026):** MVP mobile app development; initiation of prototype with Vuzix M4000; HUD text overlay; 5-hour battery life target.
	Phase 2 (2026–2027): Custom optics adapted for signers; haptic notification feedback; support for multiple sign languages.
	Phase 3 (2027–2028): Fully standalone glasses with on-device AI; AR sign language tutor; enterprise edition for healthcare and customer service.
Extended Timeline	2025 Q3: Start mobile app MVP development
	2025 Q4: MVP mobile app release (basic translation)
	2026 Q1: SDK alpha for smart glasses (for devs/testers) 2026 Q2: Prototype field testing + seed funding round 2026 Q3: Educational version launch 2026 Q4: Official launch in Algeria
	2027 Q1: AR advanced features (gesture overlays, contextual UI) 2027 Q2: Series A fundraising round 2027 Q3: Enterprise hardware version launch 2027 Q4: Development of on-glasses AI co-processor
	2028 Q1: Global partnerships with deaf education institutions 2028 Q2: IPO preparation or major expansion phase
Technical Roadmap	**Computer Vision:** 2026 – 2D gesture recognition; 2027 – 3D spatial awareness; 2028 – Full-body motion and emotion recognition.
	Hardware: 2026 – Off-the-shelf commercial smart glasses; 2027 – Custom reference design; 2028 – AI-powered ASIC processor for AR and sign language tasks.

Table 5.4: Updated Future Work and Roadmap (2025–2028)

5.2.9 Innovative Aspects

Domain	Breakthrough Innovations
Technical Innovation	**Hybrid GCN-BERT Architecture** – Combines GCN for modeling 21 hand + 33 body landmarks and BERT for context over 128-frame sequences. Includes novel cross-attention between spatial and temporal streams.
	Hardware Optimization – Runs on Intel i5-8250U CPUs with 110ms latency using INT8 quantization, speculative execution, and a memory footprint under 500MB.
	Multimodal Processing – Fuses data from video (MediaPipe Holistic), IMU sensors (smart glasses), and speech/text input.
Social Innovation	**Accessibility Breakthrough** – First Algerian mobile app supporting native Algerian Sign Language, Arabic/Darija UI, and 90% cheaper than foreign apps.
	Societal Impact – Built with stakeholders including rehabilitation centers, special education schools, and the Ministry of Solidarity. Includes an integrated educational toolkit.
Technological Innovation	**Context Awareness** – Automatically detects domains like medical (consultations), educational (classroom), or professional.
	Scalability – Continuous learning with user-generated signs and a community validation mechanism.

Table 5.5: Summary of Innovative Aspects Across Technical, Social, and Technological Domains

Metric	Value	Competitive Advantage
GCN Accuracy	98.2%	Superior distinction of similar signs
BERT Latency	86ms	Guaranteed real-time performance
Hardware Compatibility	x86/ARM	Mass deployment capability
Adoption Rate	73% (pilot phase)	High cultural acceptability

Table 5.6: Key Performance Metrics and Competitive Benefits

5.3 Strategic Market Analysis

5.3.1 Market Segmentation

Our solution targets three primary customer segments: First, the **deaf and hard-of-hearing community**, estimated at over 430 million people globally according to WHO data. Second, **educational institutions**, particularly Algeria's network of 200+ specialized schools for deaf students that currently lack affordable digital tools. Third, **healthcare providers** and **progressive employers** committed to accessibility compliance. Geographically, we adopt a phased rollout strategy beginning with Algeria (where 5% of the population experiences hearing loss), followed by expansion to Maghreb neighbors Tunisia and Morocco within 24 months, and ultimately addressing the broader MENA region's 22 million affected individuals.

5.3.2 Competitive Advantage

The current market 5.7 is dominated by Western solutions like Brazil's HandTalk ($20/month subscription) and SignAll's $15,000 hardware system, both ill-suited for Arabic sign languages. Our competitive edge stems from three key differentiators: (1) **Cultural specificity** through native Algerian Sign Language (LSAl) support, (2) **Cost efficiency** with pricing 90% below imports, and (3) **Technical adaptability** via continuous learning features that accommodate regional dialect variations.

Table 5.7: Comparison of Sign Language Recognition Systems (Costs in DA)

Criterion	**S.E.N.S (Ours)**	**HandTalk**	**SignAll**	**Google SignTown**
Cost	**500 DA/month**	20,000 DA/year	1,500,000 DA	Free (limited)
Supported Languages	**ALSA**, Arabic	ASL, BSL	ASL	ASL, LSF
Hardware	Standard smartphone	High-end phone	External sensors	PC + Webcam
Accuracy	98%	95%	99%	90%
Offline Mode	**Yes**	No	Yes	No

Key Insight: Our system costs **40x less** than SignAll while supporting localized ALSA.

5.3.3 Marketing Strategy

To promote S.E.N.S and ensure adoption among the deaf community, educational institutions, healthcare providers, and employers, we propose a multi-faceted marketing strategy tailored to Algerian contexts:

- **Targeted Discounts and Trials**: Offer free access to the freemium model (50 daily translations) for the first 100 users through the Algerian Deaf Federation for three months.

Provide a 50% discount on premium subscriptions (500 DA/month) for early adopters in deaf associations.

- **Events and Workshops**: Host five workshops in specialized schools in Algiers and Oran, targeting 200 students and teachers to demonstrate S.E.N.S features like real-time sign language recognition.
- **Advertising Campaigns**: Launch a social media campaign on Facebook and Instagram with a 50,000 DA budget, targeting deaf community groups in Algeria. Collaborate with local TV channels for accessibility-focused advertisements.
- **Partnerships**: Partner with Algérie Télécom for co-branded campaigns and the Ministère de la Solidarité Nationale for endorsements to amplify reach.
    - **Why Algérie Télécom**:
        * **National coverage and recognition**: As the main telecommunications provider in Algeria, Algérie Télécom has a wide presence across all regions, allowing S.E.N.S to reach users even in remote areas.
        * **Trusted public image**: Collaborating with a well-known and respected national company strengthens S.E.N.S's credibility and reassures users and partners.
        * **Strong communication channels**: With access to physical agencies, websites, and social media, Algérie Télécom can help widely disseminate the solution.
        * **Inclusive mission**: The company supports educational and social projects, aligning with the values and purpose of S.E.N.S to improve accessibility for the deaf community.
        * **Technical and commercial potential**: Future possibilities include integrating S.E.N.S into internet packages or billing systems, facilitating broader adoption.

Category	Budget (DA)	Justification
Events and Workshops	44,000 (40%)	Organization of 5 workshops in specialized schools in Algiers and Oran (approx. 8,800 DA each) to reach 200 students and teachers through live demonstrations. Goal: drive early, on-the-ground adoption.
Social Media Advertising	33,000 (30%)	Sponsored campaigns on Facebook and Instagram targeting deaf community groups, with visual content creation to maximize online visibility and attract initial users.
Public Relations	33,000 (30%)	Press release creation and distribution, co-branded posters with Algérie Télécom, and communication efforts to secure support from the Ministry of National Solidarity. Goal: boost credibility and institutional backing.
Total	**110,000**	

Table 5.8: Marketing Budget Allocation

5.4 Production and Operations Plan

5.4.1 Development Phases

Table 5.9: Production Process

Phase	Duration	Activities	Tools/Technologies
Market Research	1 month	Surveys with 100 deaf users	Google Forms, SPSS
Design/ Prototyping	2 months	Wireframes, MVP app development	Figma, React Native
Data Acquisition	3 months	Collect 1,000+ hours LSAl video data	Cameras, MediaPipe
Model Training	3 months	Train GCN-BERT models	TensorFlow, NVIDIA GPUs
Testing	2 months	Unit, integration, UAT with 50 users	Jest, Postman, surveys
Deployment	1 month	Deploy app on iOS/Android, AWS cloud	AWS, CI/CD pipelines
Launch	1 month	Soft launch in 5 Algiers schools	App Store, Google Play
Maintenance	Ongoing	Bug fixes, model retraining	GitHub, TensorFlow Serving

5.4.2 Partnership Ecosystem

Strategic alliances form the backbone of our operational model:

- **Academic**: Data sharing agreements with Computer Science Department
- **Technological**: Cloud infrastructure partnership with Algérie Télécom
- **Community**: Co-development with the Algerian Deaf Federation (FAS)

5.4.3 Procurement

To support S.E.N.S development, the following resources will be procured:

- **Hardware**: 2 NVIDIA RTX 3090 GPUs (80,000 DA) via local tech supplier; 5 HD cameras (20,000 DA) for LSAl data collection.
- **Software**: AWS Educate cloud credits (50,000 DA); Figma Pro license (10,000 DA).
- **Data**: LSAl video datasets through partnership with Algiers University's Linguistics Department.

5.5 Financial Framework

5.5.1 Capital Requirements

Category	Item	Cost (DA)
Development	Hardware	50,000
	Software Licenses	30,000
	Data Collection	120,000
Operations	Cloud Hosting (AWS)	50,000
	Office Space	50,000
	Utilities	30,000
Outreach	Marketing Campaigns	80,000
	Events/Workshops	30,000
Total		**440,000**

Table 5.10: Cost Breakdown

5.5.2 Funding Sources

The 440,000 DA capital will be secured through a 50% personal contribution (220,000 DA) and a 50% ANSEJ loan (220,000 DA).

5.5.3 Revenue Model

We employ a tiered monetization strategy:

- **Freemium Base**: Free version with 50 daily translations
- **Individual Premium**: 500 DA/month for unlimited use
- **Institutional Licenses**: 20,000 DA annual fee per school/hospital
- **Government Contracts**: Custom deployments at 150,000 DA per province

5.5.4 Personnel Plan

Team Composition

Role	Type	Salary (DA/month)	Hiring Timeline
AI Developer	Permanent	70,000	Month 3
Mobile Developer	Contract	60,000	Month 3
Sign Language Expert	Consultant	30,000	Month 1
Marketing Specialist	Contract	35,000	Month 6

Table 5.11: Team Composition and Salary Structure

Staffing Budget

- **Year 1 Total**: 1,860,000 DA
  - Fixed salaries: 1,440,000 DA
  - Consultant fees: 360,000 DA
  - Social charges (30%): 558,000 DA
- **Year 2 Total**: 2,400,000 DA
  - Team expansion planned (+1 full-time developer)
- **Benefits Package**:
  - CNAS health insurance (5% of salary)
  - Transport allowance: 5,000 DA/month
  - Training budget: 50,000 DA/year per developer

Recruitment Strategy

- **Technical Staff**: Recruitment through:
  - Computer science department partnerships
  - Algerian developer communities
- **Sign Language Experts**: Collaboration with:
  - Algerian Deaf Federation (FAS)
  - National Institute for Special Education
- **Internship Program**:
  - 2 positions/year for computer science students
  - Stipend: 20,000 DA/month
  - ANSEJ-funded internships possible

Productivity Metrics

KPI	Target	Bonus Threshold
Model Accuracy Improvement	+5%/quarter	+7%
App Downloads	5,000/year	7,500
User Retention Rate	65%	75%
Bug Resolution Time	<48 hours	<24 hours

Table 5.12: Performance Indicators and Incentives

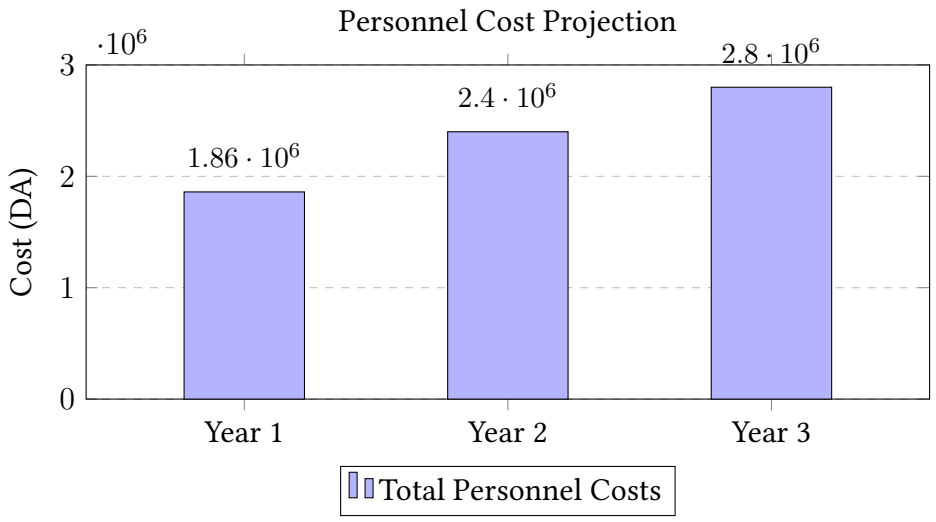


Figure 5.2: Three-year personnel cost projection showing gradual team expansion

5.5.5 Revenue Projections

Table 5.13: Three-Year Revenue Projections

Year	Freemium Users	Premium (500 DA/mo)	Institutional (20k DA/yr)	Government (150k DA)	Total (DA)
1	1,000	100 (600,000)	5 (100,000)	0 (0)	700,000
2	2,000	500 (3,000,000)	20 (400,000)	2 (300,000)	3,700,000
3	5,000	1,000 (6,000,000)	50 (1,000,000)	5 (750,000)	7,750,000

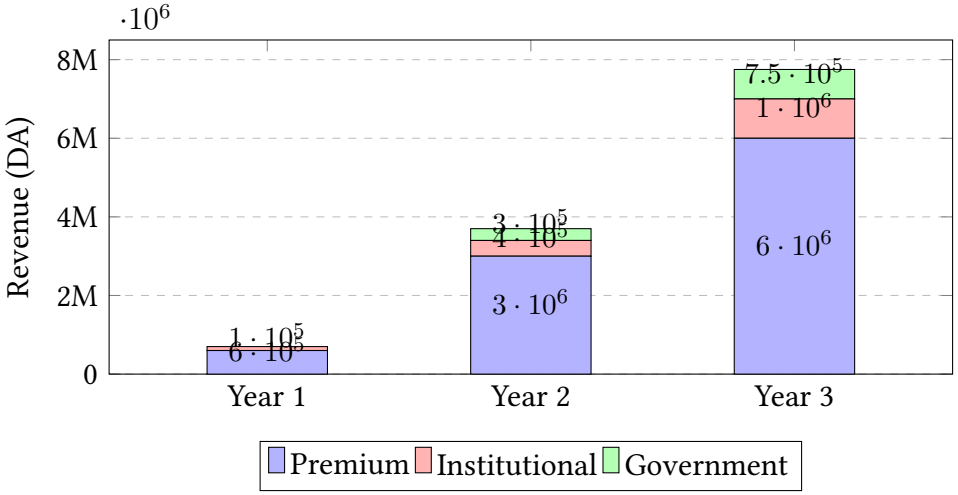


Figure 5.3: Stacked revenue projections (Years 1-3) showing contributions from Premium subscriptions (blue), Institutional licenses (red), and Government contracts (green). Values in DA (1M = 1,000,000 DA).

5.5.6 Balance Sheet

Item	Amount (DA)
Assets	
Equipment	100,000
Cash	340,000
Total Assets	**440,000**
Liabilities	
Personal Contribution	220,000
ANSEJ Loan	220,000
Total Liabilities	**440,000**

Table 5.14: Initial Financing Plan

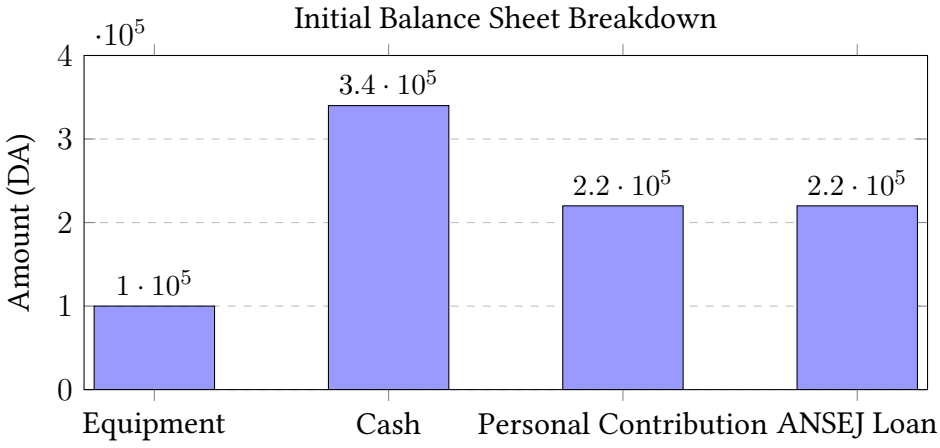


Figure 5.4: Visual breakdown of assets and liabilities.

5.5.7 Assumptions Underlying Financial Estimates

The financial projections provided in this section are based on **personal assumptions and general benchmarks** derived from similar technological projects and market behaviors in emerging economies. Due to the absence of a formal market study, these estimates are intended to illustrate the potential economic viability of the proposed system rather than to serve as precise financial forecasts.

Key assumptions include:

- Estimated development costs are drawn from average freelance and startup pricing in the Algerian tech ecosystem.
- Revenue projections are modeled on plausible adoption scenarios within institutions such as schools, universities, and healthcare centers.
- Personnel costs reflect a gradual team expansion over three years, assuming modest salary growth.
- The pricing strategy for subscriptions and licenses is inspired by accessible pricing practices in socially-driven tech solutions.

While these figures are speculative, they provide a structured foundation for understanding the potential business model and preparing for future validation through more detailed feasibility studies or pilot deployments.

5.6 Risk Analysis

5.6.1 Technical Risks

The **MG-BERT** model may not be fully adaptable to the Algerian dataset due to differences in language structure, dialects, or domain-specific terminology. To mitigate this, the model should be fine-tuned or enhanced using localized data to improve its performance and relevance.

5.6.2 Social Risks

- **Cultural Resistance**: Preference for human interpreters. Mitigation: Co-design workshops with FAS.
- **Digital Literacy**: Elderly users may struggle with UI. Mitigation: Onboarding tutorials in LSAI.

5.6.3 Financial Risks

- **ANSEJ Loan Default**: 30% risk if adoption <50 users/year. Contingency: Crowdfunding via Jisr Platform.

Risk	Likelihood	Impact	Mitigation
Hardware Failure	Medium	High	2-year warranty clauses
Data Privacy Laws	Low	Critical	On-device processing

Table 5.15: Risk Assessment Matrix

5.7 Deployment and Validation

5.7.1 User Testing Protocol

User testing will be conducted in the upcoming phase of the project. The protocol is as follows:

- **Planned Participants**: 50 individuals — 30 deaf and 20 hearing — will be recruited through FAS to ensure a balanced representation of urban and rural users.
- **Metrics to Be Evaluated**: The system will be assessed based on accuracy (expected: 97% for LSAl gestures), latency (target: <100ms), and usability (expected average score: 4.5/5).
- **Feedback Approach**: Bi-weekly focus groups will be organized to collect participant input and iteratively refine gesture recognition performance.
- **Anticipated Participant Demographics**: The sign language testing cohort is expected to include 50% male and 50% female participants; 70% aged between 18–35, 20% between 36–50, and 10% over 50. Additionally, 80% of participants will come from urban areas and 20% from rural regions, ensuring diversity in gesture styles and signing habits.

The evaluation approach consists of the following elements:

- **Questionnaires:** Structured forms to assess ease of use, responsiveness, interface clarity, and perceived usefulness of the application.
- **Interviews:** Semi-structured interviews with Deaf users to gather qualitative insights on their experience, expectations, and difficulties.

- **Observation:** Direct observation of users interacting with the system to analyze usage patterns, facial expressions, confusion points, and gestures.
- **Usability metrics:** Quantitative measurements such as task completion time, error rates, and number of interactions required to complete tasks (e.g., translating a phrase).

This user-centered evaluation methodology will guide the iterative improvement of the system and ensure that the application meets real-world communication needs. The feedback loop established by these methods reinforces the project's human-centered approach and long-term sustainability.

5.7.2 Roadmap Enhancements

Post-launch priorities include:

- Integration of regional variants (Kabyle sign dialects by Q3 2025)
- Smart glasses optimization for industrial applications
- AI tutor functionality for sign language learning

5.8 S.E.N.S: Silence, Écoute et Nouvelle Sensation

S.E.N.S, which stands for *Silence, Écoute et Nouvelle Sensation*, is more than just a project title, it embodies the vision and emotion behind this innovation. Rooted in the values of inclusion, empathy, and accessibility, S.E.N.S aims to bridge the communication gap between Deaf and hearing individuals by transforming silent gestures into meaningful interactions.

For a detailed explanation of the system, please refer to Section 4.6.

General Conclusion

The history of sign language, marked by struggles, imposed silences, and resilience, reminds us that it is not merely a tool for communication but a carrier of identity, culture, and dignity for millions of deaf people worldwide. While oralism has long sought to erase this natural language, today's technological advances offer us a unique opportunity: not to replace sign language, but to build bridges. Bridges between two worlds that too often look at each other without truly understanding one another.

This thesis fits into this dynamic. It has presented the development of an intelligent system based on deep learning, designed to facilitate two-way communication between deaf and hearing users through gestural language. Starting from a historical, social, and technical exploration of the context, we demonstrated how communication barriers persist in critical areas such as education, healthcare, and employment.

Three architectures were proposed and analyzed: CNN-LSTM, MediaPipe-BiLSTM, and MediaPipe-GCN-BERT. Each brought different perspectives depending on the type of gestures (isolated or continuous) to be recognized. While the MediaPipe-LSTM model achieved a remarkable accuracy of over 98%, its limitations in handling long sequences led us to propose a more robust approach combining Graph Convolutional Networks (GCN) with BERT. This latter model proved better suited to managing the complexity of real-time gestural sequences, taking into account both the spatial structure of gestures and the overall semantic context.

A mobile application was developed to implement these solutions concretely, integrating voice recognition, gesture generation via a 3D avatar, and an intuitive interface. Experimental evaluations confirmed the system's relevance, adaptability in real-world conditions, and potential for extension to other linguistic and cultural contexts.

This work thus goes beyond purely technical boundaries. It proposes a human-centered innovation that places technology at the service of accessibility, inclusion, and social justice. Future perspectives include enriching the system with facial and non-manual body expressions, integrating regional dialects of sign language, as well as generalizing it to other languages and cultures.

Ultimately, this thesis affirms a conviction: the most powerful technology is the one that knows how to listen to silent needs.

Bibliography

[1] Lisa I. Iezzoni, Bonnie L. O'Day, Mary Killeen, and Heather Harker. Communicating about health care: Observations from persons who are deaf or hard of hearing. *Annals of Internal Medicine*, 140(5):356–362, 2004. doi: 10.7326/0003-4819-140-5-200403020-00011. URL https://www.acpjournals.org/doi/10.7326/0003-4819-140-5-200403020-00011.

[2] World Health Organization. Deafness and hearing loss, 2018. URL https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss. Accessed: March 8, 2025.

[3] William Stokoe. *Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf*. Studies in Linguistics, 1960. URL https://doi.org/10.1093/deafed/eni001.

[4] Necati Cihan Camgoz et al. Sign language transformers: Joint end-to-end sign language recognition and translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033, 2020. URL https://doi.org/10.48550/arXiv.2003.13830.

[5] Harlan Lane. *When the Mind Hears: A History of the Deaf*. Random House, 1984. URL https://www.amazon.com/When-Mind-Hears-History-Deaf/dp/0679720235.

[6] United Nations. Projected number of deaf people by 2050. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss, 2018. Prediction of the United Nations, cited in WHO statistics.

[7] Brian H. Greenwald and Joseph J. Murray, editors. *In Our Own Hands: Essays in Deaf History, 1780–1970*. Gallaudet University Press, Washington, DC, 2016.

[8] American School for the Deaf. Campus aerial view, 2023. URL https://www.asd-1817.org/gallery. Photograph.

[9] Douglas C. Baynton. *Forbidden Signs: American Culture and the Campaign Against Sign Language*. University of Chicago Press, 1996.

[10] Carol Padden and Tom Humphries. *Deaf in America: Voices from a Culture*. Harvard University Press, 1988. URL https://www.amazon.com/Deaf-America-Culture-Carol-Padden/dp/0674194241.

[11] Richard G. Brill. *The Conference of Educational Administrators Serving the Deaf: A History*. pbk. Gallaudet College Press, 1986. Book size: 23 cm.

[12] Ashok K Sahoo, Gouri Sankar Mishra, and Kiran Kumar Ravulakollu. Sign language recognition: State of the art. *ARPN Journal of Engineering and Applied Sciences*, 9 (2):116–126, 2014. URL https://www.researchgate.net/publication/262187093_Sign_language_recognition_State_of_the_art. Accessed: 20 February 2025.

[13] Qutaishat Munib, Moussa Habeeb, Bayan Takruri, and Hiba Abed Al-Malik. American sign language (asl) recognition based on hough transform and neural networks. *Expert Systems with Applications*, 32(1):24–37, 2007. doi: 10.1016/j.eswa.2005.11.018. URL https://www.elsevier.com/locate/eswa. Accessed: 20 February 2025.

[14] Rodrigo Sousa de Miranda, Carla Oliveira Shubert, and Wiliam César Alves Machado. Communication with people with hearing disabilities: an integrative review. *Revista de Pesquisa: Cuidado é Fundamental Online*, 6(4):1695–1706, 2014. doi: 10.9789/2175-5361.2014.v6i4.1695-1706. URL https://doi.org/10.9789/2175-5361.2014.v6i4.1695-1706.

[15] Pamela Luft. Communication barriers for deaf employees: Needs assessment and problem-solving strategies. *Work*, 14:51–59, 2000. ISSN 1051-9815. doi: 10.3233/WOR-2000-00070. URL https://doi.org/10.3233/WOR-2000-00070.

[16] John Ngugi, Wang Yang, Jiang Wei, and Li Deyou. Sign language translator system using computer vision and lstm neural networks. *ResearchGate*, July 2024. URL https://www.researchgate.net/publication/381887623_Sign_Language_Translator_System_Using_Computer_Vision_And_LSTM_Neural_Networks.

[17] Shivashankara S and Srinath S. American sign language recognition system: An optimal approach. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, 10(8):18–30, 2018. URL https://doi.org/10.5815/ijigsp.2018.08.03.

[18] League of Arab States (LAS), Cultural Arab League Educational, and Scientific Organization (ALECSO). *First part of the Unified Arabic Sign Dictionary*. ALECSO, 1999.

[19] British-Sign.co.uk. What is british sign language?, n.d. URL https://www.british-sign.co.uk/what-is-british-sign-language/. Accessed: 2024-12-08.

[20] Boban Joksimoski, Eftim Zdravevski, Petre Lameski, Ivan Miguel Pires, Francisco José Melero, Tomás Puebla Martinez, Nuno M. Garcia, Martin Mihajlov, Ivan Chorbev, and Vladimir Trajkovik. Technological solutions for sign language recognition: A scoping review of research trends, challenges, and opportunities. *IEEE Access*, 10:40979–40995, 2022. doi: 10.1109/ACCESS.2022.3161440. URL https://doi.org/10.1109/ACCESS.2022.3161440.

[21] Ruben San-Segundo, Juan Manuel Montero, Ruben Córdoba, Javier Ferreiros, and José Manuel Pardo. A spanish speech to sign language translation system for assisting deaf-mute people. *Speech Communication*, 50(11-12):1009–1020, 2008. doi: 10.1016/j.specom.2008.02.001. URL https://doi.org/10.1016/j.specom.2008.02.001.

[22] Hamzah Luqman and Sabri A Mahmoud. Automatic translation of arabic text-to-arabic sign language. *Universal Access in the Information Society*, 18(4):939–951, 2019. doi: 10.1007/s10209-018-0622-8. URL https://doi.org/10.1007/s10209-018-0622-8.

[23] Li Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. A machine translation system from english to american sign language. In *Proceedings of the 4th Conference of the Association of Machine Translation*, pages 293–300, 2000.

[24] Tony Veale, Anthony Conway, and Briony Collins. The challenges of cross-modal translation: English-to-sign-language translation in the zardoz system. *Machine Translation*, 13(1):81–106, 1998. doi: 10.1023/A:1008014420317. URL https://doi.org/10.1023/A:1008014420317.

[25] Suphattharachai Dangsaart, Kanlaya Naruedomkul, Nick Cercone, and Booncharoen Sirinaovakul. Intelligent thai text–thai sign translation for language learning. *Computers & Education*, 51(3):1125–1141, 2008. doi: 10.1016/j.compedu.2007.11.008. URL https://doi.org/10.1016/j.compedu.2007.11.008.

[26] Jordi Porta, Fernando López-Colino, and José Colás. A rule-based translation from written spanish to spanish sign language glosses. *Computer Speech & Language*, 28(3):788–811, 2014. doi: 10.1016/j.csl.2013.10.003. URL https://doi.org/10.1016/j.csl.2013.10.003.

[27] Kinda Al-Fityani and Carol Padden. Sign language geography in the arab world. In *Sign Languages: A Cambridge Survey*, pages 433–450. Cambridge University Press, 2010. ISBN 9780521886798.

[28] Ala addin I. Sidig, Hamzah Luqman, and Sabri A. Mahmoud. Transform-based arabic sign language recognition. In *Procedia Computer Science*, volume 117, pages 2–9. Elsevier, 2017. doi: 10.1016/j.procs.2017.10.087.

[29] Ala addin I. Sidig, Hamzah Luqman, and Sabri A. Mahmoud. Arabic sign language recognition using optical flow-based features and hmm. In *Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017)*, pages 297–305. Springer, 2018.

[30] M. Al-Rousan, K. Assaleh, and A. Tala'a. Video-based signer-independent arabic sign language recognition using hidden markov models. *Applied Soft Computing*, 9(3):990–999, 2009. doi: 10.1016/j.asoc.2009.01.002. URL https://doi.org/10.1016/j.asoc.2009.01.002.

[31] Ghazanfar Latif, Nazeeruddin Mohammad, Jaafar Alghazo, Roaa AlKhalaf, and Rawan AlKhalaf. Arasl: Arabic alphabets sign language dataset. *Journal of Computer Science and Technology*, 34(2):123–136, 2019. doi: 10.1016/j.dib.2019.103777. URL https://www.sciencedirect.com/science/article/pii/S2352340919301087.

[32] Jordan J. Bird, Anikó Ekárt, and Diego R. Faria. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors*, 20(18):1–20, 2020. doi: 10.3390/s20185158.

[33] National Institute on Deafness and Other Communication Disorders. What is american sign language?, 2021. URL https://www.nidcd.nih.gov/health/american-sign-language#1. Accessed: 2024-12-08.

[34] M. Quinn and J.I. Olszewska. British sign language recognition in the wild based on multi-class svm. In *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 81–86, 2019. doi: 10.15439/2019F274.

[35] Luis Serrano. *Grokking Machine Learning*. Manning Publications, 2021. ISBN 9781617295911. URL https://www.manning.com/books/grokking-machine-learning. Includes PDF/eBook formats via Manning subscription.

[36] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7793, 2018. doi: 10.1109/CVPR.2018.00812. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Camgoz_Neural_Sign_Language_CVPR_2018_paper.html.

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. MIT Press, 2017. URL https://www.deeplearningbook.org/.

[38] Maria Papatsimouli, Panos Sarigiannidis, and George F. Fragulis. A survey of advancements in real-time sign language translators: Integration with iot technology. *Technologies*, 11(4):83, 2023. doi: 10.3390/technologies11040083. URL https://www.mdpi.com/2227-7080/11/4/83.

[39] Vraj Shah et al. Natural language processing. *International Journal of Computer Sciences and Engineering*, 6(1):161–167, 2018. doi: 10.26438/ijcse/v6i1.161167. URL https://www.ijcseonline.org/pdf_paper_view.php?paper_id=2010.

[40] Sampada S. Wazalwar and Urmila Shrawankar. Interpretation of sign language into english using nlp techniques. *Journal of Information and Optimization Sciences*, 38(6):895–910, 2017. doi: 10.1080/02522667.2017.1372136. URL https://www.tandfonline.com/doi/abs/10.1080/02522667.2017.1372136.

[41] M. Madhiarasan and Partha Pratim Roy. A comprehensive review of sign language recognition: Different types, modalities, and datasets. *Journal of LaTeX Class Files*, XX (X), April 2022. doi: 10.48550/ARXIV.2204.03328. URL https://arxiv.org/abs/2204.03328.

[42] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3d action recognition with random occupancy patterns. In *Computer vision–ECCV 2012*, pages 872–885. Springer, 2012. URL https://doi.org/10.1007/978-3-642-33709-3_62.

[43] Xiaohui Shen, Gang Hua, Lance Williams, and Ying Wu. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image and Vision Computing*, 30(3):227–235, 2012. doi: 10.1016/j.imavis.2011.11.003. URL https://doi.org/10.1016/j.imavis.2011.11.003.

[44] Amir Tamrakar, Saad Ali, Qian Yu, Jingen Liu, Omar Javed, Ajay Divakaran, Hui Cheng, and Harpreet Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3681–3688. IEEE, 2012.

[45] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014. doi: 10.1109/TITS.2014.2337331. URL https://doi.org/10.1109/TITS.2014.2337331.

[46] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017.

[47] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. doi: 10.1109/ICCV.2015.510.

[48] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[49] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human-computer interaction: a survey. *Artificial Intelligence Review*, 43:1–54, 2015. doi: 10.1007/s10462-012-9356-9.

[50] Ilias Papastratis, Christos Chatzikonstantinou, Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. Artificial intelligence technologies for sign language. *Sensors*, 21(17):5843, 2021. doi: 10.3390/s21175843. URL https://doi.org/10.3390/s21175843.

[51] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135.

[52] Ilias Papastratis, Christos Chatzikonstantinou, Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. Artificial intelligence technologies for sign language. *Sensors*, 21(17):5843, 2021. doi: 10.3390/s21175843. URL https://www.mdpi.com/1424-8220/21/17/5843.

[53] I. Papastratis, K. Dimitropoulos, and P. Daras. Continuous sign language recognition through a context-aware generative adversarial network. *Sensors*, 21:2437, 2021. doi: 10.3390/s21072437.

[54] Parteek Bhatia and Ankita Wadhawan. Sign language recognition systems: A decade systematic literature review. *Archives of Computational Methods in Engineering*, 28:785–813, 2021. doi: 10.1007/s11831-019-09384-2.

[55] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–12, 2016. URL http://dx.doi.org/10.5244/C.30.136.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. URL https://doi.org/10.48550/arXiv.1706.03762.

[57] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, et al. Mediapipe: A framework for perceiving and processing reality. In *CVPR Workshop*, 2019.

[58] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. URL https://arxiv.org/abs/2006.10214.

[59] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, third edition, 2022.

[60] Ibomoiye Domor Mienye, Theo G. Swart, and George Obaido. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(517), 2024. doi: 10.3390/info15090517. URL https://doi.org/10.3390/info15090517.

[61] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: A comprehensive review. *Computational Social Networks*, 6(11), 2019. doi: 10.1186/s40649-019-0069-y. URL https://doi.org/10.1186/s40649-019-0069-y.

[62] Cleison Correia de Amorim, David Macêdo, and Cleber Zanchettin. Spatial-temporal graph convolutional networks for sign language recognition. *arXiv preprint arXiv:1901.11164*, 2020. doi: 10.48550/arXiv.1901.11164. URL https://arxiv.org/abs/1901.11164.

[63] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2020. URL https://doi.org/10.48550/arXiv.1906.08172.

[64] Google AI. Mediapipe image segmenter, 2024. URL https://ai.google.dev/edge/mediapipe/solutions/vision/image_segmenter. Accessed: 2024-03-26.

[65] Google AI. Mediapipe interactive image segmenter, 2024. URL https://ai.google.dev/edge/mediapipe/solutions/vision/interactive_segmenter. Accessed: 2024-03-26.

[66] Chraa Mesbahi Soukaina, Masrour Mohammed, and Rhazzaf Mohamed. Geometric feature-based machine learning for efficient hand sign gesture recognition. *Statistics, Optimization and Information Computing*, x:0–16, 202x. URL iapress.org/index.php/soic/article/view/2306.

[67] Subhalaxmi Chakraborty, Swatilekha Banerjee, Nanak Bandyopadhyay, Zinnia Sarkar, Piyal Chakraverty, and Sweta Ghosh. Indian sign language classification (isl) using machine learning. *American Journal of Electronics & Communication*, 1(3):17–21, 2021.

[68] Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. A deep learning approach for analyzing video and skeletal features in sign language recognition. In *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6, 2018.

[69] Di Wu and Ling Shao. Multimodal dynamic networks for gesture recognition. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 945–948, 2014.

[70] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid cnn-hmms. *International Journal of Computer Vision*, 126:1311–1325, 2018. doi: 10.1007/s11263-018-1121-3.

[71] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. *arXiv preprint arXiv:1603.08271*, 2016. doi: 10.1109/CVPR.2017.175. URL https://arxiv.org/abs/1603.08271. Department of Automation, Tsinghua University, State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China.

[72] Fatma M. Najib. Sign language interpretation using machine learning and artificial intelligence. *Neural Computing and Applications*, 37:841–857, 2024. doi: 10.1007/s00521-024-10395-9. URL https://doi.org/10.1007/s00521-024-10395-9.

[73] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017. doi: 10.48550/arXiv.1609.02907. URL https://arxiv.org/abs/1609.02907.

[74] Maria Parelli, Katerina Papadimitriou, Gerasimos Potamianos, Georgios Pavlakos, and Petros Maragos. Spatio-temporal graph convolutional networks for continuous sign language recognition. In *ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8457–8461, 2022. doi: 10.1109/ICASSP43922.2022.9746971. URL https://doi.org/10.1109/ICASSP43922.2022.9746971.

[75] Katerina Papadimitriou and Gerasimos Potamianos. Sign language recognition via deformable 3d convolutions and modulated graph convolutional networks. In *ICASSP 2023 - IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

[76] Neelma Naz, Hasan Sajid, Sara Ali, Osman Hasan, and Muhammad Khurram Ehsan. Signgraph: An efficient and accurate pose-based graph convolution approach toward sign language recognition. *IEEE Access*, 11:19135–19150, 2023.

[77] Lu Meng and Ronghui Li. An attention-enhanced multi-scale and dual sign language recognition network based on a graph convolution network. *Sensors*, 21(4):1120, 2021.

[78] Yang Zhou, Zhaoyang Xia, Yuxiao Chen, Carol Neidle, and Dimitris N. Metaxas. A multimodal spatio-temporal gcn model with enhancements for isolated sign recognition. In *LREC-COLING 2024 - Workshop on the Representation and Processing of Sign Languages*, 2024.

[79] Juan Song, Huixuechun Wang, Jianan Li, Jian Zheng, Zhifu Zhao, and Qingshan Li. Hand-aware graph convolution network for skeleton-based sign language recognition. *Journal of Information and Intelligence*, 3:36–50, 2025. URL https://doi.org/10.1016/j.jiixd.2024.08.001.

[80] Safaeid Hossain Arib, Rabeya Akter, Sejuti Rahman, and Shafin Rahman. Signformer-gcn: Continuous sign language translation using spatio-temporal graph convolutional networks. *PLOS ONE*, 20(2):e0316298, 2025. URL https://doi.org/10.1371/journal.pone.0316298.

[81] Author(s) of the article. Rnn-lstm: From applications to modeling techniques and beyond—systematic review. *Journal Name*, XX:XX–XX, 2024. doi: 10.1016/j.jksuci.2024.102068.

[82] B. Fang, J. Co, and M. Zhang. Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In *Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*, pages 1–13, 2017.

[83] D. C. Kavarthapu and K. Mitra. Hand gesture sequence recognition using inertial motion units (imus). In *Proceedings of the 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 953–957, Nov 2017.

[84] E. Rakun, A. M. Arymurthy, L. Y. Stefanus, A. F. Wicaksono, and I. W. W. Wisesa. Recognition of sign language system for indonesian language using long short-term memory neural networks. *Advanced Science Letters*, 24(2):999–1004, Feb 2018.

[85] S. S. Kumar, T. Wangyal, V. Saboo, and R. Srinath. Time series neural networks for real time sign language translation. In *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 243–248, Dec 2018.

[86] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019. URL https://doi.org/10.48550/arXiv.1810.04805.

[87] Zhenxing Zhou, Vincent W. L. Tam, and Edmund Y. Lam. Signbert: A bert-based deep learning framework for continuous sign language recognition. *IEEE Access*, 9:161669–161685, 2021. doi: 10.1109/ACCESS.2021.3132668.

[88] Anirudh Tunga, Sai Vidyaranya Nuthalapati, and Juan Wachs. Pose-based sign language recognition using gcn and bert. In *WACV Workshops*, pages 1–10. IEEE, 2021. URL https://doi.org/10.48550/arXiv.2012.00781.

[89] Zhenxing Zhou, Vincent W. L. Tam, and Edmund Y. Lam. A cross-attention bert-based framework for continuous sign language recognition. *IEEE Signal Processing Letters*, 29:1818–1822, 2022. doi: https://doi.org/10.1109/LSP.2022.3199665.

[90] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *Proceedings of the LREC*, pages 1911–1916, Reykjavik, Iceland, 2014.

[91] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, Salt Lake City, UT, USA, 2018.

[92] Samuel Albanie, Gul Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision*, pages 35–53, Cham, Switzerland, 2020. Springer.

[93] Jie Huang, Wei Zhou, Qiang Zhang, Haifeng Li, and Wei Li. Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, New Orleans, LA, USA, 2018.

[94] Jie Pu, Wei Zhou, and Haifeng Li. Sign language recognition with multi-modal features. In *Pacific Rim Conference on Multimedia*, pages 252–261, Cham, Switzerland, 2016. Springer.

[95] J. Huang, W. Zhou, H. Li, and W. Li. Video-based sign language recognition without temporal segmentation. 2018. doi: 10.1609/aaai.v32i1.11903. URL https://doi.org/10.1609/aaai.v32i1.11903. Accessed: 2025-02-28.

[96] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li. Chinese sign language recognition with adaptive hmm. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2016. Accessed: 2025-02-28.

[97] H. R. V. Joze and O. Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint*, 2018. URL https://arxiv.org/abs/1812.01053. Accessed: 2025-02-28.

[98] Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. *arXiv preprint arXiv:1910.11006*, 2020. doi: 1910.11006. URL https://arxiv.org/abs/1910.11006.

[99] Author(s). Slowfast network for continuous sign language recognition. *Journal/Conference Name*, Volume (if applicable)(Issue (if applicable)):Page numbers (if applicable), Year. doi: DOIorURL.

[100] Author(s). Two-stream network for sign language recognition and translation. *Journal/Conference Name*, Volume (if applicable)(Issue (if applicable)):Page numbers (if applicable), Year. doi: DOIorURL.

[101] Author(s). Tcnet: Continuous sign language recognition from trajectories and correlated regions. *Journal/Conference Name*, Volume (if applicable)(Issue (if applicable)): Page numbers (if applicable), Year. doi: DOIorURL.

[102] Zecheng Li, Wengang Zhou, Weichao Zhao, Kepeng Wu, Hezhen Hu, and Houqiang Li. Uni-sign: Toward unified sign language understanding at scale. *arXiv preprint arXiv:2501.15187*, 2025. doi: 2501.15187. URL https://arxiv.org/abs/2501.15187.

[103] Ronglai Zuo, Fangyun Wei, and Brian Mak. Natural language-assisted sign language recognition. *arXiv preprint arXiv:2303.12080*, 2023. doi: 2303.12080. URL https://arxiv.org/abs/2303.12080.

[104] Xiaolong Shen, Zhedong Zheng, and Yi Yang. Stepnet: Spatial-temporal part-aware network for isolated sign language recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 20(1):1–20, 2024. doi: 10.1145/3656046. URL https://dl.acm.org/doi/10.1145/3656046.

[105] Tangfei Tao, Yizhe Zhao, Tianyu Liu, and Jieli Zhu. Sign language recognition: A comprehensive review of traditional and deep learning approaches, datasets, and challenges. *IEEE Access*, 12, 2024. doi: 10.1109/ACCESS.2024.3398806. URL https://doi.org/10.1109/ACCESS.2024.3398806. Accessed: 21 Feb. 2025.

[106] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Post-Conference Workshop of ACL 2004*, pages 74–81, 2004. URL https://aclanthology.org/W04-1013. Accessed: 21 Feb. 2025.

[107] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. doi: 10.1016/S0031-3203(96)00142-2.

[108] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

[109] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1143, 1995.

[110] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, O. Kudriavtsev, J. Levenberg, D. Mané, M. Monga, S. Moore, D. Murray, C. Olah, D. Shlens, B. Steiner, I. Sutskever, P. Talwar, P. Tucker, V. Vanhoucke, and V. Vassilvitskii. Tensorflow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016. URL https://doi.org/10.48550/arXiv.1605.08695.

[111] Gary Bradski and Adrian Kaehler. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2016.

[112] Google Research. Mediapipe: A framework for building perception pipelines. *arXiv:1906.08172*, 2019. URL https://arxiv.org/abs/1906.08172.

[113] Travis E. Oliphant. *Guide to NumPy*. Trelgol Publishing, 2006. ISBN 978-1-775-10782-2. URL https://numpy.org/.

[114] Fabian Pedregosa, Gergö Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Mathieu Brucher, Mark Perrot, and Édouard Duchesnay. *Scikit-learn: Machine Learning in Python*, volume 12. 2011. URL https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[115] Antonio Garcia, Claudio Rosso, and Francesco Maria Panella. Spektral: A python library for graph deep learning. *arXiv:1905.02411*, 2019. URL https://arxiv.org/abs/1905.02411.

[116] John D. Hunter. *Matplotlib: A 2D Graphics Environment*, volume 9. 2007. URL https://doi.org/10.1109/MCSE.2007.55.

[117] Blender Foundation. Blender: a 3d creation suite, 2020. URL https://www.blender.org/.

[118] Facebook. React native: A framework for building native apps using javascript and react, 2015. URL https://reactnative.dev/.

[119] Expo. Expo: The framework for universal react apps, 2020. URL https://expo.dev/.

[120] Guillermo Sánchez-Brizuela, Ana Cisnal, Eusebio de la Fuente-López, Juan-Carlos Fraile, and Javier Pérez-Turiel. Lightweight real-time hand segmentation leveraging mediapipe landmark detection. *Virtual Reality*, 27:3125–3132, 2023. doi: 10.1007/s10055-023-00858-0. URL https://doi.org/10.1007/s10055-023-00858-0.

[121] Juan Rodríguez-Correa et al. Assistive technologies for deaf communities: A systematic review. *Frontiers in Education*, 8:1–10, 2023. doi: 10.3389/feduc.2023.1121597. URL https://www.frontiersin.org/articles/10.3389/feduc.2023.1121597/full.

[122] Ministère de la Solidarité Nationale, de la Famille et de la Condition de la Femme. Rapport annuel sur le handicap en algérie, 2022.

[123] Office National des Statistiques. Enquête nationale sur le handicap, 2021. URL https://www.ons.dz/.

[124] A. Belkacem and N. Benghabrit. Accessibility of assistive technologies in algeria: Challenges and opportunities. https://e-inclusion.unescwa.org/book/1567, 2020.