JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Real-Time Machine Learning Framework for Sign Language Detection Using TensorFlow and SSD

[1]Viveshan Kumar viveshankumar824@gmail.com,
[2]Naveen Kumar Ravi naveenravi1882002@gmail.com
[3]Abhishek Kumar Singh abhisheksinghmaknahi@gmail.com
[4]Harsh Bhushan Paswan harshbhushanpaswan@gmail.com
[5]Ajeet Kumar Gupta ajeetkgupta12@gmail.com
[1,2,3,4] B.Tech Student, [5]Assistant Professor

[1,2,3,4,5]Computer Science and Engineering, Darbhanga College of Engineering Darbhanga, Darbhanga, India

Abstract—The ML based Sign Language Detection system aims at communicating with differently abled people without the help of any expensive human interpreter. This model translates the signs gestures captured into text so that the user can simply read and know what the person is trying to convey irrespective of whether the user has knowledge about sign language or not. In this project, a real-time ML-based system is built using images captured with the help of webcam for sign language detection. The main purpose of this project is to design a system for people with differenabilities so that they can easily communicate with other people. The existing digital translator is very slow, because each letter must be gestured with, and it takes a long time to form a simple sentence. The model uses the SSD ML algorithm, and overcomes the above problems by directly recognizing the symbols as words instead of letters. Model will be built with the help of Label-Img software and TensorFlow Object Detection API, using real coloring images and detect sign language in real time using OpenCV. The LabelImg software is used for graphically labeling the images that is further used when recognizing the images. The labeling is done by drawing a box around the gesture made. SSD (Single Shot Detection) algorithm is designed for object detection in real-time. SSD speeds up the process by eliminating the need for the region proposal network. The TensorFlow Object Detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models Index Terms—Component, formatting, style, styling, insert.

Keywords: Real-Time Gesture Recognition, Computer Vision, Deep Learning, Convolutional Neural Network (CNN), Hand Gesture Recognition, OpenCV, Human-Computer Interaction, Assistive Technology, Text-to-Speech)

Introduction

Sign language is one of the most effective communication methods for deaf and speech-impaired individuals, allowing them to convey ideas, emotions, and daily needs through structured hand gestures, finger positions, and facial expressions. However, communication barriers arise when non-sign-language users are unable to understand these gestures, which often limits interaction in education, healthcare, workplaces, and public environments. To address this challenge, researchers are increasingly applying computer vision and deep learning techniques to automate sign interpretation systems. Recent studies have demonstrated that convolutional neural networks (CNNs) provide highly accurate gesture classification for sign language recognition tasks [1], [2].

Real-time sign language detection focuses on recognizing gestures instantly from live video streams captured through webcams or mobile cameras. The captured frames are first preprocessed to isolate the hand region, remove background noise, and normalize image features. After preprocessing, feature extraction techniques such as hand landmark detection using media pipe are employed to capture finger joint positions and gesture structure efficiently [3], [4]. These extracted

features are then passed into CNN-based classification models that identify alphabets, numbers, or meaningful words with high precision.

Several recent works have shown significant improvements in recognition accuracy using hybrid frameworks that combine media pipe-based landmark extraction with CNN models. For example, Kumar et al. Achieved 99.95% accuracy in real-time asl recognition using media pipe and CNN integration [3], while Verma et al. Reported 99.12% accuracy with enhanced real-time gesture processing [4]. Similarly, Pandey highlighted the importance of deep learning and computer vision for scalable and user-friendly assistive communication systems [5].Therefore, this study aims to develop an efficient real-time sign language detection system that translates hand gestures into readable text or speech output, improving communication accessibility and supporting inclusive human-computer interaction.

I. RELATED WORKS

Authors [6] used a real-time American Sign Language (ASL) gesture recognition system that integrates MediaPipe for 21-point hand landmark extraction and CNN for classification. Their work demonstrated that landmark-based preprocessing significantly reduces the influence of background noise and illumination changes. The CNN classifier was trained on ASL alphabets and achieved an impressive 99.95% recognition accuracy, proving the efficiency of combining lightweight hand tracking with deep convolutional learning for real-time deployment. Their study mainly emphasized static alphabet recognition but laid a strong foundation for webcam-based sign translators.

Authors [7] further enhanced this approach by optimizing MediaPipe-based preprocessing and CNN classification. Their proposed framework focused on accurate hand localization, efficient gesture segmentation, and low-latency inference for real-time ASL recognition. The system achieved 99.12% accuracy, showing improved robustness in real-world environments. Their research highlighted that accurate landmark extraction before CNN classification directly improves recognition reliability and reduces computational overhead.

Authors [8] developed a MediaPipe and deep learning–based real-time sign detection framework that targeted practical communication support systems. Their study emphasized robustness under varying lighting and hand orientations. By combining landmark-based feature extraction with neural classification, they improved system usability for real-time scenarios. The work particularly contributed toward better preprocessing strategies for live sign capture applications.

Authors [9] introduced a dynamic sign language recognition framework using MediaPipe landmarks with recurrent neural networks (RNNs). Unlike static image-based models, their work focused on temporal dependencies in hand motion sequences, making it highly effective for dynamic words and sentence-level gestures. The RNN model captured frame-to-frame motion continuity and achieved very high recognition performance, proving the importance of sequential learning for real-time dynamic sign translation.

Authors [10] extended this concept by proposing a MediaPipe-LSTM enhanced framework for inclusive communication systems. Their approach utilized temporal hand landmark sequences extracted from video streams and modeled them using LSTM layers for improved context preservation. This method was particularly effective for continuous sign recognition tasks, where gesture transitions between frames play a significant role in classification accuracy.

Authors [11] explored CNN-based fingerspelling detection for Mexican Sign Language using MediaPipe frame preprocessing. Their work demonstrated that landmark-driven feature alignment can adapt effectively to regional sign languages. The proposed system was highly efficient for alphabet-level spelling recognition and showed the adaptability of CNN architectures across multilingual sign datasets.

Authors [12] proposed MP-GestLSTM, a real-time gesture detection framework combining MediaPipe with LSTM networks. Their work focused on achieving an optimal balance between recognition accuracy and processing speed, especially in dynamic gesture scenarios. The system effectively handled sequential hand motions and proved suitable for low-latency assistive communication devices.

Authors [13] developed an Indonesian Sign Language detection system using MediaPipe with LSTM. Their study extended real-time gesture recognition beyond ASL and demonstrated that the MediaPipe-LSTM combination generalizes well across language-specific gesture vocabularies. The proposed model improved recognition consistency for dynamic hand movements in real-time settings.

Authors [14] designed a neural-network-based web application for Pakistani Sign Language recognition, emphasizing practical deployment through web technologies. Their work highlighted the usability of deep learning sign recognition models in browser-based assistive tools, making sign translation more accessible and platform-independent.

Authors [15] introduced a more advanced 3D CNN + LSTM architecture for real-time ASL recognition. By using 3D convolutions, the model captured both spatial and short-term temporal motion patterns directly from video clips, while LSTM layers modeled long-term sequential dependencies. This hybrid framework significantly improved word-level recognition and was especially effective for continuous sign sequences.

Authors [16] proposed a hybrid CNN-LSTM sign recognition system based on MediaPipe. Their work combined landmark-based keypoint extraction with spatiotemporal deep learning, enabling better recognition of gestures with similar spatial structures but different movement patterns. The hybrid model demonstrated strong performance for real-time educational and healthcare applications.

Study [17] introduced MediaPipe Hands, which has become the backbone of most modern sign language recognition systems. It provides robust 21-point hand landmark detection with low latency, making it ideal for real-time gesture-based applications on CPUs, GPUs, and mobile devices. This framework serves as the preprocessing foundation in many CNN, RNN, and LSTM-based recognition pipelines.

Authors [18] introduced the OpenCV library, which remains essential for video frame capture, preprocessing, ROI extraction, noise reduction, and real-time visualization in sign language systems. It is widely used as the primary computer vision backend in webcam-based gesture recognition applications.

Authors [19] provided the practical deep learning foundation through Deep Learning with Python, where CNNs, recurrent models, and TensorFlow-based implementations are extensively discussed. These concepts are directly applicable to sign language classification pipelines involving spatial and temporal gesture learning.

Authors [20] established the theoretical backbone of this domain in Deep Learning, covering the mathematical principles of CNNs, RNNs, LSTMs, optimization, and generalization, which support the design of robust real-time sign recognition architectures.

II. METHODOLOGY

The proposed real-time sign language detection system is designed to recognize hand gestures captured through a PC webcam and translate them into meaningful labels in real time. The methodology combines computer vision, deep learning, and object detection techniques using OpenCV, LabelImg, TensorFlow Object Detection API, SSD, and CNN-based classification. The complete workflow consists of dataset creation, preprocessing, model training, real-time detection, and deployment. The first step of the proposed methodology involves the creation of a custom gesture dataset. A webcam connected to the PC is used to capture images of predefined hand gestures from multiple users. To ensure robustness, gesture images are collected under different angles, hand orientations, lighting conditions, and backgrounds. In this work, five different gestures are selected, and approximately 15–20 images per gesture are captured. The captured images are then manually annotated using LabelImg, an open-source graphical image annotation tool. For each image, a bounding box is drawn around the hand gesture, and the gesture is assigned a meaningful class label. The annotation process automatically generates an XML file in VOC format, which stores the coordinates of the gesture region and the corresponding label name. These XML files are essential for supervised training because they guide the object detection model regarding the exact location of the hand sign.

After dataset creation, preprocessing is applied to improve model performance. All images are resized into a fixed input dimension compatible with the SSD network. The images are further converted into grayscale or normalized RGB format depending on the selected training configuration. In addition, the dataset is shuffled randomly to remove any ordering bias before splitting. The complete dataset is divided into training and testing sets. Out of 15 images per class, 13 images are used for training and 2 images for testing, while in another configuration 20 images per class are divided into 15 training and 5 testing samples. This split ensures that the trained model can be evaluated on unseen gesture samples. For feature extraction and object localization, the proposed system uses the Single Shot Detector (SSD) deep learning algorithm. SSD is chosen because it performs object localization and classification in a single forward pass, making it highly suitable for real-time applications. Unlike Faster R-CNN, which uses a separate region proposal network, SSD directly predicts bounding box coordinates and class probabilities from multiple feature maps at different scales. This significantly improves inference speed while maintaining high detection accuracy. The SSD network learns gesture-specific features such as hand shape, finger positions, palm orientation, and gesture boundaries. The extracted dataset and annotations are trained using the TensorFlow Object Detection API, which provides a scalable framework for building and deploying object detection models. A pre-trained SSD model from the TensorFlow Model Zoo is selected as the base architecture and fine-tuned on the custom sign gesture dataset.

During training, the input images and XML annotations are converted into TensorFlow-supported record files. The training process optimizes the network weights through multiple epochs, enabling the model to learn gesture patterns effectively. The SSD backbone automatically extracts deep hierarchical features, while the classification head predicts the corresponding gesture label. In addition to SSD-based detection, a Convolutional Neural Network (CNN) is used as

a fine classifier for cropped hand regions. After hand localization, the detected gesture region is resized to 28 × 28 pixels and converted into grayscale format. The CNN architecture consists of:

- Input Layer
- Two 2D Convolution Layers
- Pooling Layer
- Flatten Layer
- Two Dense Layers
- ReLU Activation Function

The CNN is trained for 30 epochs with a batch size of 120, enabling efficient feature learning on GPUs. The convolutional layers extract local gesture patterns, while dense layers perform final classification into the predefined gesture labels. In the real-time phase, the trained SSD + CNN model is integrated with OpenCV for live webcam streaming. Each incoming frame from the webcam is passed to the trained TensorFlow detection pipeline. The SSD network first identifies the hand gesture and generates a bounding box around the detected region. The cropped hand area is then passed to the CNN classifier for fine-grained recognition. Finally, the predicted class label is displayed on the video frame in real time. This process continues continuously for every frame, allowing seamless gesture detection and instant prediction. For practical deployment, the trained TensorFlow model can be converted into a TensorFlow.js Graph Model. This enables deployment in web-based applications using React.js and HTML Canvas, where real-time detections can be visualized directly inside a browser. Such deployment improves accessibility and allows the system to be used as an assistive communication tool on multiple platforms.

III. RESULT & DISCUSSION

The experimental evaluation of the proposed real-time sign language detection system was carried out on a custom dataset created using a webcam and OpenCV. The dataset consisted of five predefined hand gestures, where each gesture class contained approximately 15 images captured from different angles, hand orientations, and lighting conditions. This resulted in a dataset that better represents real-world variability during gesture recognition. For supervised learning, the captured images were annotated using LabelImg software, where a bounding box was manually drawn around the hand gesture in each image. Each labeled image generated a corresponding XML annotation file in VOC format, which stored the coordinates of the gesture region and its class label. These XML annotations were used during training to guide the SSD-based object detection model toward accurate hand localization.

The complete dataset was trained using the TensorFlow Object Detection API with the SSD (Single Shot Detector) architecture for 10,000 training steps. Model checkpoints were periodically saved to ensure recovery from interruptions and to monitor progressive improvement in detection accuracy.

The performance of the proposed model was evaluated using standard object detection metrics, including localization loss, confidence loss, precision, recall, and Intersection over Union (IoU). The localization loss measures the mismatch between the ground truth bounding box created during annotation and the predicted bounding box generated by the SSD model during testing. In the proposed system, the localization loss converged to 0.05, indicating that the predicted gesture boundaries closely matched the manually labeled hand regions. The confidence loss represents the classification error in predicting the correct gesture class label. For the trained SSD model, the confidence loss was observed to be 0.19, which indicates reliable class-level discrimination among the five gesture classes.

The evaluation also included precision and recall analysis, where IoU (Intersection over Union) was used as the matching criterion. IoU determines the overlap ratio between the predicted bounding box and the ground truth annotation. A higher IoU confirms better localization quality and more reliable real-time gesture detection.

The training performance was analyzed over 10,000 optimization steps, during which the loss function consistently decreased. The decreasing trend of the total loss confirms that the model successfully learned discriminative gesture features from the dataset.

TensorFlow checkpoints played an important role during this process by storing intermediate model weights at regular intervals. This checkpoint mechanism ensured fault tolerance and allowed the model to resume training in case of system interruptions.

The learning rate schedule and continuous reduction in loss values demonstrate stable convergence of the SSD network. As the number of training iterations increased, the model became better at both gesture localization and classification, thereby improving overall recognition reliability. The trained model was integrated with OpenCV for real-time webcam-based testing. During inference, the SSD network successfully detected hand gestures and generated bounding boxes around the gesture region in live video frames. The predicted class label corresponding to the detected gesture was displayed instantly on the screen.

The system performed effectively under moderate variations in hand angle, background, and illumination, showing its suitability for practical deployment. The use of SSD enabled fast frame-by-frame inference, which is critical for real-time sign language communication systems.

The obtained results clearly demonstrate that the proposed SSD + TensorFlow + OpenCV framework is capable of performing efficient real-time sign language detection. The low localization loss (0.05) confirms accurate hand boundary

prediction, while the confidence loss (0.19) indicates strong gesture classification capability. The consistent reduction in training loss across 10,000 steps further validates the robustness of the proposed learning pipeline. Moreover, the use of a custom webcam dataset with multiple angles and lighting conditions improved the model's ability to generalize to real-world environments. However, the current study is limited to only five gesture classes, which restricts its vocabulary size. Future work can focus on larger multilingual sign datasets, dynamic gesture recognition using LSTM/Transformer models, and sentence-level translation systems for broader assistive communication applications.

IV. CONCLUSION

This research paper presented a real-time sign language detection system designed to reduce the communication gap between hearing-impaired individuals and the general community. The proposed framework combines computer vision, deep learning, and object detection techniques to recognize hand gestures directly from live webcam video streams. By integrating OpenCV for image acquisition, LabelImg for annotation, TensorFlow Object Detection API for training, and the SSD (Single Shot Detector) algorithm for gesture detection, the system is capable of identifying hand signs efficiently in real time. The study began with the creation of a custom dataset containing five predefined gesture classes, captured from multiple angles, lighting conditions, and background variations. Accurate annotation using XML-based bounding boxes enabled the SSD model to learn both gesture localization and classification simultaneously. Experimental results obtained after 10,000 training steps showed promising performance, with a localization loss of 0.05 and a confidence loss of 0.19, indicating precise hand boundary detection and reliable class prediction. The decreasing loss trend throughout training further validated the effectiveness of the proposed deep learning pipeline. A major contribution of this work is the use of the SSD-based direct word-level sign recognition approach, which improves speed compared to traditional alphabet-based digital translators. Existing systems often require users to gesture each alphabet separately, making sentence formation slow and inefficient. In contrast, the proposed model directly recognizes predefined sign words, significantly improving communication speed and usability. Overall, the proposed system demonstrates that SSD-based real-time sign language detection can provide accurate, fast, and scalable assistive communication support. The framework can be further extended in the future toward larger vocabularies, dynamic gestures, sentence-level translation, multilingual sign languages, and deployment on mobile or web-based assistive platforms.

REFERENCES

[1] M. N. Saiful, A. A. Isam, H. A. Moon, R. T. Jaman, M. Das, M. R. Alam, and A. Rahman, "Real-Time Sign Language Detection Using CNN," *International Conference on Data Analytics for Business and Industry (ICDABI)*, 2022.

[2] P. R. Vijaya, A. R, A. A. Reddy, H. R, K. Himansee, and S. K. A. Sattar, "Sign Language Recognition Using CNN," *14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023.

[3] R. Kumar, A. Bajpai, and A. Sinha, "Mediapipe and CNNs for Real-Time ASL Gesture Recognition," *arXiv preprint arXiv:2305.05296*, 2023.

[4] A. R. Verma, G. Singh, K. Meghwal, B. Ramji, and P. K. Dadheech, "Enhancing Sign Language Detection through Mediapipe and Convolutional Neural Networks (CNN)," *arXiv preprint arXiv:2406.03729*, 2024.

[5] A. Pandey, "Real-Time Sign Language Recognition Using Deep Learning and Computer Vision," *Technical Report, Parul University*, 2025.

[6] R. Kumar, A. Bajpai, and A. Sinha, "Mediapipe and CNNs for Real-Time ASL Gesture Recognition," *arXiv preprint arXiv:2305.05296*, 2023.

[7] A. R. Verma, G. Singh, K. Meghwal, B. Ramji, and P. K. Dadheech, "Enhancing Sign Language Detection through Mediapipe and Convolutional Neural Networks (CNN)," *arXiv preprint arXiv:2406.03729*, 2024.

[8] T. Chakraborty, S. Ghosh, A. Pal, and D. Ghosh, "Real-Time Sign Language Detection using MediaPipe and Deep Learning," *International Journal of Computer Applications*, vol. 187, no. 57, pp. 78–83, 2025.

[9] G. H. Samaan *et al.*, "MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition," *Electronics*, vol. 11, no. 19, p. 3228, 2022.

[10] N. Anithadevi, S. Palanisamy, S. S. Rubini, and S. Shrestha, "MediaPipe-LSTM-Enhanced Framework for Real-Time Dynamic Sign Language Recognition in Inclusive Communication Systems," *Engineering Reports*, vol. 7, no. 7, 2025.

[11] T. J. Sánchez-Vicinaiz *et al.*, "MediaPipe Frame and CNN-Based Fingerspelling Detection in Mexican Sign Language," *Technologies*, vol. 12, no. 8, p. 124, 2024.

[12] T. S. Varshini and P. Rukmani, "MP-GestLSTM: Real Time Gesture Detection using MediaPipe and LSTM," *Journal of Information and Telecommunication*, 2025.

[13] W. Utomo, Y. Suhanda, H. Ar-Rasyid, and A. Dharmalau, "Indonesian Language Sign Detection using Mediapipe with LSTM Algorithm," *Journal of Informatics and Web Engineering*, 2025.

[14] M. A. Khan *et al.*, "A Neural-Network Based Web Application on Real-Time Recognition of Pakistani Sign Language," *Engineering Applications of Artificial Intelligence*, vol. 135, 2024.
[15] D. Key, "Real-Time American Sign Language Recognition Using 3D CNN and LSTM: Architecture, Training, and Deployment," *arXiv preprint*, 2025.
[16] F. S. Takouchouang and H. T. Vinh, "Automatic Sign Language Recognition: A Hybrid CNN-LSTM Approach Based on MediaPipe," *arXiv preprint*, 2025.
[17] Google, "MediaPipe Hands: On-device Real-time Hand Tracking," Google Research, 2023.
[18] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
[19] F. Chollet, *Deep Learning with Python*. Manning Publications, 2021.
[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016

