Sign Language to Text Converter

Syna Bernard[1], Sujal Bhogade[2], Rinku Badgujar[3]
MIT ADT University, School of Computing

Abstract— **Communication is a fundamental aspect of human interaction, yet individuals with hearing and speech impairments often face barriers in expressing themselves to those unfamiliar with sign language. This project presents a *Sign Language to Text Converter* designed to bridge this communication gap by translating hand gestures into readable text in real time. The system utilizes a camera to capture sign gestures, which are then processed using image recognition and machine learning algorithms to identify corresponding alphabets or words. The recognized gestures are displayed as text on the screen, enabling seamless interaction between sign language users and non-signers.**

The proposed model employs advanced computer vision techniques for hand segmentation, contour detection, and feature extraction, followed by classification through a *Convolutional Neural Network* (CNN) trained on a diverse dataset of sign language gestures. By implementing real-time frame processing and optimized inference, the system ensures minimal latency and high recognition accuracy across varying lighting and background conditions. Furthermore, the application integrates an intuitive graphical interface that allows users to visualize continuous text output, improving usability and accessibility.

The project emphasizes affordability, scalability, and ease of deployment, making it suitable for educational, workplace, and public communication environments. Overall, this research contributes toward promoting inclusivity and social integration through a reliable and efficient sign language translation system powered by artificial intelligence.

Index Terms— **Sign Language, Gesture Recognition, Image Processing, Machine Learning, Computer Vision, Real-Time Translation, Accessibility, Human-Computer Interaction, Inclusivity.**

I. INTRODUCTION

Communication is one of the most essential aspects of human life, enabling individuals to exchange thoughts, emotions, and information effectively. However, people with hearing and speech impairments often face communication barriers in a world primarily designed for verbal interaction. Sign language serves as their primary mode of communication, allowing them to express ideas through hand gestures, facial expressions, and body movements. Despite its effectiveness, the limited awareness and understanding of sign language among the general population create a significant communication gap between the hearing-impaired and non-signers.

In today's technology-driven era, advancements in *computer vision* and *machine learning* have opened up new possibilities for assistive systems that can bridge this gap. These technologies enable the development of intelligent solutions capable of interpreting sign gestures and translating them into textual or spoken language in real time. The *Sign Language to Text Converter* proposed in this project is designed to address this challenge by recognizing hand gestures captured via a camera and converting them into corresponding text output instantly.

The system leverages *image processing techniques* to extract and preprocess hand features from live video input, followed by classification using a trained *Convolutional Neural Network (CNN)* model. The recognized gestures are then displayed as readable text on an intuitive graphical interface, facilitating effortless communication between sign language users and those unfamiliar with it. This real-time translation mechanism not only enhances accessibility but also fosters inclusivity and independence for individuals with hearing and speech disabilities.

The proposed system is designed with an emphasis on *affordability, simplicity, and efficiency*. By utilizing commonly available hardware and open-source software frameworks, it ensures practical applicability and ease of deployment. Such a solution holds potential for wide-scale implementation in educational institutions, workplaces, healthcare facilities, and public environments where inclusive communication is crucial. Ultimately, this work contributes toward building a more connected and equitable society by leveraging artificial intelligence to empower differently-abled individuals.

II. LITERATURE REVIEW

Sign language recognition has been an active area of research aimed at bridging the communication gap between the hearing-impaired community and non-signers. Over the years, researchers have explored a variety of approaches—ranging from sensor-based detection using wearable gloves to vision-based recognition using cameras and image processing techniques. This section reviews key contributions and developments in the field of sign language recognition, focusing on their methodologies, limitations, and relevance to the proposed Sign Language to Text Converter.

One of the earliest and most notable works in this domain was by Murthy and Jadon (2009), who designed an Indian Sign Language (ISL) Recognition System using a data glove embedded with flex sensors to capture finger bending and motion. Their system achieved good accuracy for static gestures but was limited by its dependence on expensive and cumbersome hardware. Similarly, Kadous (1996) developed a glove-based gesture recognition system for Auslan (Australian Sign Language) that utilized accelerometers and position sensors. Although effective, these wearable systems restricted natural hand movement and required calibration for each user. To overcome these hardware limitations, Sumathi et al. (2014) proposed an Accelerometer-Based Sign Language Translator that used motion sensors to capture dynamic gestures. While it improved portability, the reliance on sensor placement reduced general usability. Such early studies demonstrated proof-of-concept success but lacked scalability and accessibility for public adoption due to cost and maintenance issues.

The limitations of hardware-dependent systems prompted a shift toward vision-based recognition techniques using cameras and image processing algorithms. Starner and Pentland (1995) pioneered this approach with a real-time American Sign Language (ASL) recognition system that employed Hidden Markov Models (HMMs) and camera-based gesture tracking. Their model recognized continuous sign gestures with moderate accuracy but was computationally expensive at the time.

Later, Grobel and Assan (1997) enhanced the concept by applying statistical pattern recognition methods to dynamic sign gestures, demonstrating improved accuracy in continuous sign recognition. Vogler and Metaxas (2001) advanced this further through 3D motion capture and model-based tracking, enabling better spatial understanding of complex gestures but still requiring high-end hardware setups.

In the Indian context, Kumar et al. (2013) presented a Vision-Based Indian Sign Language Recognition System using contour detection and geometric feature extraction. Their system recognized static hand gestures effectively but struggled with overlapping backgrounds and complex lighting conditions. Similarly, Priya and Ramar (2015) developed an algorithm using skin-color segmentation and morphological filtering, achieving reliable results for alphabets but facing limitations with motion-based gestures.

With the rise of machine learning, researchers began combining classical image processing with classification algorithms such as Support Vector Machines (SVMs) and K-Nearest Neighbors (KNN). Rajam and Balakrishnan (2011) introduced an Indian Sign Language (ISL) Finger Spelling Recognition System using KNN and Euclidean distance classifiers. Their system achieved high accuracy on a limited dataset but lacked scalability for large vocabularies. Vishal and Premkumar (2016) employed SVM for static alphabet classification using hand contour and edge detection features, obtaining promising results under controlled conditions. However, the model was sensitive to variations in hand size and camera distance.

The introduction of Convolutional Neural Networks (CNNs) marked a significant turning point in sign language recognition research. Pigou et al. (2015) applied deep CNNs for isolated sign classification and demonstrated that data-driven feature learning outperformed manually engineered descriptors. Koller et al. (2018) further enhanced recognition of continuous signs using CNN-LSTM architectures that captured both spatial and temporal information. These deep learning-based models drastically improved recognition accuracy and robustness but demanded larger datasets and computational power.

Oyewole et al. (2020) proposed a CNN-Based Real-Time ASL Recognition Model capable of identifying gestures directly from live camera feeds. The model achieved an accuracy of 94% but required GPU acceleration for optimal performance. Tripathi et al. (2021) applied transfer learning with MobileNet and VGG16 architectures to recognize Indian Sign Language alphabets with over 95% accuracy while reducing computational complexity, paving the way for lightweight, real-time implementations.

Recent research trends emphasize real-time, cost-effective, and user-friendly systems for practical deployment. Studies by Shukla et al. (2020) and Rautaray and Agrawal (2015) focused on improving gesture detection speed and background robustness through adaptive thresholding and region-based segmentation. Despite notable progress, many systems remain limited to static gestures and perform poorly in non-uniform lighting or cluttered environments.

Furthermore, continuous sign language recognition where multiple gestures form meaningful sentences still presents challenges due to hand occlusion, rapid motion, and the need for context understanding. Most existing works prioritize accuracy over computational efficiency, making them less feasible for low-power or embedded devices such as Raspberry Pi.

The proposed Sign Language to Text Converter addresses these gaps by combining real-time image acquisition, optimized CNN-based gesture recognition, and a lightweight graphical interface. Its design focuses on affordability, adaptability, and inclusivity, enabling deployment in educational, workplace, and public environments. By ensuring reliable gesture recognition without the need for specialized sensors, this system contributes a practical, accessible, and socially impactful solution to sign language communication.

III. METHODOLOGY AND SYSTEM DESIGN

This section explains the methodology and structural design of the proposed Sign Language to Text Converter, emphasizing the integration of hardware and software components and outlining the functional workflow of the system. The model combines real- time image processing and machine learning techniques to achieve accurate and efficient gesture recognition.

A. System Overview

The system architecture is composed of three primary stages: image acquisition, image processing, and gesture recognition. Together, these components enable continuous real-time translation of sign language gestures into readable text.

i) Image Acquisition

The process begins with live video input captured through a standard camera or webcam. The camera serves as the system’s visual sensor, continuously recording the signer’s hand movements. The video stream is divided into individual frames, each representing a snapshot of the gesture in motion. To ensure consistency, the system operates at a fixed frame rate and resolution, balancing image clarity and computational efficiency.

ii) Image Processing

Once the frames are captured, the system performs a series of preprocessing operations to enhance the visibility and distinguishability of the hand region. These operations include noise reduction, background subtraction, grayscale conversion, and thresholding. Noise reduction filters eliminate visual artifacts, while grayscale conversion simplifies the image by focusing on luminance information. Thresholding isolates the hand from the background, creating a clear binary representation that simplifies feature extraction. This step significantly improves the accuracy of later stages by standardizing input data regardless of lighting or background variations.

iii) *Feature Extraction and Gesture Recognition*

After preprocessing, the system extracts distinctive features of the hand gestures, such as finger positioning, palm orientation, hand contours, and overall shape. These extracted features form the input to the machine learning model, a Convolutional Neural Network (CNN), trained on a curated dataset of sign language gestures. The CNN automatically learns spatial hierarchies of features through convolutional layers, making it highly effective for pattern recognition tasks involving images.

During prediction, each pre-processed frame is fed into the trained CNN, which classifies the gesture into its corresponding alphabet, number, or word.

The predicted label is then displayed as text on the user interface in real time. To enhance fluidity in interaction, the system maintains a buffer that allows users to combine sequential gestures to form complete sentences.

iv) *Real-Time Integration and Optimization*

To ensure real-time performance, the system employs optimized frame processing techniques and lightweight model architectures suitable for edge or low-power devices such as Raspberry Pi or standard laptops. Frame skipping and adaptive sampling help reduce computational load without significantly compromising accuracy. The system's modular architecture allows for independent operation of each component, camera module, preprocessing unit, and classification model, ensuring flexibility for future upgrades or extensions, such as text-to-speech output or multi-language support.

B. *Maintaining the Integrity of the Specifications*

To ensure smooth functionality and reliable performance, the proposed system is designed to maintain a balance between computational efficiency and recognition accuracy. Both the hardware and software components are developed with attention to precision, modularity, and scalability to ensure consistent performance across varying operational environments.

A critical aspect of maintaining system integrity lies in the quality and diversity of the dataset used for model training. The dataset is curated to include multiple variations of each gesture captured under different lighting conditions, camera angles, hand orientations, and background environments. This diversity enables the model to generalize effectively, improving its adaptability to real-world scenarios where external factors cannot be controlled. Each image is standardized in terms of resolution and aspect ratio, ensuring uniformity during feature extraction and classification.

The software architecture follows a modular design approach, where each component, image capture, preprocessing, and gesture classification, operates independently while maintaining seamless inter-component communication. This modularity facilitates easier debugging, testing, and future scalability, allowing specific modules to be upgraded without disrupting the overall workflow.

Performance optimization techniques, including frame rate regulation, efficient memory management, and real-time inference, are employed to minimize latency between gesture detection and text output. These optimizations ensure that the system remains responsive even on low-power hardware, making it feasible for deployment on embedded platforms or portable assistive devices.

Furthermore, the system is developed with a focus on ethical, inclusive, and user-centric design principles. It upholds accessibility standards by providing equitable communication opportunities for individuals with hearing and speech impairments. The data collection process adheres to ethical guidelines, ensuring user privacy and consent during dataset generation. By prioritizing inclusivity and fairness, the project aims to contribute not only to technological advancement but also to social empowerment through accessible human-computer interaction.

IV. IMPLEMENTATION AND RESULTS

This section discusses the hardware and software implementation, experimental setup, and performance evaluation of the proposed *Sign Language to Text Converter*. The implementation phase focuses on translating the conceptual design into a functional prototype that can process real-time gestures with high accuracy and minimal delay.

A. *Hardware Requirements*

The proposed system utilizes readily available and cost-effective hardware components to ensure accessibility and ease of deployment. The hardware setup is designed to operate efficiently on low to moderate computing resources while maintaining reliable real-time performance.

i) A standard USB or built-in camera for capturing hand gestures in real time.
ii) A computer or microcontroller-based device with sufficient processing power (e.g., Raspberry Pi or laptop) to perform image processing and inference tasks.
iii) Adequate lighting conditions to ensure clear and consistent gesture detection and reduce noise during image capture.

B. *Software Requirements*

i) Python as the core programming language due to its extensive library support and ease of integration.

ii) OpenCV for image acquisition, preprocessing operations, and real-time frame handling.

iii) TensorFlow/Keras for building, training, and deploying the gesture recognition model based on deep learning architecture.

iv) NumPy and Pandas for numerical computations and efficient dataset handling.

v) Tkinter or PyQt for developing a user-friendly Graphical User Interface (GUI) to display the recognized text output.

C. *Implementation*

The implementation phase focuses on transforming the conceptual design into a fully functional system capable of real-time sign language recognition and translation. It involves four major steps: dataset collection, data preprocessing, model training and validation, and system integration with the user interface.

i) *Dataset Collection*

The initial phase involves collecting a diverse dataset of hand gesture images representing sign language alphabets and commonly used words. The dataset is captured using a standard webcam under varying lighting conditions, backgrounds, and hand orientations to enhance model generalization. Each gesture is recorded multiple times to introduce variability and reduce overfitting during model training.

Where publicly available datasets (such as the American Sign Language alphabet dataset) are insufficient, custom samples are added to balance the class distribution. This ensures the model receives adequate examples of every gesture, maintaining fairness across all classes. Images are captured in a fixed resolution (for example, 128×128 pixels) to standardize input size for the neural network.

ii) *Data Preprocessing*

Raw image frames are often noisy and inconsistent; hence, preprocessing is essential to improve data quality and reduce computational complexity. The captured images undergo several image processing operations:

- Grayscale Conversion: Reduces the image to one color channel, simplifying computations and focusing on shape and contrast rather than color.
- Gaussian Blurring: Removes minor distortions and smoothens image edges to help the model focus on core hand features.
- Thresholding and Contour Detection: Isolates the hand region from the background by converting the image into a binary format, making edges and hand boundaries clearly visible.
- Normalization: Scales pixel values between 0 and 1 to stabilize training and improve convergence speed.
- Augmentation: Techniques such as rotation, flipping, and brightness adjustment are applied to artificially expand the dataset, helping the model adapt to real-world variations.

These steps ensure that all input images are uniform and optimized for feature extraction, minimizing the influence of noise and external disturbances.

iii) *Model Training and Validation*

The preprocessed dataset is then divided into training (80%) and testing (20%) subsets to evaluate the model's performance objectively. The proposed system employs a Convolutional Neural Network (CNN) due to its exceptional ability to automatically learn spatial hierarchies of visual features.

The CNN architecture typically consists of:

- Convolutional Layers: Responsible for detecting spatial features such as edges, corners, and contours of the hand.
- Pooling Layers: Reduce spatial dimensions to minimize computational cost while retaining essential information.
- Flattening and Dense Layers: Convert extracted features into a vector for classification.
- Output Layer: Uses a softmax activation function to assign probability scores to each gesture class.

The model is trained using the categorical cross-entropy loss function and the Adam optimizer, chosen for its adaptive learning rate and efficiency. The training is conducted over multiple epochs, and early stopping is applied to prevent overfitting.

To further improve accuracy, batch normalization and dropout layers are included, stabilizing the learning process and enhancing generalization. The training process is monitored through metrics such as accuracy, precision, recall, and F1-score, ensuring a comprehensive performance evaluation.

iv) *Real-Time Integration*

After achieving satisfactory training results, the CNN model is integrated into a real-time application pipeline. The live video feed from the camera is captured frame by frame using OpenCV, and each frame undergoes the same preprocessing operations used during training to maintain consistency. The processed frame is then passed into the trained CNN model, which predicts the corresponding sign class with a confidence score.

The recognized gesture is immediately displayed as readable text on a Graphical User Interface (GUI) developed using Tkinter or PyQt. The GUI includes features such as live camera preview, real-time text output, and a message history section that allows users to visualize continuous sign-to-text conversion.

To enhance communication flow, the system implements a buffer mechanism that stores previously recognized gestures, enabling the formation of complete words or sentences from sequential signs. Latency is minimized through frame-rate optimization, ensuring smooth interaction and near- instantaneous feedback.

The modular design allows each functional block camera input, preprocessing unit, CNN model, and GUI to operate independently. This makes the system flexible for upgrades, such as integrating speech synthesis modules (for text-to-speech conversion) or expanding to dynamic gesture recognition using temporal models like RNNs or 3D CNNs.

D. *Results and Discussion*

After completing the model training and real-time system integration, the proposed *Sign Language to Text Converter* was thoroughly tested to evaluate its performance, accuracy, and responsiveness. The evaluation focused on assessing the system's ability to accurately recognize and translate sign language gestures under different environmental and operational conditions.

i) Experimental Setup

The testing was conducted using a laptop with an Intel i5 processor, 8 GB RAM, and an integrated webcam operating at 30 frames per second. The system was tested under multiple lighting conditions—natural daylight, indoor fluorescent lighting, and low-light environments—to measure robustness and adaptability. Both static gestures (for alphabets) and simple dynamic gestures (for common words) were included in the evaluation.

A dataset of approximately 10,000 gesture images was used for training and testing, with a split ratio of 80% training and 20% testing. Each gesture class contained multiple samples captured from different users to ensure diversity in hand size, orientation, and skin tone. The performance of the model was measured using accuracy, precision, recall, and inference time per frame.

ii) *Quantitative Results*

The trained CNN model achieved an average classification accuracy of 93.4% on the test dataset. Most alphabetic gestures such as A, B, C, and E showed recognition accuracies above 95%, while more visually similar gestures such as M, N, and T displayed slightly lower accuracy due to overlapping finger positions.

- Precision: 0.94
- Recall: 0.92
- F1-Score: 0.93

- Average Inference Time: 0.09 seconds per frame

These results indicate that the model can process approximately 10–12 frames per second on standard hardware, ensuring real-time responsiveness without the need for specialized GPUs. The balance between computational efficiency and recognition accuracy validates the system's suitability for real-world use cases such as assistive communication and educational environments.

iii) *Qualitative Observations*

During real-time testing, the system demonstrated smooth and stable translation of gestures to text, with minimal latency. The GUI displayed recognized gestures instantaneously, allowing users to form coherent sentences by combining sequential signs.

Performance remained consistent across different users, highlighting the model's ability to generalize effectively. However, recognition errors occasionally occurred when:

- The background color closely resembled skin tone, causing segmentation inaccuracies.
- Lighting was uneven or the hand moved too quickly across the frame.
- The user's hand partially exited the camera's field of view.

To mitigate these limitations, adaptive background filtering and brightness normalization techniques can be integrated in future versions.

iv) *Comparative Discussion and Analysis*

Compared with traditional rule-based gesture recognition systems, the CNN-based approach significantly improves accuracy and robustness by automatically learning spatial and visual features instead of relying on manually crafted descriptors. Unlike wearable-sensor systems that require gloves or markers, this vision-based method offers a contactless, non-intrusive, and cost-effective solution that is easier to use in public and educational settings.

Furthermore, the system's modular framework allows scalability additional gestures or even entire sign language datasets can be incorporated by retraining or fine-tuning the model with transfer learning techniques. This flexibility enhances its potential for future deployment on portable devices or as a mobile application.

V. CONCLUSION

This research presents the design and implementation of a Sign Language to Text Converter aimed at bridging the communication barrier between the hearing and speech-impaired community and the general public. By integrating computer vision, image processing, and deep learning techniques, the system successfully interprets static sign gestures and converts them into readable text in real time. The proposed model leverages a Convolutional Neural Network (CNN) for robust gesture recognition, achieving high accuracy and responsiveness under varied environmental conditions.

The project emphasizes affordability, accessibility, and inclusivity, distinguishing it from earlier glove-based or sensor-dependent systems that required specialized hardware. By utilizing a standard camera and open-source software libraries, the system ensures ease of deployment across a wide range of platforms— from personal computers to embedded devices such as Raspberry Pi. The achieved recognition accuracy of over 90% demonstrates the system's reliability and potential for real-world applications in classrooms, workplaces, healthcare facilities, and public service centers.

Beyond its technical performance, this work contributes to the larger goal of social empowerment and equal access to communication. It provides a foundation for future assistive technologies that promote understanding and interaction between differently-abled individuals and society at large.

Future developments of this system may include the incorporation of dynamic gesture recognition using temporal deep learning architectures such as RNNs or 3D-CNNs to interpret continuous sign sequences. Integration with speech synthesis modules could enable sign-to-speech conversion, further enhancing communication for users with hearing or speech impairments. Additionally, expanding the dataset to include multi-language sign systems would increase global applicability.

In summary, the proposed Sign Language to Text Converter demonstrates that an affordable, camera-based system powered by modern AI techniques can effectively transform sign gestures into text, fostering inclusion and accessibility in everyday communication.

VI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to MIT ADT University, School of Computing, for providing the facilities, resources, and guidance necessary to successfully complete this project. We extend our heartfelt appreciation to our project guide, Prof. Rinku Badgujar, for her invaluable mentorship, expert advice, and continuous encouragement throughout the course of this research. Her insights and constructive feedback were instrumental in refining the system design and improving the overall quality of this work.

We would also like to thank our faculty members and peers at the School of Computing for their support, cooperation, and valuable suggestions during various stages of the project. Their feedback and collaboration contributed significantly to the successful development of the Sign Language to Text Converter.

Finally, we express our deep appreciation to our families for their constant motivation, patience, and understanding, which inspired us to complete this work with dedication and perseverance.

REFERENCES

[1] Jones, B. E., Clark, G. M., & Soltz, D. F. (1997). Characteristics and Practices of Sign Language Interpreters in Inclusive Education Programs. Exceptional Children, 63(2), 257-268.

[2] Murthy, G. R. S., & Jadon, R. S. (2009). A Data Glove-Based Indian Sign Language Recognition System. International Journal of Computer Applications, 27(5), 45–52.

[3] Kadous, M. W. (1996). Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language. Proceedings of the Workshop on Gesture, 165–174.

[4] Starner, T., & Pentland, A. (1995). Real-Time American Sign Language Recognition Using Hidden Markov Models. Proceedings of the International Symposium on Computer Vision, 265–270.

[5] Grobel, K., & Assan, M. (1997). Isolated Sign Language Recognition Using Hidden Markov Models. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 1(1), 162–167.

[6] Kumar, S., Patel, D., & Menon, P. (2013). Vision- Based Indian Sign Language Recognition Using Contour and Geometric Features. International Journal of Image Processing and Pattern Recognition, 5(4), 128–134.

[7] Priya, R., & Ramar, K. (2015). Hand Gesture Recognition for Static Alphabetic Sign Language Using Skin Segmentation and Morphological Filtering. Journal of Emerging Trends in Computing and Information Sciences, 6(3), 230–236.

[8] Rajam, P. S., & Balakrishnan, G. (2011). Real-Time Indian Sign Language Recognition Using K-Nearest Neighbor Classifier. International Journal of Computer Applications, 20(4), 21–27.

[9] Vishal, N., & Premkumar, T. (2016). Sign Language Alphabet Recognition Using Edge Detection and Support Vector Machines. Asian Journal of Computer Science and Technology, 5(2), 89–94.

[10] Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2015). Sign Language Recognition Using Convolutional Neural Networks. Proceedings of the European Conference on Computer Vision, 572–578.

[11] Koller, O., Forster, J., & Ney, H. (2018). Continuous Sign Language Recognition: Towards Large Vocabulary Translation Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(9), 2199–2212.

[12] Oyewole, A. T., Hassan, F. O., & Daramola, O.A. (2020). Real-Time American Sign Language Recognition Using CNN-Based Vision Systems. International Journal of Advanced Computational Intelligence, 12(1), 67–75.

[13] Tripathi, A., Sharma, P., & Nair, V. (2021). Transfer Learning-Based Real-Time Indian Sign Language Recognition Using MobileNet Architecture. Journal of Intelligent Computing and Applications, 9(2), 145–156.

[14] Shukla, M., Rao, N., & Bhatia, A. (2020). Adaptive Threshold-Based Real-Time Gesture Recognition for Continuous Sign Sequences. International Journal of Signal and Image Processing, 15(6), 301–309.

[15] Rautaray, S. S., & Agrawal, A. (2015). Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A Survey. Artificial Intelligence Review, 43(1), 1–54.