iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82886

Real Time American Sign Language Recognition with Deep Neural Networks

Amrita Sarkar[1], Samar[2], Anish Sharma[3], Abhiranjan Kumar[4], Parantap[5], Krishu Kumar[6]
[1]Assistant Professor, [2, 3, 4, 5, 6]BCA students, Department of CSE, Birla Institute of Technology, Mesra (Lalpur)

Abstract: American Sign Language (ASL) is a language consisting of hand signs, gestures, facial expressions, and body movements. This paper focuses on building a real-time system that can recognise ASL hand gestures using deep learning and computer vision. The primary goal is to assist individuals with hearing or speech difficulties in communicating more easily by translating their hand signs into text and speech. This system uses a webcam to capture live hand movements and applies Google's MediaPipe technology to track 21 key points on the hand. Instead of using raw images, the system works with these points to better understand each gesture. A deep learning model is trained using a combination of CNN (Convolutional Neural Network) to understand the shape of the hand and LSTM (Long Short-Term Memory) networks to know how the hand moves over time. Each gesture is seen as a sequence of 30 frames, which helps the model learn how signs are made in real life. Once a gesture is recognised, the system shows the result on a simple and easy-to-use interface created with Tkinter. It can also read the recognised words aloud using a text-to-speech feature, making the system helpful even for users with visual challenges.
Keywords: American Sign Language, Hand Gesture Recognition, Deep Learning, CNN, LSTM, MediaPipe.

I. INTRODUCTION

Communication is an important part of our daily lives. While most people use spoken or written language to express themselves, individuals who are deaf or have difficulty speaking often use sign language, such as American Sign Language (ASL), to communicate. However, since not everyone understands ASL, it can create barriers between people, limiting interaction and understanding.

To solve this problem, we developed a real-time ASL recognition system that uses a webcam to detect hand gestures and translates them into text and speech. The system applies MediaPipe, a powerful tool developed by Google to track 21 key points on the user's hand. These points help the system understand what gesture is appropriate. We then feed these points into a deep learning model that combines two types of neural networks, CNN for understanding the shape of the hand and LSTM for understanding the movement over time. This helps the system recognize not just still gestures but also signs that involve motion. Each ASL gesture is recorded as a sequence of 30 frames (images), which allows the model to learn the full movement of the sign. After recognizing a gesture, the system shows the corresponding letter or word on the screen using a user-friendly Tkinter interface. In this way, it can form complete sentences. To make it even more comprehensive, a text-to-speech feature reads the sentence aloud. It helps people who are deaf or mute and also gives support those who have visual losses.

The main aim of this research is to bridge communication gaps that assist people who interact more easily using sign language with those who don't understand it. It makes everyday communication smoother and more effective.

II. LITERATURE REVIEW

The field of Sign Language Recognition (SLR) has seen significant advancements over the past two decades, steered by progress in computer vision, machine learning, and deep learning. Researchers have focused on minimizing the communication gap between the hearing-impaired community and the broader society by developing systems that translate visual hand gestures into spoken or written language.

A. Sign-Language Recognition Systems and Methodologies

Initial research in sign language recognition primarily centred on static gestures and finger-spelling. An early Indian Sign Language (ISL) system applied skin-colour segmentation, binary image processing, and classification techniques by employing neural networks and Support Vector Machines (SVMs) and achieved acceptable accuracy levels [Rokade & Jadav, 2017]. Advancing this approach, researchers employed Scale-Invariant Feature Transform (SIFT) descriptors in combination with a feed-forward backpropagation neural network, further optimized using the Artificial Bee Colony algorithm, resulting in an impressive accuracy of 99.43% [Kaur & Krishna, 2019].

With the advancement of deep learning, Convolutional Neural Networks (CNNs) appeared as the dominant approach in sign language recognition. A signer-independent CNN working on silhouette images recognised static ISL alphabets at 98.64 % accuracy [Sruthi & Lijiya, 2019]. A real-time ASL system that classifies alphabetic hand shapes captured from live video defined the feasibility of translation for deaf-and-mute users [Deshpande et al., 2023].

Dynamic-gesture work leans on spatiotemporal modelling. A 3-D CNN trained on depth and intensity data reached 77.5 % accuracy under challenging lighting conditions [Molchanov et al., 2016], while a Faster R-CNN plus 3-D CNN and LSTM pipeline reported up to 99 % accuracy on continuous-sign sequences [He, 2019]. Simpler vision pipelines—convex-hull features with k-nearest neighbours—remain viable for lightweight isolated-gesture systems, though with lower performance (65 % accuracy) [Amrutha & Prabu, 2021]. Hybrid systems that combine gesture recognition with speech-to-sign conversion have also been explored [Kumar et al., 2016].

B. Language-Specific Approaches

Research has historically focused on American Sign Language (ASL), yet multiple studies underline the urgency of localised ISL solutions. Signer-independent CNNs tailored to India's RPwD Act 2016 highlight the social importance of ISL tools [Sruthi & Lijiya, 2019]. Feature-descriptor methods further address ISL challenges such as hand-orientation variance [Kaur & Krishna, 2019; Rokade & Jadav, 2017].

C. Feature-Extraction and Classification Techniques

Beyond basic colour and contour cues, Histogram of Oriented Gradients (HOG) combined with SVM/KNN has proved effective for complex symbol recognition [Chikmurge & Shriram, 2021]. Traditional CNN feature extractors remain a staple [Deshpande et al., 2023], whereas spatiotemporal features harvested via 3-D CNNs and LSTMs enhance dynamic-gesture performance [He, 2019; Molchanov et al., 2016]. Decision-fusion hybrids—e.g., MLP-CNN ensembles—show promise for integrating complementary spatial and spectral cues [Zhang et al., 2018].

D. System Architectures and Integration

Incorporating non-manual cues (facial expression, body posture) is critical for fluent, continuous-sign recognition [Loeding et al., 2004]. Comprehensive surveys still rate SLR behind automatic speech recognition in robustness and vocabulary size [Er-Rady et al., 2017]. Borrowing from speech-technology concepts, self-organised “fenone” sub-units have been proposed to scale vocabularies without exhaustive training data [Bauer & Karl-Friedrich, 2002].Virtually every modern SLR pipeline relies on the NumPy array-programming library for fast numerical operations on image and video tensors, underscoring its foundational role in scientific Python ecosystems [Harris et al., 2020].

The literature shows a decisive shift from handcrafted image processing to deep, often multimodal architectures. Key challenges persist in dynamic-gesture recognition, signer independence, integration of non-manual signals, and real-time deployment. Hybrid models that combine spatial, temporal, and contextual information are a rising path toward robust and scalable SLR solutions.

III. METHODOLOGY

We followed a step-by-step method that combines real-time video processing, deep learning techniques, and an interactive user interface to develop a reliable and user-friendly American Sign Language (ASL) recognition system. This section explains the framework of our system where we defined process of creation of dataset, and the design of the deep learning model used for gesture recognition. This section also illustrated how the American Sign Language (ASL) recognition system was built by applying deep learning model and making it for real time application.

IV. SYSTEM ARCHITECTURE

Our ASL recognition system is built to function smoothly in real time, making it easy and intuitive for users. It begins by capturing live video through a webcam as the user performs hand gestures, which serves as the raw input. For accurate detection, the user's hand gestures must be visible and well-lit. Each frame from the video is then passed through Google's MediaPipe framework, a fast and efficient framework that detects hand landmarks without requiring any additional sensors. The system finds and tracks 21 key landmarks on the hand in each video frame using MediaPipe. It also maps finger and palm positions and movements accurately. Once the hand landmarks are extracted, these landmark points are then fed into a deep learning model that analyzes the gesture and predicts its meaning. .

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

The recognized gesture is instantly displayed in a user-friendly graphical interface, where individual letters combine to form complete words or sentences. The system also includes text-to-speech functionality by reading the recognized sentence aloud for clear, real-time audio feedback.Figure1 shows the data flow diagram of complete sign language recognition process

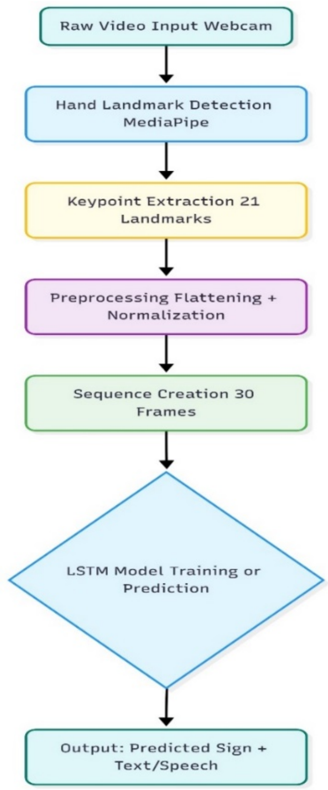


Figure 1: Dataflow and Processing Pipeline

In this study Figure 2 demonstrates the system architecture of a sign language recognition system that converts hand gestures into text and speech. The process begins capturing hand movements with a webcam. Then these hand movements processed using ROI cropping and MediaPipe hand tracking to monitor the hand. Key features are extracted from the tracked hand landmarks and fed into an LSTM (Long Short-Term Memory) model, which is well-suited for recognizing sequences of gestures. The model interprets these gestures and outputs the corresponding text, which is both displayed on the screen and converted into speech through a text-to-speech module, enabling effective communication for individuals with hearing or speech impairments.

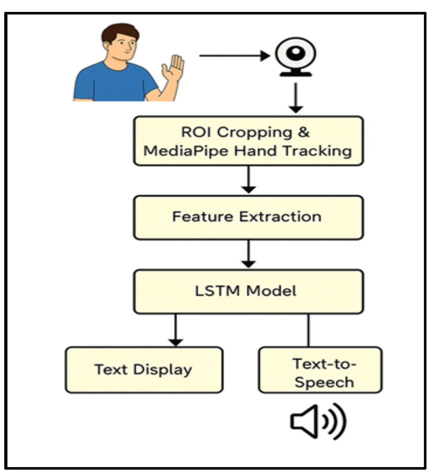


Figure 2: System Architecture

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Data Generation

Instead of using a ready-made dataset, we decided to create our own, which would allow the system to perform better in real-life situations. We used a webcam to capture the images of different ASL hand signs, including all 26 letters of the alphabet, as well as special signs like "Space" and "Delete." Each sign was captured as an image of 30 frames to capture the full movement of the hand.

In every frame, we used MediaPipe to find 21 key points on the hand. These points show the position of the fingers and palm in 3D using x, y, and z values. We saved these values in .npy files using NumPy so the data would be well-organized and easy to use later. Since we collected the data in the same environment where the system would run, it helped improve the accuracy and smoothness of real-time gesture detection.Figure 3 represents the collection of hand gesture images used to train the model for recognizing all 26 alphabets, along with space and delete commands. Whereas Figure 4 shows the process of capturing hand landmarks with MediaPipe and saving them as .npy files for model training.

Figure 3: Dataset (A-Z, Space and Delete)

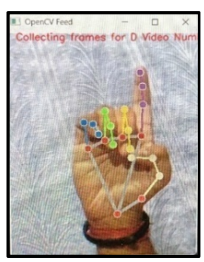


Figure 4: Data Generation into.npy Using MediaPipe

V. MODEL ARCHITECTURE

In our research, we built a real-time American Sign Language (ASL) recognition system by designing a deep learning model that combines the strengths of both Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) layers. The primary input to the model consists of sequences of hand landmarks captured using MediaPipe. Each frame contains 21 hand landmark points, each with x, y, and z coordinates, resulting in a total of 63 features per frame. However, because hand gestures involve movement, analyzing a single frame is not enough. To capture this motion, we collect a sequence of 30 frames, forming a 30×63 matrix as the model input. This format allows the model to learn not only the position of the hand but also the transition and movement of gestures over time.

To handle this sequential data effectively, we use LSTM layers instead of standard CNNs. LSTM is particularly well-suited for learning from time-series data because it retains memory of previous steps in the sequence.

The first LSTM layer begins by learning the basic motion pattern from the input. The second LSTM layer processes the information further and starts to identify gesture-specific patterns. Optionally, a third LSTM layer can be added to increase model depth and handle more complex variations in gesture movement. After passing through the LSTM layers, the output is fed into Dense layers (also called fully connected layers). These layers serve as decision-makers, interpreting the features extracted from the sequences and classifying them into specific gesture categories. We apply a ReLU activation function in these layers to allow the model to handle non-linear relationships between the features and their labels more efficiently. The last layer of our model is a Softmax layer, which outputs a probability distribution over all possible gesture classes (such as A to Z, Space, and Delete). The class with the highest probability is selected as the predicted gesture. For instance, if the highest probability value is associated with the letter "C", the system interprets the gesture as "C". Lastly, our model architecture takes in 30-frame sequences of 63 features, processes them through multiple LSTM layers to learn temporal patterns, uses Dense layers for decision-making, employs Dropout layers for regularization, and finally uses a SoftMax output layer to classify the gesture. This combination allows the system to accurately recognize ASL gestures in real time and convert them into meaningful text and speech output.

Figure 5 presents the architecture and parameter summary of the LSTM-based sign language recognition model. The model consists of three stacked LSTM layers followed by three dense (fully connected) layers. The LSTM layers progressively increase and then reduce the dimensionality of the sequential input data, capturing temporal patterns across the gesture sequences. The dense layers then map these features to the final output classes. The final dense layer outputs 28 classes, corresponding to 26 alphabet letters plus "Space" and "Delete." The model has a total of 564,470 parameters, out of which 188,156 are trainable, and 376,314 are optimizer parameters, with no non-trainable parameters, indicating all layers are actively updated during training.

Layer(type)	Output Shape	Param #
Lstm(LSTM)	(None, 30, 64)	32,768
Lstm_1(LSTM)	(None, 30, 128)	98,816
Lstm_2(LSTM)	(None, 64)	49,408
Dense(DENSE)	(None, 64)	4,160
Dense_1(DENSE)	(None, 32)	2,080
Dense_2(DENSE)	(None, 28)	924

```
Total params : 564,470 (2.15 MB)
Trainable params : 188,156 (734.98 KB)
Non-Trainable params : 0 ( 0 MB)
Optimizer params : 376,314 (1.44 MB)
```

Figure 5: LSTM Model Summary and Parameters

Proposed Deep Learning Model

The model architecture used in this research is based on LSTM layers followed by Dense layers. It starts with three LSTM layers. The first LSTM layer outputs 64 units over 30 time steps, followed by a second LSTM with 128 units, and then a third LSTM that compresses it back to 64 units. After that, the output is passed through three Dense layers — the first has 64 neurons, the second has 32, and the final layer has 28 neurons, which is probably for 28 ASL signs. Total trainable parameters are around 188k, and optimizer params take a large chunk, around 376k. This LSTM-Dense combo is very useful for gesture recognition over time. The model is small enough to run on most computers and good at learning time-based patterns like hand movements.

The LSTM layers help in understanding the sequence of keypoints that come from hand gestures. Instead of treating each frame individually, it learns from the sequence, which is very important for dynamic signs. The Dense layers at the end basically convert that understanding into prediction scores for each possible sign. Also, since we didn't use any non-trainable parameters, it means everything in the model contributes to learning. The final softmax layer gives the predicted sign with the highest probability. This architecture gave decent accuracy during testing and worked well with real-time input using MediaPipe.

VI. EXPERIMENTS AND RESULTS

In this section, we present the performance evaluation of our real-time American Sign Language (ASL) recognition system through a series of experiments focused on hand landmark extraction, real-time responsiveness, and predictive accuracy.

A. MediaPipe Hand Landmark Points

In our research, we use MediaPipe to extract 21 key landmark points from each frame to capture the pose and movement of the hand. These landmarks include the wrist, finger joints, and fingertips, and are represented by their x, y, and z coordinates. The coordinates are flattened into a single vector and normalized to ensure consistency across different frames and users.

This processed data is then fed into our LSTM model to predict the corresponding ASL gesture. Accurate capture and preprocessing of these keypoints are crucial, as the quality of this input directly affects the model's ability to recognize signs correctly.

Figure 6 provides a visual representation of these 21 hand landmarks as defined by the MediaPipe framework. Each red dot marks a specific joint or fingertip, such as point 0 at the wrist, point 4 at the tip of the thumb, and point 20 at the tip of the pinky. The green lines connecting the landmarks form a skeletal outline of the hand, enabling precise tracking of finger positions and motions. By offering detailed spatial and structural information in real time, MediaPipe greatly enhances the effectiveness of gesture-based AI systems like ours, making it especially valuable for real-time sign language recognition.

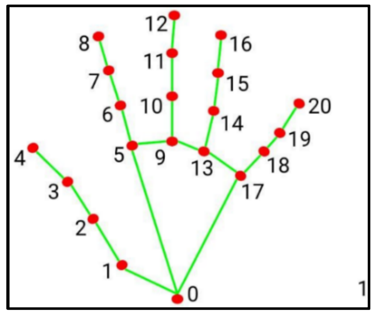


Figure 6: MediaPipe Hand Landmark Points

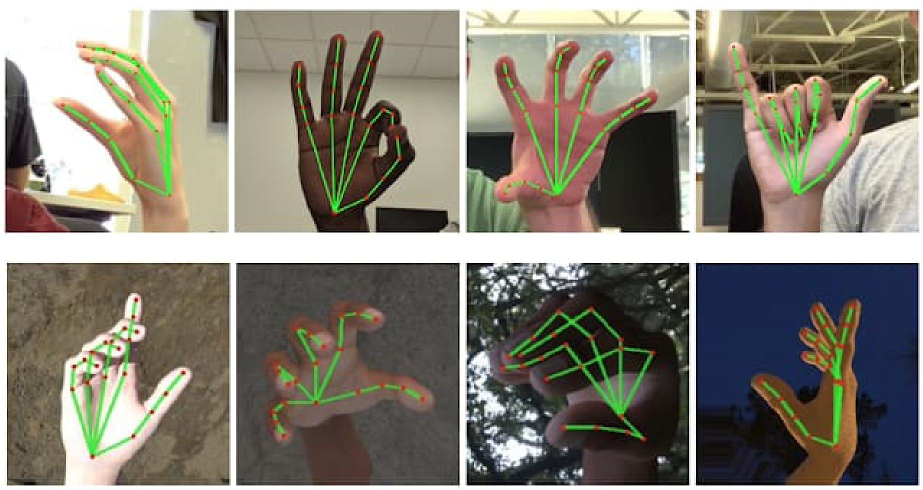


Figure 7: Hand gesture tracking using MediaPipe

Figure 7 above highlights the robustness of MediaPipe’s hand tracking capabilities by showcasing its consistent and accurate detection of hand landmarks across diverse conditions. In each frame, red dots indicate the 21 keypoints on the hand, while green lines connect them to form the skeletal structure. The top row displays a variety of hand gestures captured in well-lit indoor settings, while the bottom row demonstrates the system’s reliability in more challenging environments such as low lighting, motion blur, outdoor scenes, and shadows. This consistent performance across different gestures, skin tones, and backgrounds underscores MediaPipe’s adaptability and reliability, making it a powerful tool for real-time gesture recognition, including sign language interpretation.

B. Real-Time Performance

One of the key factors of our American Sign Language (ASL) recognition system is its real-time performance. The system instantly captures and processes hand gestures as they are seen in front of a webcam. MediaPipe quickly and accurately detects 21 hand landmarks with minimal delay. These landmarks are converted into numerical data and fed into the trained deep learning model for gesture recognition. By applying efficient processing pipeline and optimized architecture, the system runs at a stable frame rate of approximately 15–20 frames per second on a standard CPU. This confirms that users can see the recognized letter almost instantly. In addition, the system includes a user-friendly Tkinter GUI that displays gesture predictions in real time. It helps user to see the recognized output promptly. To enhance accessibility, a text-to-speech component has been integrated.

It provides audible feedback that is helpful for users with visual impairments. Performance evaluations confirmed that the model performs consistently well during both training and testing. The real-time results demonstrated that the system is not only accurate but also real-world applicable. This tool also bridges communication gaps between individuals with hearing or speech impairments and the broader community. Figure 8 shows the system in action, that recognizes the ASL gesture for "HELLO" in real time using a webcam. It also displays the result on both the OpenCV feed and the interface.

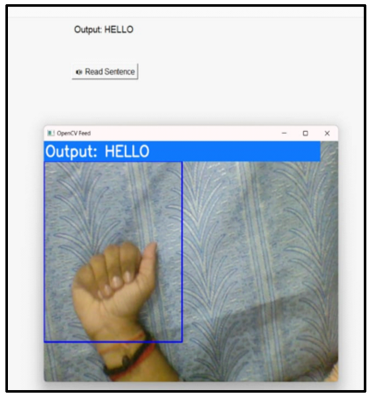
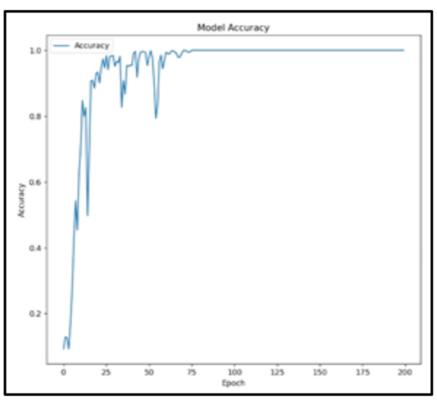


Figure 8: Real-time ASL gesture recognition displaying the output "HELLO" using a deep neural network via webcam input.

C. Evaluation Metrics

In this study, we evaluated the performance of our American Sign Language (ASL) recognition system using several metrics to ensure accurate gesture interpretation. The primary metric was accuracy, which indicates how often the system correctly identifies the gesture. Figure 9 displays the training accuracy of the ASL recognition model over 200 epochs. The graph shows a rapid increase in accuracy during the initial stages of training, that indicates the model's effectiveness in learning gesture patterns at early stage. By nearly the 50th epoch, the accuracy stabilizes near 1.0. It suggests strong generalization on the training data. The smooth, high-accuracy curve suggests that the model is well-trained and ready for real-time application. Our model achieved an impressive overall accuracy of around 95%, that means it correctly recognized 95 out of every 100 gestures. By applying these evaluation metrics, we confirmed the reliability of the model. This model can recognize not only the common signs but also on less frequent ones like "Space" and "Delete."



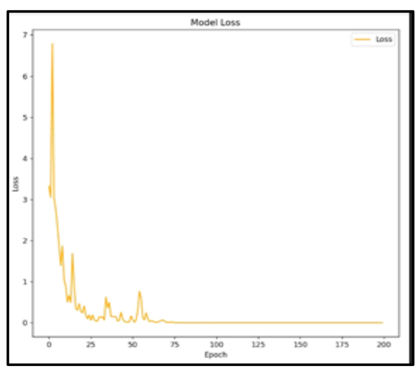


Figure 9: Model's Accuracy Figure 10: Model's Loss

In this research Figure 10 illustrates the corresponding training loss over the same 200 epochs. Initially, the model exhibits a high loss value that is above 6. It reflects significant prediction errors. However, the loss decreases sharply in the early epochs especially within the first 25. It indicates that the model is learning efficiently and minimizing errors. After about the 50th epoch, the loss levels off and approaches zero. It shows that the model has converged without signs of overfitting or underfitting. This combination of steady decline in loss with high accuracy, confirms that the model is not only well-optimized but also capable of detecting accurate real-time gesture recognition.

VII. DISCUSSION

Developing this real-time American Sign Language (ASL) recognition system was both an exciting and challenging journey. Instead of relying on pre-existing datasets, we chose to collect our hand gesture data using a standard webcam. This method allowed us to capturing variations in lighting, hand shapes, and positioning of users' hand. One of the key strengths of our approach is the use of dynamic gesture recognition. Instead of analyzing individual static images, our system looks at sequences of gestures over time. This was possible by combining MediaPipe's precise hand tracking with the learning capabilities of an LSTM-based deep learning model. While MediaPipe helped to identify 21 key landmarks in real time, and the LSTM layers enabled the system to recognize hand gestures. It makes it suitable for both static signs and with motion.

The integration of OpenCV, MediaPipe, and TensorFlow developed a smooth workflow, that captures and preprocesses video frames and performs model training and real-time prediction. Our user interface created using Tkinter, that provides instant visual feedback and reads aloud the predicted text by emloying text-to-speech engine. This not only enhances user interaction but also extends accessibility to users with visual impairments. The system isn't full proof. Some gestures with similar finger shapes—such as "U" and "V" or "M" and "N"—were occasionally misclassified. To overcome this, we worked on collecting more uniform training samples and included dropout layers in the model to reduce overfitting. Overall, the system performed well in real-time tests and offers smooth and responsive feedback. Its ability to convert sign language into readable and audible text shows that it can be a practical communication tool for people who are deaf, hard of hearing, or have speech impairments.

VIII. CONCLUSION AND FUTURE WORK

This research successfully demonstrates the development of a real-time American Sign Language (ASL) recognition system using MediaPipe for hand landmark detection and an LSTM-based deep learning model for gesture classification. By capturing 21 hand landmarks and analyzing sequences of 30 frames, the system effectively identifies both static and dynamic gestures with high accuracy 95%. The integration of a simple GUI and text-to-speech support further enhances usability, making it accessible not only to hearing- or speech-impaired users but also to those with visual impairments. Overall, the system provides a smooth, responsive, and practical solution for real-time sign language communication.

Looking ahead, there are several directions to improve and expand this work. Future iterations of the system can be enhanced to recognize more complex signs, support two-handed gestures, and handle continuous sentence-level inputs. Improving model robustness under varying lighting, backgrounds, and hand orientations is also essential for real-world deployment. Additionally, training the model on a more diverse dataset will improve its adaptability across different users. Porting the system to mobile or web-based platforms and incorporating multilingual speech output can make it more versatile and accessible for broader use across different regions and user communities.

REFERENCES

[1] Rokade, Y. I., & Jadav, P. M. (2017). Indian Sign Language Recognition System. International Journal of Engineering and Technology (IJET), 9(3S), 189 – 196. DOI: https://doi.org/10.21817/ijet/2017/v9i3/170903S030
[2] Kaur, J., & Krishna, C. R. (2019). An Efficient Indian Sign Language Recognition System using SIFT Descriptor. International Journal of Engineering and Advanced Technology (IJEAT), 8(6), 1456 – 1461. DOI: https://doi.org/10.35940/ijeat.F8124.088619
[3] Sruthi, C. J., & Lijiya, A. (2019). Signet: A Deep Learning based Indian Sign Language Recognition System. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), IEEE, Chennai, India. DOI: https://doi.org/10.1109/ICCSP.2019.8698006
[4] Deshpande, A., Shriwas, A., Deshmukh, V., & Kale, S. (2023) Sign language recognition system using CNN. In Proceedings of the 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), IEEE.
[5] DOI: https://doi.org/10.1109/IITCEE57236.2023.10091051
[6] Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2016) Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 1 – 7). DOI: https://research.nvidia.com/sites/default/files/pubs/2015-06_Hand-Gesture-Recognition/CVPRW2015-3DCNN.pdf
[7] He, S. (2019) Research of a Sign Language Translation System Based on Deep Learning. In Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), IEEE, Dublin, Ireland. DOI: https://doi.org/10.1109/AIAM48774.2019.00083
[8] Amrutha, K., & Prabu, P. (2021) ML ML-Based Sign Language Recognition System. In Proceedings of the 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), IEEE, Kottayam, India. DOI: https://doi.org/10.1109/ICITIIT51526.2021.9399594
[9] Kumar, A., Thankachan, K., & Dominic, M. M. (2016) Sign language recognition. In Proceedings of the 3rd International Conference on Recent Advances in Information Technology (RAIT), IEEE, Dhanbad, India. DOI: https://doi.org/10.1109/RAIT.2016.7507939
[10] Sruthi, C. J., & Lijiya, A. (2019) Signet: A Deep Learning based Indian Sign Language Recognition System. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), IEEE, Chennai, India, April 4–6, 2019. DOI: https://doi.org/10.1109/ICCSP.2019.8698006
[11] Kaur, J., & Krishna, C. R. (2019) An Efficient Indian Sign Language Recognition System using SIFT Descriptor. International Journal of Engineering and Advanced Technology (IJEAT), 8(6), 1456–1461. DOI: https://doi.org/10.35940/ijeat.F8124.088619

[12] Chikmurge, D., & Shriram, R. (2021) Marathi Handwritten Character Recognition Using SVM and KNN Classifier. In A. Abraham, S. Shandilya, L. Garcia-Hernandez, & M. Varela (Eds.), Hybrid Intelligent Systems (HIS 2019), Advances in Intelligent Systems and Computing (Vol. 1179). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-49336-3_32

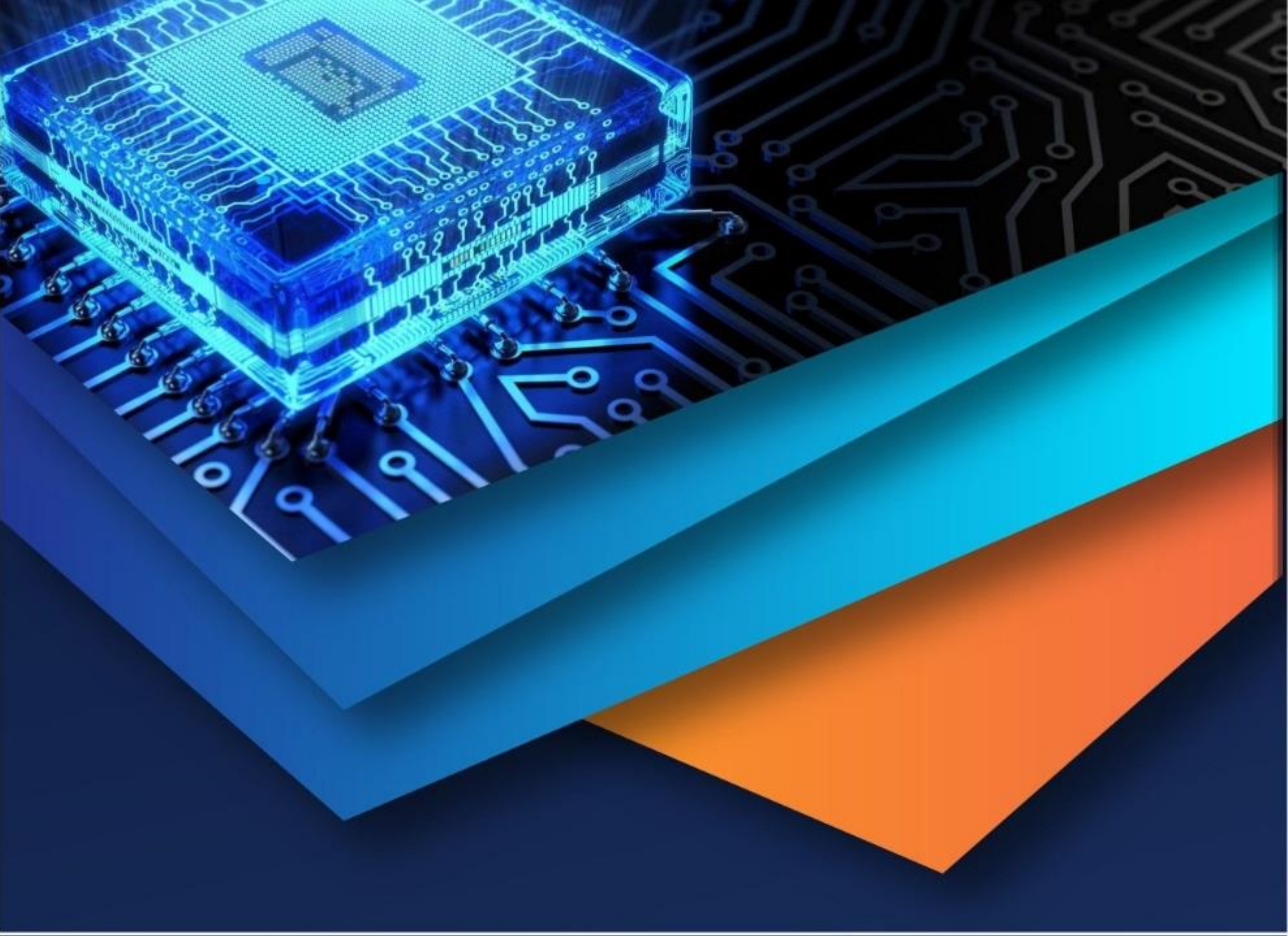
[13] Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., & Atkinson, P. M. (2018) A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS Journal of Photogrammetry and Remote Sensing, 140, 133 – 144. DOI: https://doi.org/10.1016/j.isprsjprs.2017.07.014

[14] Loeding, B. L., Sarkar, S., Parashar, A., & Karshmer, A. I. (2004) Progress in Automated Computer Recognition of Sign Language. In K. Miesenberger, J. Klaus, W. L. Zagler, & D. Burger (Eds.), Computers Helping People with Special Needs. ICCHP 2004, Lecture Notes in Computer Science (Vol. 3118, pp. 1079 – 1087). Springer, Berlin. DOI: https://doi.org/10.1007/978-3-540-27817-7_159

[15] Er-Rady, A., Faizi, R., Oulad Haj Thami, R., & Housni, H. (2017) Automatic sign language recognition: A survey. In Proceedings of the 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), IEEE, Fez, Morocco. DOI: https://doi.org/10.1109/ATSIP.2017.8075561

[16] Bauer, B., & Karl-Friedrich, K. (2002) Towards an Automatic Sign Language Recognition System Using Subunits. In I. Wachsmuth & T. Sowa (Eds.), Gesture and Sign Language in Human-Computer Interaction. GW 2001, Lecture Notes in Computer Science (Vol. 2298, pp. 64 – 75). Springer, Berlin. DOI: https://doi.org/10.1007/3-540-47873-6_7

[17] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … & Oliphant, T. E. (2020) Array programming with NumPy. Nature, 585(7825), 357 – 362. DOI: https://doi.org/10.1038/s41586-020-2649-2



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY