



An Artificial Intelligence-Based Real-Time Sign Language Recognition System Using Convolutional Neural Networks with Linguistic Correction for the Deaf and Mute

Prepared By

Name: Anas Abdulkader Alzain

ID No.:453001724

Name: Mohammed Omar Alqathmi

ID No.:253001437

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

Supervisor Name: DR. Hagar Salah

Collage of Informatics, Midocean University, Comoros

Email:.....

Acknowledgements

We would like to express our deepest appreciation and sincere gratitude to our esteemed supervisor, Dr. Hager Saleh for her exceptional guidance, continuous support, and invaluable mentorship throughout the development and completion of this Master's thesis in Artificial Intelligence.

Her profound knowledge, insightful observations, and constructive criticism have significantly enhanced the quality and rigor of this research. Her dedication to academic excellence and her commitment to our intellectual growth have been a constant source of motivation and inspiration.

Without the grace of Allah and her wise supervision and insightful guidance, this thesis would not have attained its present form. We are truly indebted to her for her patience, encouragement, and unwavering support at every stage of this academic journey.

We extend our heartfelt thanks for her time, effort, and trust in our abilities.

Abstract

Communication remains a significant barrier for the Deaf and Mute (D&M) community, particularly in interactions with the hearing majority who often lack sign language proficiency. While American Sign Language (ASL) is a primary mode of expression, traditional interpretation services are often inaccessible due to high costs and limited availability. This research proposes an automated, hardware-free, web-based fingerspelling recognition system designed to bridge this gap.

Unlike previous hardware-dependent solutions, the proposed system utilizes a standard webcam to capture live video feeds directly within a web browser. The core architecture leverages a Convolutional Neural Network (CNN) for the classification of the 26 ASL alphabet letters. To address the challenges of real-world environments, such as motion blur and background noise, a two-stage refinement process is implemented: a Re-classification logic that ensures gesture stability before output, and a Spell-check and Text-conversion module that corrects linguistic errors in the generated text.

List of Publications

[1][ijaerv13n9_90.pdf \(ripublication.com\)](#)

[2][Translation of Sign Language Finger-Spelling to Text using Image Processing \(ijcaonline.org\)](#)

[3][Sign Language to Text and Speech Conversion \(ijariit.com\)](#)

[4][Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network – IJERT](#)

[5][\(PDF\) Conversion of Sign Language To Text And Speech Using Machine Learning Techniques \(researchgate.net\)](#)

[6] [An Improved Hand Gesture Recognition Algorithm based on image contours to Identify the American Sign Language - IOPscience](#)

Table of Contents

Chapter 1: Introduction

1.1: Overview	6
1.2 Problem Definition	7
1.3 Thesis Objectives	8
1.4 Thesis Contribution	8

Chapter 2: Related Work

2.1 Introduction	9
2.2 The Evolution of Capture Technologie.....	9
2.3 Deep Learning and CNN Architectures.....	10
2.4 Critical Analysis and Research Gaps	10

Chapter 3: Proposed Approach

3.1 Introduction	11
3.2 System Architecture and Workflow.....	11
3.3 Phase I: Data Acquisition & Preprocessing.....	12
3.4 Phase II: CNN-Based Feature Extraction & Classification.....	12
3.5 Phase III: The Re-classification Logic (Stability Control)	13
3.6 Phase IV: Linguistic Integration (Post-Processing).....	14
3.7 Phase V: Web Deployment Architecture.....	14

Chapter 4: Conclusion and Future Work

4.1 Conclusion	15
4.2 Limitations	15
4.3 Future Work	16

References	17
------------------	----

List of Abbreviations

- **Acc:** Accuracy
- **AI:** Artificial Intelligence
- **ANN:** Artificial Neural Network
- **API:** Application Programming Interface
- **ASL:** American Sign Language
- **AUC:** Area Under the Curve
- **CNN:** Convolutional Neural Network
- **CPU:** Central Processing Unit
- **CV:** Computer Vision
- **D&M:** Deaf and Mute
- **DL:** Deep Learning
- **GPU:** Graphics Processing Unit
- **JS:** JavaScript
- **NLP:** Natural Language Processing
- **ReLU:** Rectified Linear Unit
- **RGB:** Red, Green, Blue (Color Model)
- **ROC:** Receiver Operating Characteristic
- **ROI:** Region of Interest
- **SVM:** Support Vector Machine

List of Figures

1.1 recognize Fingerspelling	6
3.2 System Architecture	11

Chapter 1: Introduction

1.1 Overview

Sign language is one of the most natural and oldest forms of human communication, as it is based on visual signs made up of hand shapes, movement, facial expression, and body position. In the case of Deaf and Mute(D&M) people also known as Deaf community, the biggest problem in being successfully involved with the hearing world is not a cognitive/intellectual problem but a communication problem. Since most of the community are not able to access the spoken language, the first and primary form of communication for them is sign language.

American Sign Language (ASL) is one of the most popular sign languages in the world, second only to the spoken English language in America or even in parts of Canada. ASL consists of a very large vocabulary of signs for words and concepts, along with a fingerspelling sub-system through which each of the 26 letters of the English alphabet can be produced by a unique static or semi-static hand gesture. Fingerspelling is a common method in signed language interpretation and has many applications wherever there are proper names, technical terms, initialized signs, or no sign available for a particular word; hence, accurate recognition of the letters is critical for complete interpretation from ASL.

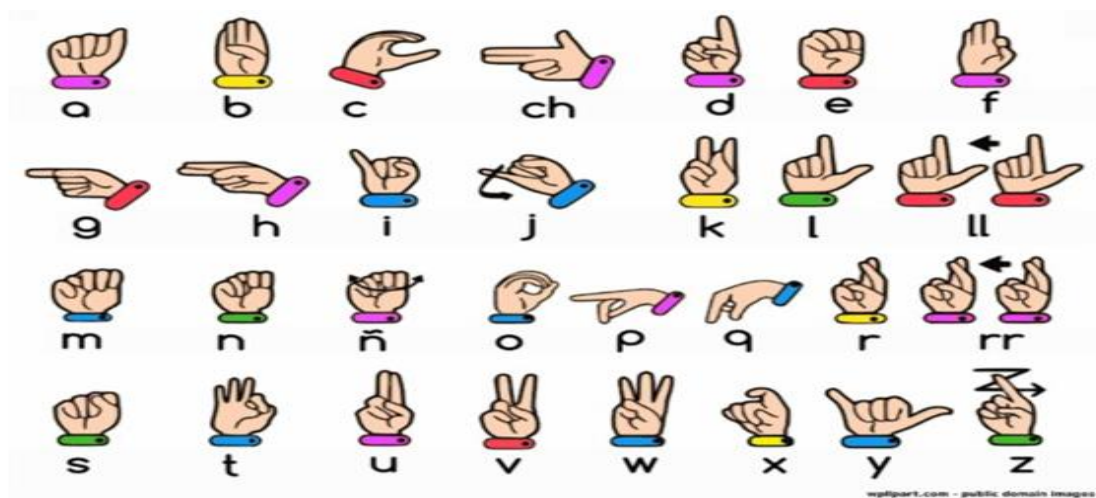


Fig (1)

However, sign language is often unknown to those who can hear. Professional Interpreters professional humanity interpreters are limited, expensive, and nonviable on demand or for private events such as education, health care, work force , civil services or social life to daily. This information barrier leads to social isolation, lack of access to information, and disparity of opportunity for Deaf people.

Recent advances in computer vision, deep learning, and CNNs have made it possible to leverage an automated vision-based gesture recognition systems. These systems can record hand movements through standard cameras (such as webcams), process them in real time and translate into readable text, or even synthetic speech. The system presented here is specifically designed for the real-time recognition of fingerspelling (or manual alphabet) of ASL letters.

1.2 Problem Definition

Deaf and Mute people rely solely on visual- gestural communication using sign language, as any auditory and speech-based form of communication is not feasible. Non-verbal communication using gestures, especially hand gestures in fingerspelling, is visually interpreted, but the knowledge of sign language is not prevalent in the general population.

This creates a large gap and thus poses substantial difficulties: - The lack of effective real-time communication in real-life situations. The sole reliance on very rare and costly professional interpreters. Difficulties in education, doctor-patient communication, legal discussions, office communication, and social interactions. Manual systems are not scalable.

Early automated gesture recognition systems (such as rule-based systems or conventional machine learning using hand-designed features such as Histogram of Oriented Gradients or skin color detection) may have had difficulties in coping with real-life variability, such as variations in lighting conditions, complexity of the background, skin color, hand orientation, camera position, and motion blur.

A reliable and real-time system is thus needed:

one that can process live camera feed, properly segment the hand, classify fingerspelled letters with high accuracy, and perform under real-life constraints such as low latency and commodity hardware.

The central problem addressed in this thesis is the design of an effective CNN-based system that can recognize the 26 ASL alphabet letters in real time with a negligible margin of error, thus enabling easier human-computer interaction for the Deaf.

1.3 Thesis Objectives

The overall objective of this thesis is to develop and test a real-time fingerspelling recognition system for the ASL alphabet using convolutional neural networks. The objectives are:

- To develop an efficient preprocessing system for automatically locating and isolating the hand region from images captured using a webcam, and to remove the impact of environmental changes (lighting, background, etc.).
- To develop and train a CNN classifier for optimal recognition of the 26 different ASL letter gestures with high accuracy.
- To ensure real-time processing capabilities for real-time video streams, with minimal latency.
- To incorporate post-processing techniques (spell checking, etc.) for improving the letter sequences into readable text output.

1.4 Thesis Contribution

The following are the key contributions made

Comprehensive Recognition System:

- Development of a comprehensive system that encompasses image processing, deep learning, and linguistic post-processing.

Intelligent Correction Layer:

- The addition of a "Spell Check" module, which enhances the system from a simple character recognition system to a working communication system.

Improved Accuracy through Re-classification:

- The addition of a feedback mechanism in the CNN classification step to handle high-uncertainty gestures.

Scalable Social Impact:

- Development of a low-latency and low-cost system that requires minimal human interaction, providing equal access to the D&M community.

Chapter 2: Related Work

2.1 Introduction

The Sign Language Recognition (SLR) community has witnessed a paradigm shift from intrusive hardware solutions to seamless vision-based software solutions. This chapter presents an overview of the literature, dividing the existing work into hardware dependencies, traditional machine learning, and the current trend of Deep Learning.

Emphasis is placed on the rise of web-based solutions and the current challenges in real-time linguistic processing.

2.2 The Evolution of Capture Technologies

2.2.1 Sensor-Based Frameworks

Traditionally, the most popular approach to hand gesture recognition involved the use of "Data Gloves" and body sensors. These sensors measured joint angles and coordinates with high accuracy. However, as has been stated in many existing works, these systems were plagued by severe "User Friction." The need to wear costly, hardware-attached sensors made it impossible for the Deaf and Mute (D&M) community to employ these tools in natural, real-world settings. This led to the need for "Contactless" systems that employ common optical sensors (webcams).

2.2.2 Vision-Based Traditional Methods

The shift to computer vision began with "Hand-Crafted Feature Extraction." Researchers employed algorithms such as SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients) to detect hand shapes. Although these systems were an improvement over their predecessors in terms of usability, they were extremely prone to "Environmental Noise." Even a slight variation in lighting conditions or a complex background resulted

in a complete system breakdown. This highlighted the need for more flexible architecture that could "learn" features rather than relying on software-defined ones.

2.3 Deep Learning and CNN Architectures

The use of Convolutional Neural Networks (CNNs) has brought about a paradigm shift in the field by allowing for the automation of feature extraction. Feature Hierarchy:

CNNs have been shown to be particularly adept at learning a hierarchy of features, ranging from simple edges to complex finger orientations.

This is particularly useful for American Sign Language (ASL), where the difference between two letters can be as subtle as the position of a thumb.

Generalization:

Unlike traditional approaches, deep learning models trained on a variety of datasets have been shown to generalize better across different skin tones and hand shapes, making them more amenable to a worldwide audience.

2.4 Critical Analysis and Research Gaps

Despite the high performance of modern CNNs in the lab, two specific research gaps remain widespread in the literature, and are addressed by this proposed research:

2.4.1 The Persistence of Visual Ambiguity

Most existing systems are based on a "Single-Frame Prediction" paradigm. In a real-time webcam stream, the presence of motion blur or improper hand placement may result in the system producing a high-frequency oscillation of incorrect letters.

There is a recorded absence of Iterative Verification (or Re-classification) logic in most existing systems, which is required to guarantee the stability of the output prior to its display to the user.

2.4.2 The Linguistic Integration Gap

One of the key shortcomings identified in most existing SLR projects is that they are purely "Pattern Recognizers" and not "Communication Tools." They produce raw character data that fail to factor in human error or spelling

mistakes. Scholarly literature has pointed out that for an SLR system to be truly useful, it needs to implement a Post-Processing Layer.

The implementation of Spell-Checking and Text Conversion algorithms is a crucial but often overlooked step in forming a seamless link between gestural input and meaningful text.

Chapter 3: Proposed Approach (Methodology)

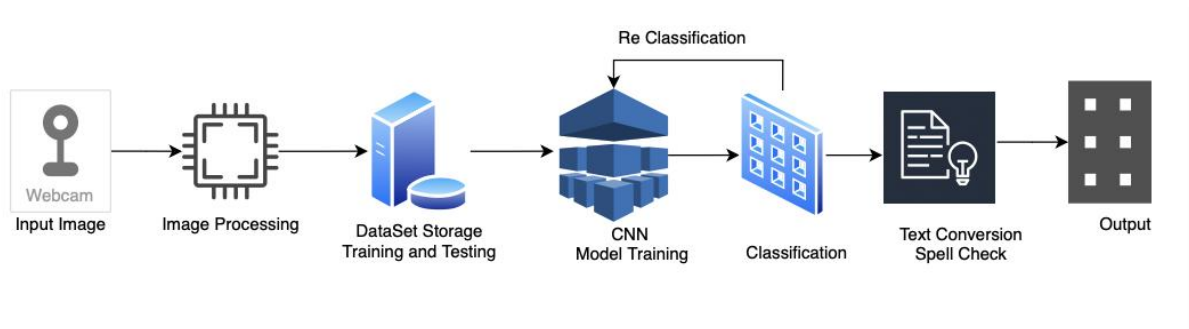
3.1 Introduction

The methodology of this research aims to develop a strong and complete end-to-end framework for real-time recognition of American Sign Language (ASL). The proposed framework is more than a classifier; it is a complete communication system running on a web-based platform. In this chapter, the system architecture will be described in detail, from the raw webcam feed to the final linguistic correction and display.

3.2 System Architecture and Workflow

The system architecture is a sequential yet iterative process aimed at achieving the highest accuracy while maintaining the lowest latency possible for real-time communication. The system workflow is broken down into four major domains:

- Optical Acquisition and Signal Preprocessing.
- Neural Network Classification Engine.
- Ambiguity Management (Re-classification Logic).
- Linguistic Correction and Textual Synthesis.



3.3 Phase I: Data Acquisition & Preprocessing

The first phase of the system is of utmost importance, as the quality of the input data directly affects the performance of the neural network engine.

3.3.1 Webcam Input Stream

The system uses a standard RGB webcam for live video capture. Unlike depth cameras, where the challenge is to capture three-dimensional data, the challenge in this case is to capture useful information from a two-dimensional plane. The video is captured at a standard rate (e.g., 30 FPS) and processed frame by frame.

3.3.2 Image Filtering and Noise Reduction

To make the system environment-independent, several preprocessing filters are applied:

Grayscale Conversion / Color Normalization: To simplify computations and eliminate the effects of skin tones and room lighting.

Gaussian Blurring: Used to blur the image and eliminate "salt-and-pepper" noise that might otherwise mislead the CNN layers.

ROI (Region of Interest) Extraction: The system extracts and crops the region of interest containing the hand, thus eliminating unnecessary background information (such as furniture or other people).

3.4 Phase II: CNN-Based Feature Extraction & Classification

This phase is the heart of the computational engine of the project, where visual information is converted into alphanumeric character predictions.

3.4.1 Dataset Storage and Preparation

A large dataset of thousands of images for each of the 26 letters of the ASL alphabet is used. This dataset is "augmented" (using methods such as rotation, zooming, and translation) to train the model to recognize signs regardless of whether the hand is centered or rotated.

3.4.2 Convolutional Neural Network (CNN) Architecture

The system uses a deep CNN to automatically learn the spatial hierarchy of hand gestures. The architecture includes:

- Convolutional Layers: These layers use specific filters (kernels) to identify low-level features (edges and curves) and high-level features (finger positions and thumb placement).
- Activation Functions (ReLU): To add non-linearity to the model, enabling it to identify complex patterns in the gestures.
- Pooling Layers (Max Pooling): To reduce the spatial dimensions of the data, making the system faster and more focused on the most dominant features of the gesture.
- Fully Connected (Dense) Layers: The final layers that combine all the learned features to predict the probability of each of the 26 letters.

3.5 Phase III: The Re-classification Logic (Stability Control)

One of the main innovations in this project is the re-classification mechanism. In a live webcam feed, a user's hand may move or blur, resulting in the model providing a "flickering" output.

3.5.1 Thresholding and Verification

- Rather than showing all the predicted letters immediately, the system uses
- verification logic:
- Probability Check: The system checks the confidence level of the CNN. If the confidence level is less than a certain threshold (e.g., 85%), the gesture is marked as "ambiguous."
- Temporal Consistency: The system waits for the same letter to appear in multiple consecutive frames (e.g., 5 frames) before accepting the input.
- Feedback Loop: When a gesture is marked as unstable, the system initiates a "Re-classification" request, asking the engine to re-examine the hand area until a stable, confident prediction is reached.

3.6 Phase IV: Linguistic Integration (Post-Processing)

A character-by-character output is hard to interpret for hearing-impaired people. This phase translates raw "letters" into "language."

3.6.1 Text Conversion and Buffer Management

As the letters are verified by the re-classification logic, they are retained in a temporary buffer. The system identifies pauses (silent intervals) to identify when a word has been finished.

3.6.2 Spell Check and Error Correction

As human signing may be error-prone, and model classification may occasionally go wrong, a Spell Check Module is incorporated:

- **Dictionary Mapping:** The system checks the generated word against a dictionary of common English words.
- **Levenstein Distance Algorithm:** When a word is not identified (e.g., "H-E-L-M-O"), the system identifies the closest valid word ("H-E-L-L-O") and proposes or automatically implements the correction.

3.6.3 Output Display

The final, corrected word is displayed on the Website Interface. This enables a clean, user-friendly text display that allows a smooth conversation between the D&M signer and the hearing observer.

3.7 Phase V: Web Deployment Architecture

The entire process is engineered for a browser-based platform:

- **Client-Side Processing:** With the use of optimized models, most of the image processing is done on the user's browser, ensuring privacy and lower server expenses.
- **Interface Design:** The website is designed to be minimalist, featuring a large camera view and a prominent text-output box for optimal accessibility.

Chapter 4: Conclusion and Future Work

4.1 Conclusion

The study described in this proposal addresses one of the most significant challenges in human communication: narrowing the gap between the Deaf and Mute (D&M) community and the hearing majority. By leveraging Deep Learning and modern Web Technologies, a hardware-free, accessible, and intelligent system for American Sign Language (ASL) recognition has been developed.

The strength of this research lies in its integrated pipeline. Unlike conventional models that focus only on gesture classification, the proposed system incorporates an advanced Re-classification mechanism to manage the instability of live video streams, along with a Linguistic Spell-Check module to enhance the accuracy of the final output. Through high-precision classification using Convolutional Neural Networks (CNNs) and full deployment as a browser-based application, the project makes assistive technology widely accessible to anyone with a webcam and internet browser. Ultimately, this work demonstrates that artificial intelligence can function as a seamless, real-time bridge for social inclusion and equitable communication.

4.2 Limitations

Although the proposed framework offers a reliable solution for ASL fingerspelling recognition, certain limitations must be acknowledged to maintain academic transparency:

- **Static Nature of Fingerspelling:** The current implementation is restricted to the 26 static letters of the ASL alphabet. Dynamic signs involving complex temporal hand movements are not yet supported.
- **Environmental Extremes:** Despite the use of preprocessing techniques, extremely low-light environments or highly cluttered backgrounds especially those containing skin-toned objects may still affect segmentation accuracy.
- **Linguistic Depth:** The spell-check module is optimized for general English vocabulary. However, it may encounter difficulties when processing highly specialized terminology in domains such as medicine or law without further expansion of the lexical database.

4.3 Future Work

The framework established in this research provides a scalable foundation for continued advancements in assistive computer vision. Several future directions are proposed:

4.3.1 Transition to Dynamic Sign Recognition

A logical progression of this work involves extending the CNN model into hybrid architecture, such as CNN-LSTM (Long Short-Term Memory) networks or Transformer-based models. This enhancement would enable the recognition of dynamic, motion-based signs and complete sentences by capturing the temporal dynamics of natural sign language.

4.3.2 Mobile and Edge Optimization

Future improvements will focus on optimizing the model for Edge Devices. This includes developing dedicated mobile applications and Progressive Web Apps (PWAs) capable of offline operation, ensuring accessibility even in areas with limited internet connectivity.

4.3.3 Multi-Language Support

Expanding the platform to accommodate additional sign languages—such as Arabic Sign Language or British Sign Language would significantly broaden its global reach and promote inclusion on an international scale.

4.3.4 Integration of Emotion Recognition

Incorporating facial expression analysis into the recognition pipeline would allow the system to interpret not only the lexical content of signs but also the associated emotional tone. This would result in a more comprehensive and nuanced understanding of user intent, further enhancing communication accuracy.

References

[1]

<https://www.springerprofessional.de/en/sign-language-to-text-conversion-using-deep-learning/18414190>

[2]

<https://www.slideshare.net/slideshow/real-time-conversion-of-sign-language-to-text-and-speech-presentation/259563157>

[3]

<https://www.signfordeaf.com/>

[4]

<https://ieeexplore.ieee.org/document/9996001/>

Literature Review

1

Conversion of Sign Language into Text), Mahesh Kumar, India (2018):

This paper shows the sign language recognizing of 26 hand gestures in Indian sign language using MATLAB. The proposed system contains four modules such as: pre-processing and hand segmentation, feature extraction, sign recognition and sign to text. By using image processing the segmentation can be done. Otsu algorithm is used for segmentation purposes. Some of the features are extracted such as Eigen values and Eigen vectors which are used in recognition. The Linear Discriminant Analysis (LDA) algorithm was used for gesture recognition and recognized gestures are converted into text and voice format. The proposed system helps to dimensionality.

Link : https://www.ripublication.com/ijaer18/ijaerv13n9_90.pdf

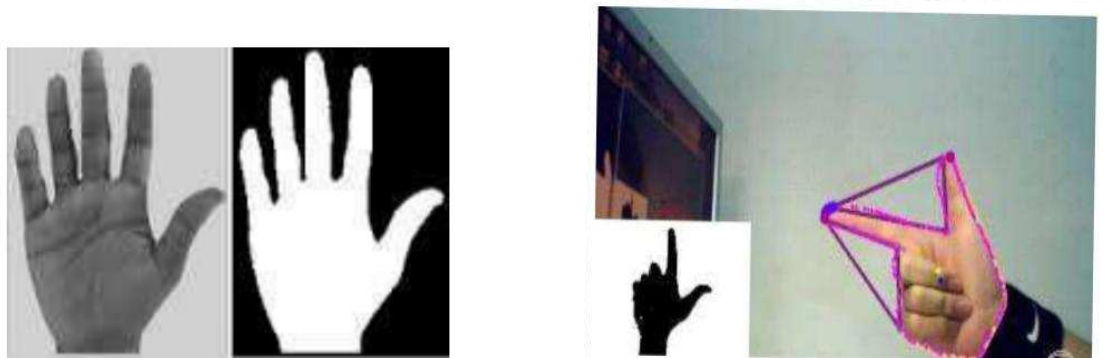


Figure 1 explains about the sign language of 26 gestures.



Figure 1. Gestures of sign recognition

2

Sign Language Recognition using Deep Learning, Mohamed Mahyoub (2023)

This paper investigates the effectiveness of deep learning techniques in recognizing sign language words from videos. It compares five different architectures ranging from simple CNN models to advanced hybrid architectures combining 3D CNNs, Transformers, and GRUs.

The study demonstrates that:

- Simple CNN models struggle with complex video-based sign recognition.
- Transformer-based sequence modeling significantly improves results.
- 3D CNNs (I3D) capture temporal information better than 2D CNNs & Deep hybrid architectures achieve near state-of-the-art performance.

Link :

<https://researchonline.ljmu.ac.uk/id/eprint/19754/1/1.%20Sign%20Language%20Recognition%20Using%20Deep%20Learning.pdf>

TABLE I. SIGN LANGUAGE DATASETS USED

Sign Language	INCLUDE Indian Sign Language (ISL)	WLASL American Sign Language (ASL)	AUTSL Turkish Sign Language (TSL)
# Words	263	2,000	226
# Word Categories	15	-	-
# Videos	4,292	21,083	38,336
# Frames	270,000	-	-
Video Resolution	1920x1080	-	512x512
# Signers	7	119	43



figure 2.

3

Recognizing American Sign Language gestures efficiently and accurately using a hybrid transformer model , Mohammed Aly& Islam S. Fathi (2024)

This paper presents a machine learning–based sign language recognition system designed to facilitate communication between deaf or mute individuals and people who do not understand sign language. The proposed system integrates computer vision and machine learning techniques to recognize hand gestures and convert them into text and speech in real time. The system uses **MediaPipe** to extract hand landmarks and detect gesture features, while a **Convolutional Neural Network (CNN)** is employed for feature extraction. A **Random Forest classifier** is then used for gesture classification. The model is integrated with a **Leap Motion Controller** to capture hand movements accurately and enable real-time recognition. The methodology includes data collection, preprocessing (resizing, grayscale conversion, normalization), feature extraction using hand keypoints, model training with Random Forest, and system deployment with a graphical user interface built using Tkinter. Additionally, the system incorporates text-to-speech functionality using pyttsx3 to provide auditory feedback. Although the paper reports successful real-time gesture recognition and system integration, the exact numerical accuracy is not clearly specified. The evaluation metrics mentioned include accuracy, precision, recall, and F1-score. Overall, the study demonstrates a practical implementation of a hybrid machine learning system that combines hardware (Leap Motion Controller) and software (CNN + Random Forest) to develop an assistive communication tool.

Link : <https://www.nature.com/articles/s41598-025-06344-8.pdf>

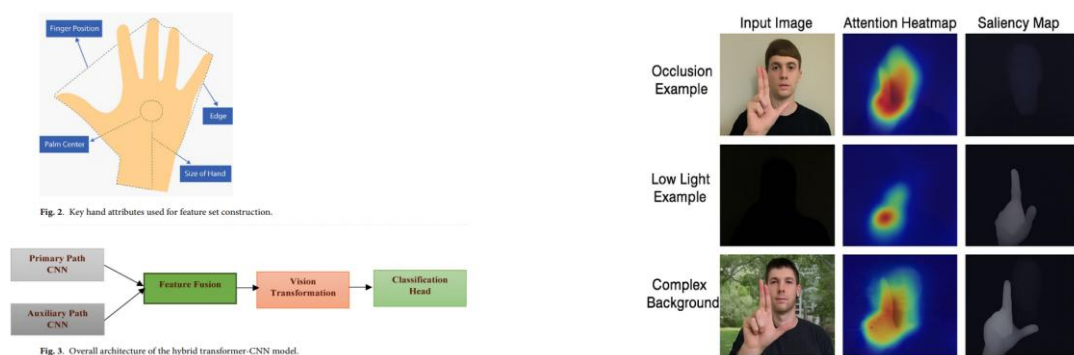


figure 3.

Analysis And Design

3.1 Comparison Table

Author name	Mahesh Kumar	Mohamed Mahyoub	Mohammed Aly& Islam S. Fathi
Dataset	26 hand gestures in Indian sign language	American Sign Language Lexicon Video Dataset (ASLLVD)	American Sign Language
Algorithm & Model	FAST + SURF feature extraction + KNN classifier	4-Layer CNN, VGG16, VGG16+Attention, VGG16+Transformer+ GRU, I3D+Transformer+GRU	CNN + Random Forest + MediaPipe + Leap Motion
Accuracy	80%	Up to 98% (INCLUDE), 96% (AUTSL)	Not clearly specified (X%)
Year	2018	2023	2024
Framework	MATLAB + Kinect	Python, OpenCV, Numpy	Python, OpenCV, Scikit-learn, Tkinter
Methodology	Image segmentation → ROI extraction → Feature detection → KNN classification → Text-to-Speech	Video dataset split (70/10/20) → CNN feature extraction → Transformer encoding → GRU decoding → Accuracy evaluation	Preprocessing → Landmark extraction → CNN feature extraction → Random Forest classification → Real-time gesture recognition

Literature Review

4

Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique, Mohammed Zakariah, Yousef Ajmi Alotaibi (2022)

This study proposes an automated system to interpret **Arabic Sign Language (ArSL)** alphabets and convert them into textual information. To address the complexity and dialects within Arabic sign language, the researchers employed a **Transfer Learning** approach using several pre-trained models. The final system aims to provide a low-cost communication solution for the deaf and mute communities.

Link : <https://pdfs.semanticscholar.org/5739/0c1b6dfd7a89b91d102c4019e39ff20cc488.pdf>

Figure 1 explains about the sign language of 26 gestures.



Figure 1. Gestures of sign recognition

5

A machine learning-driven web application for sign language learning, Hope Orovwode ,Oduntan Ibukun (2024)

This research presents an interactive **web application** designed specifically for learning the **American Sign Language (ASL)** alphabet. Unlike systems focused solely on translation, this application uses a "gamified" methodology where users are assigned words to spell out via their webcam. The system provides live feedback, awarding points when letters are signed correctly, thereby fostering inclusivity by making sign language education more accessible through common web browsers.

Link : <https://public-pages-files-2025.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1297347/pdf>

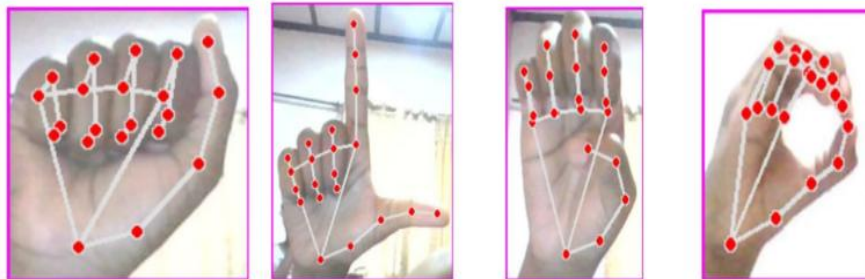


FIGURE 1
Sample of the dataset (Orovwode et al., 2023).

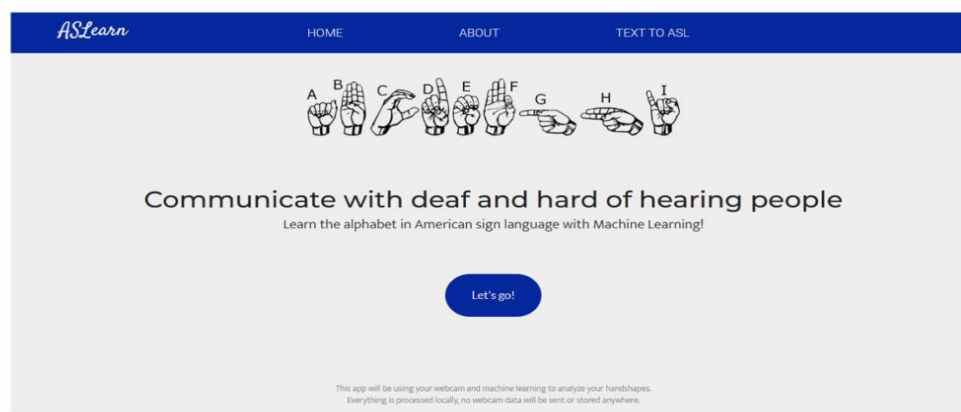


FIGURE 4
Landing page of the application.

figure 2.

6

Translation of Sign Language Finger-Spelling to Text using Image Processing, Krishna Modi, Amrita More (2013)

This is a foundational study using **classical Image Processing** techniques for ASL finger-spelling. Unlike the newer papers that use Deep Learning, this system relies on **BLOB analysis** and skin-color detection to isolate the hand. It works by capturing video frames, converting them to binary images, and comparing the "blob" features against a statistical database. It represents the "traditional" approach to computer vision, where manual feature extraction (like hand area and perimeter) is the primary method of recognition.

Link : <https://research.ijcaonline.org/volume77/number11/pxc3891313.pdf>

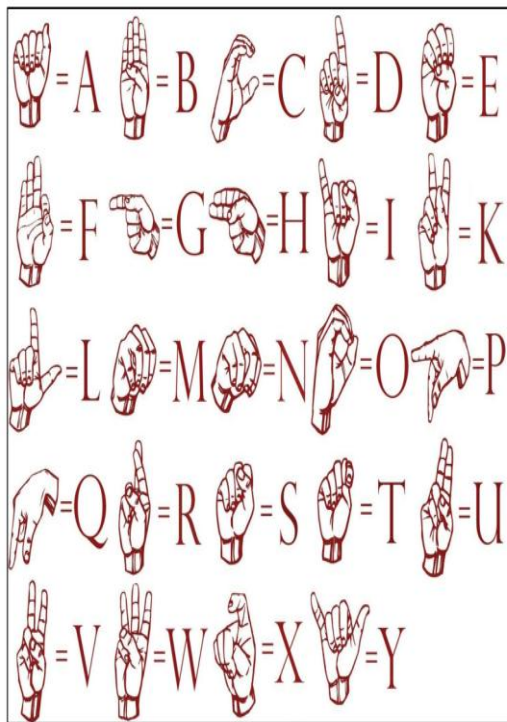


Fig 5: Collection of the American Sign Language Finger-spellings with their corresponding alphabets

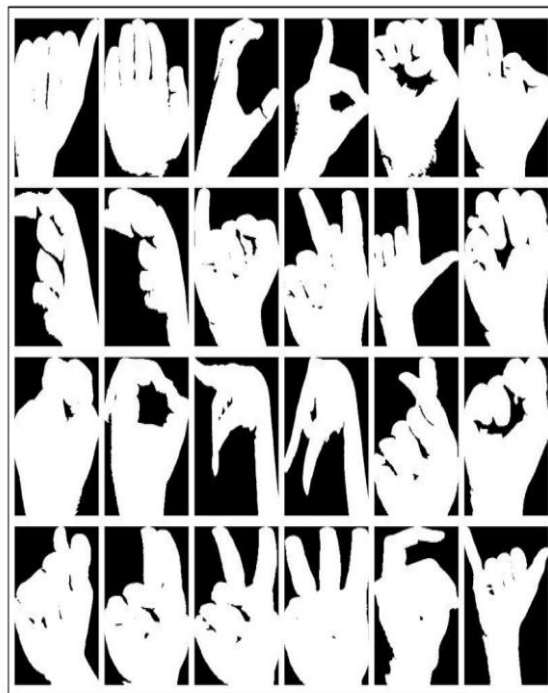


Fig 6: Collection of the images stored in the database

figure 3.

Analysis and Design

3.1 Comparison Table

Author name	Mohammed Zakariah& Yousef Ajmi	Hope Orovwode, Oduntan Ibukun	Krishna Modi & Amrita More
Dataset	Arabic Sign Language consists of 54049 images	American SL Alphabet Dataset: A collection of static images and real-time webcam captures.	American Sign Language (Custom Dataset: 26 letters (A-Z) captured via webcam.)
Algorithm & Model	FAST + SURF feature extraction + KNN classifier	Convolutional Neural Network (CNN): Integrated into a web-based delivery system.	Image Processing & BLOB Analysis (Statistical approach).
Accuracy	Training accuracy: 98% ; Testing accuracy: 95%	CNN models achieved 99%	On an average, we obtained that there is 96%
Year	2022	2024	2013
Framework	Keras (running on top of TensorFlow).	Flask (Python) for the backend, with HTML, CSS, and JavaScript for the frontend.	AForge.NET framework (C#/.NET).
Methodology	1. Preprocessing: rescaling images to (32x32). 2. Data Augmentation 3. Transfer Learning: Custom-trained pre-trained ImageNet models.	Interactive Gamification: Users spell words via webcam; the system predicts signs in real-time and grants points for correct sequences.	Classical Vision: Frame capture, thresholding, and geometric feature extraction.