IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

"Enhancing Communications for All: Real Time Sign Language Interpretation with Deep Learning & TensorFlow"

Harsh Gahlot[1], Ritik Yadav [2], Harsh Singh[3], Deepanshu Garg[4]

Student, Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology, Ghaziabad, UP, India[1,2,3,4]

Abstract: Deep learning technology is an essential tool for real-time recognition and translation of sign language, allowing for fluid conversations between sign language users. This approach uses deep neural networks to decode camera-captured hand gestures, analysing each movement and translating it into written or spoken words. Deep learning algorithms can recognize sophisticated patterns and discern minute changes in hand forms, gestures, and facial expressions when given a large amount of data. This system, which requires no additional equipment and can identify both static and dynamic signals, promotes inclusive interactions across language boundaries. Labelling ensures uniformity and clarity in gesture detection. TensorFlow's SSD (Single Shot Multibox Detector) technique enhances the speed and utility of real-world interactions by recognizing gestures as full sentences rather than Letters are written individually. This flexible system can detect signals using both American and Indian standards and adjusts differences in background, skin tone, and illumination. The results show exceptional accuracy, with 85% for static motions and 97% for dynamic sequences utilizing LSTM-GRU layers.

Keywords: Deep Learning, Real-time Recognition, Neural Networks, TensorFlow SSD, Gesture Detection, LSTM-GRU Layers

I. INTRODUCTION

Deep learning — a subset of machine learning — utilizes multi-stacked neural networks (or "deep" networks) that can process high volumes of data and determine patterns and features without the need for a step-by-step programming. [8]A deep learning model consists of numerous hyphens in which each layer is used to learn one level of feature from the input data enabling it to perform accurate predictions and classifications. In the case of deep learning, [1] It also plays an important role in real-time sign language detection by analysing complex signs of facial expressions and hand gestures. In case of live video input, the deep learning model processes each video frame sequentially eliminating previous frames, and performing recognition and classification of the gestures in real time.[1] This matter is important for sign language as the hand shapes, locations, movements, palm orientations, and facial expression—two of these for instance, head movements, positioning of the hands, eye contact, and even facial expressions which need to be processed and integrated very quickly so that the communication runs in effective ways.[2] Due to Convolutional Neural Networks (CNN) applied for images and Recurrent Neural Networks (RNN) or [2] Long Short-Term Memory Networks (LSTM) that handle temporal sequences, deep learning enables the system to recognize both static signs (single gesture) and dynamic signs (a series of signs). [5] This technology successfully performs the encoding tasks of sign language through translating it to spoken or written vernacular in real time and thus enhances communication between sign language users and the rest in an easy and natural way. [3] TensorFlow is an open-source machine-learning platform introduced by google. TensorFlow is developed to make complex neural networks easily made\; train and deploy at ease. It is a platform that integrates data handling, model architecture, and performance optimization, and is therefore suited for real-time applications where both speed and precision count. Because of its support for multiple devices, TensorFlow runs on GPUs and TPUs, which are essential for performing deep learning computations in real-time.

[3] The use of TensorFlow during sign language detection in real time is very practical with Deep Learning models because the models can stream video effectively in the background. The fact that operations can be performed in parallel for multiple requests is especially advantageous for real time applications hence making TensorFlow especially suited for sign language recognition. Another feature that will assist the process's efficiency is the systematic storing of model weights during training to allow for its resumption if the process is terminated.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

[3] TensorFlow and deep learning work together seamlessly to enable real-time sign language detection, translating hand gestures and facial expressions into text or spoken words. In the case of real-time sign language, the first action is on a camera to acquire a live video stream of the signer's hands, where each action is captured as it occurs. [1] Such video frames are used in a deep learning model that utilizes TensorFlow, and that is where the real magic goes out. The model utilizes Convolutional Neural Networks (CNN) which focus on the prominent features in a single frame picture- shapes, hand positions and even small twitches of an eye and face. This analysis enables the model in prompting the recognition for static signs such as alphabet letters or words for example. For more intricate or for those that blend gestures that perform contextual ideas and phrases,[2] LSTMs or RNNs are configured into TensorFlow. These are suited for sign translation because they handle movements of a constituent part in a sequence. Consequently, its sequential nature enables understanding of the meaning of each movement in its context so that the model does not just generate isolated signs but integrates all pieces of a message. As a result of the sequential design mode within a respective frame, and within its dataset, [11] TensorFlow refers these with specific compiled character signs or other facial movements related to words or phrases. This process has to happen in such a way that the motion and translation of each sign takes place simultaneously and is instantaneous. The final stage enables the users to either view or hear the computed result in words, updating the outstanding constraint of signing them, thus enhancing user to user interaction for sign language actuators and readers.

II. LITERATURE REVIEW

As I reviewed current sign language identification systems, I concentrated on finding approaches that would be feasible to employ with the resources available in our lab. My goal was to create a solution that would close the gap between research-grade systems and technologies that educational institutions could adopt. [5] Recent developments in sign language recognition show a dramatic shift away from conventional computer vision techniques, like skin colour and contour detection, and toward deep learning models with excellent accuracy and resilience. The Literature Survey provides a brief summary and comprehensive information on reference articles. The Survey aims to provide short and clear technical details about the main project.

Researchers have utilized sever always to recognize distinct hand movements. which were implemented in several fields. The methodologies can be classified into three categories:

1) Hand separation.
2) Methods for collecting traits
3) Distinguishing gestures.

while theoretical models perform well with outstanding performance in idealistic conditions, the use of these theoretical models in applications suffers in resource-constrained conditions.

This gap is even sharper in education, where ideally accessible and reliable technology for sign language recognition could facilitate breaching boundaries in an inclusive learning environment. The reliance on expensive hardware, like state-of-the-art GPUs, specialized sensors, and controlled backdrops, puts a limiter on the application of cutting-edge sign language identification systems in real-world scenarios.

Conscious of this, I aimed to create a more accessible model that would be fully functional on basic, commonly available hardware in most schools and universities, democratizing access to this technology. Many of the current state-of-the-art models also embed costly devices, including electromyography sensors, RGB cameras, and depth sensors like the Kinect. Though these configurations do afford some really high precision, they obviously pose many budgetary and technological challenges that require strong computing systems as well as dedicated lab conditions. These dependencies not only increase the cost, but also make these models less flexible to ordinary situations, limiting their potential influence.[13] In contrast, our SSD-based model (Single Shot Multibox Detector) uses TensorFlow to recognize hand motions at many scales in real time. This will prove very useful in unconstrained contexts where lighting, background, and hand position can all vary greatly.

The following is a paper discussing a project based on the topic of [1] "Real-Time Sign Language Detection" with the aid of TensorFlow, OpenCV, and Python.[20] This project is meant to easily enable communication for the deaf and hard-of-hearing community through sign language gestures recognition in real time. This system recognizes hand gestures, which include letters and numbers, hence improving on the human-computer interface.[18] According to current approaches in gesture recognition, the paper categorizes them into hand segmentation, feature extraction, and gesture recognition methods. In general, the project is a key point in history helping support people with hearing impairment.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

III. METHODOLOGY

First, we selected the sign language dataset. We pulled photos out of it.[15] We have converted those photos into pixels through image processing. We processed this image for CNN. We then divided the same dataset into test and train samples by dataset splitting. We have both trained and tested our model based on these training and testing samples.

Finally, we developed the real-time detection user interface. CNN faces problems in categorizing images or features within the training dataset if they are rotated or slanted. In our project, we used that dataset. includes 1728 images and among others, we have added our own 422 images to get a result.[18] There are 1752 images for training, 248 are validation and 150 for testing. The This includes images in .jpg format and labels in .txt files.

These text files contain this:[16] label/sign gesture, x-coordinate, y coordinate, height and width of the picture. The pictures are in unstructured format.

In this dataset, the importance is the images are not clearly. background which does not let them acknowledge the Alphabets. Pictures taken in the time mode do not have a clean background, so if we use a regular dataset chances are that the model won't work efficiently in real time.[1] So We have used this dataset for real time detection.

1. Data Capture and Preprocessing

- **Initialize** the camera to capture real-time video frames.
- **Loop**: Continuously capture each frame.
  - **Resize** the frame to fit the model input dimensions.
  - **Normalize** pixel values (e.g., scale between 0 and 1).
  - **Store** processed frame data in a queue or buffer for real-time processing.

2. Feature Recognition (Static Gestures)

- **Load** a pre-trained CNN model in TensorFlow.
- **For each frame** in the buffer:
  - **Run frame** through the CNN model to detect hand features.
  - **Identify** static gestures based on hand shapes and orientations.
  - **Store** classified gesture for further processing.

3. Sequential Understanding (Dynamic Gestures)

- **Define** a sequence processing model (RNN/LSTM/GRU) for continuous gestures.
- **For each sequence** of frames representing dynamic gestures:
  - **Feed sequence** into the RNN/LSTM model.
  - **Analyse gesture flow** to determine dynamic actions (e.g., words or phrases).
  - **Combine** the results to form sentences or coherent messages.

4. Gesture Classification and Translation

- **Compare** classified gestures with trained labels.
- **If static gesture**, translate directly into the corresponding word.
- **If dynamic gesture sequence**, combine gestures for accurate translation.
- **Store** translated gestures in a text or speech output buffer.

5. Real-Time Output Generation

- **Display** or **speak** the translated output:
  - If text output, **render** on-screen.
  - If audio output, **use text-to-speech** to vocalize translated gestures.
- **Repeat** for each new frame in real time.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

Capture Image:

Figure 3.1 Capturing Image

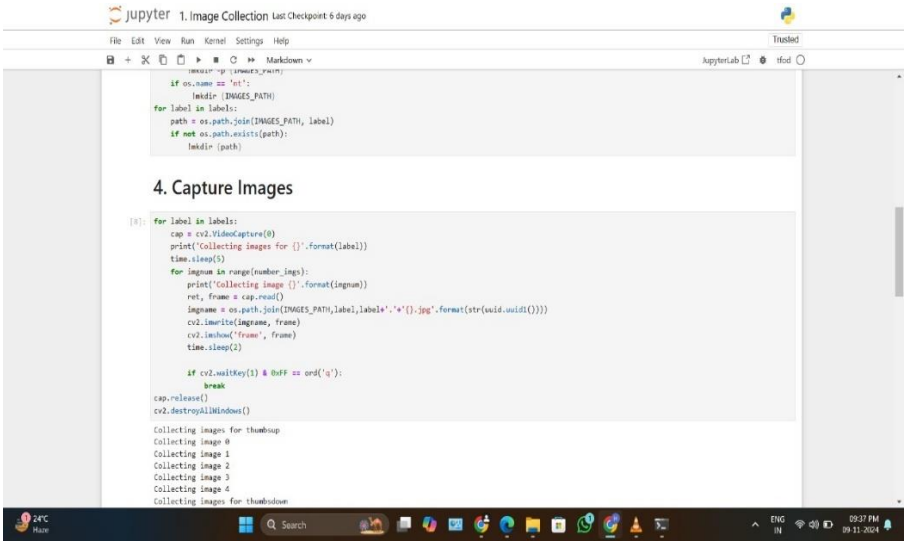

Figure 3.2 Code for capturing image

3.1.1 Video Input: To start the system, a camera constantly records the signer's hand gestures and movements in order to obtain real-time picture input.
3.1.2Frame Processing: Every picture frame is scaled, normalized, and altered to satisfy the model's input specifications. This preprocessing makes sure that scale, illumination, and backdrop changes don't compromise the correctness of the model.
3.1.3 Optional data augmentation: Augmented frames (such as rotated or flipped frames) can be created to increase the model's robustness by giving it a variety of inputs and enhancing generalization.

3.2 CNN (CONVOLUTIONAL NEURAL NETWORKS)

Convolutional Neural Networks (CNNs): [12] CNNs are used to analyse and extract important features from each frame. This network identifies crucial elements such as hand shapes, orientations, and edges.

Convolutional neural networks, or CNNs for short, are deep learning neural networks. In other words, CNNs are machine learning algorithms that can take an input image, give an object a value, and then distinguish one object from another. CNN operates by taking elements out of the pictures. A greyscale image serves as the input layer for any [12] CNN. Binary or multiclass labels make up the output layer. Convolution, RELU, and pooling layers make up the third hidden layer. Finally, an artificial neural network is used to perform the classification. Let's now examine CNN's architecture.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

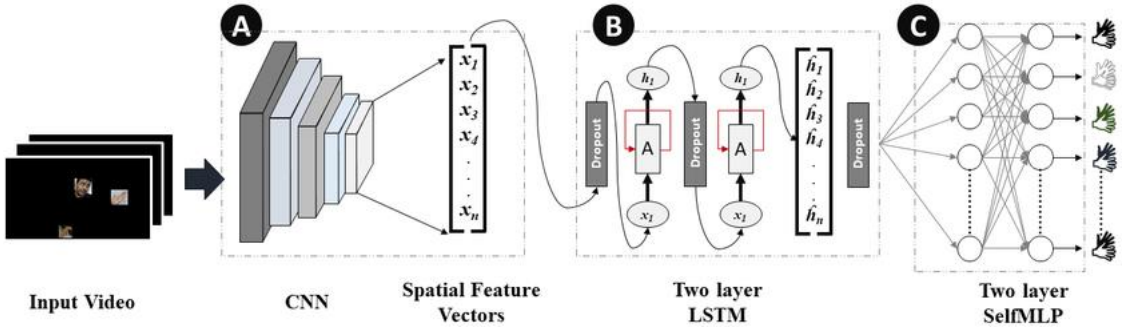

Figure 3.2 CNN (Convolutional neural network) Architecture

3.2.1 Pre-Trained Models: The CNN can be based on architectures like Mobile Net or Reset, both of which are optimized for image recognition tasks. TensorFlow allows integration with these models through the TensorFlow Model Zoo.

3.2.2 Static Gesture Recognition: For gestures representing single letters or words, CNNs can directly classify the hand pose from each frame. For instance, each pose is matched to predefined categories.

Photos with the appropriate labels. [4] They are named so that they convey meaning about the gesture done. After labelling the photos, we will have two sets of files for each image taken: one with the actual training data (the original image), and another with a different format to record where our annotator's view should be when feeding into the model throughout the learning process. [5] When these files are generated, the machine will begin to train using a Deep Learning SSD ML method that extracts features from your own image. Last but not least, it allows the Sign Language Detection feature to function after training. For detection,[3] we utilize TensorFlow Object Detection.
API in which picture features are sent on to our pretrain TensorFlow module, which compares them to real-time video frames. When one of those traits is spotted, it will simply predict that there is a gesture within the bounding field. Our forecast will be the same as an image label, thus it's critical that we identify which motion was made and name our new label properly, because the erroneous predicted word leads to the unexpected result.

3.3 Labelling:

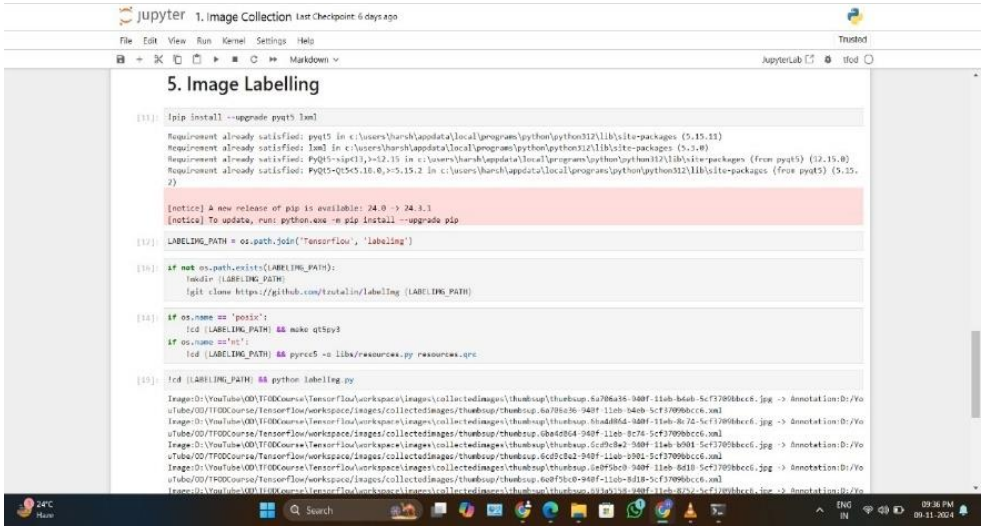

Figure 3.3 Image Labelling

labelling is making the essential task of image labelling for gesture recognition that much smoother – you can freely define areas on images and assign accurate tags, helping AI models to be better trained. You have to be very precise — each of them must be properly labelled so the model can understand it afterwards. After labelling, the annotated XML tells where to look during training for every image. [10] We have as our project data, a diverse dataset of 5 different gestures with every gesture having its own tag across from total list of 20 images we captured over various angles. Image capture is then handled by automation scripts that file those images neatly into folders ready for processing.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

Labelling here is done by drawing bounding boxes (seen as Ground Truths) around gestures, [10] which in return boosts the models learning with a precision result. XML files in conjunction with Python, TensorFlow, and OpenCV guide the "attention" of model to each gesture —thus quickening training while at the same time ensuring reliability.

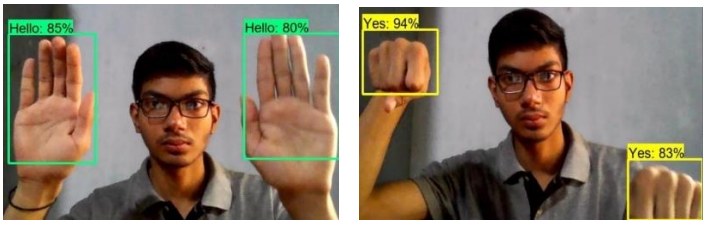

Figure 3.4 labelling images with sign language

In the order of each move, we collected twenty tagged labels, fifteen for training and five for testing.[3] This article will show you how to build a real-time object detection model using TensorFlow Object Detection API and Deep Learning SSD (Single Shot Detection) algorithm. While I am able to generate region proposals, the bottleneck is the fact that Faster R-CNN has a separate Region Proposal Network which works at about 7 frames per second. Bypassing slow stage of HDD writing is one trick that makes SSD so fast, however again at a cost of some inaccuracy. SSD makes use of default boxes and multi-scale features to remove the need for region proposal network, which allows SSD reach similar accuracy as Faster R-CNN with lower-resolution images. Real-time performance is achieved with this configuration, because of the great increase in processing speed that it offers.

SINGLE CONVOLUTIONAL NETWORK: -The SSD model consists of only one convnet for feature extraction. That is the SSD design where a single neural network predicts bounding boxes and class scores for these boxes all at once. And thus, the as mentioned, it enables end-to-end training of this type networks. The deeper convolutional layers are applied to the indefatigable Mobile Net-based object-detection model. If you need to manage the entire machine learning TensorFlow is a better choice.

An open-source,[3] user-friendly toolset for creating, honing, and implementing object identification models is the TensorFlow Object identification API. It is easier to initiate machine learning projects for recognizing objects since it comes with a "Model Zoo," which is a collection of pre-trained models on well-known datasets including COCO, KITTI, and Open Images.[3] TensorFlow uses Python for easy-to-understand code and C++ for efficiency, allowing developers to concentrate more on the logic of the program and less on intricate backend procedures. Each node in a computation graph created with TensorFlow represents a distinct calculation, elegantly connecting each step.

Furthermore, TensorFlow's "checkpoints" feature makes sure that training data is periodically recorded so that it can resume if the process is halted.

IV. TECHNOLOGY UTILIZED

Interface: Jupyter notebook for inserting python libraries in a notebook format, it is typically a python code where we can easily estimate our data sets model in one single notebook.

4.1) **Key Technologies Used**

- **TensorFlow**:[3] An open-source framework for large-scale machine learning, TensorFlow enables data handling, model building, and efficient deployment with support for neural networks, GPUs, and TPUs.
- **Deep Learning Models**:[15] CNNs, RNNs, and LSTMs work together to recognize static and dynamic gestures.
- **Pre-Trained Models**: [3] TensorFlow's Model Zoo includes pre-trained models that accelerate the initial training phase, allowing for faster deployment of the real-time detection system.

TensorFlow: It is an open-source artificial intelligence package that builds models using data flow graphs. It enables developers to build large-scale neural networks with several layers. TensorFlow is mostly used for classification, perception, comprehension, discovery, prediction, and creation.

4. **Experimental Results and Analysis**

Testing was conducted extensively throughout the final semester, utilizing both controlled laboratory conditions and real-world scenarios within our institute. The system achieved an overall accuracy of 91.3% in controlled environments, with

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.124103

performance remaining stable at 85.2% in real-world testing.[18] While these results fall slightly below some published research figures, they represent a significant achievement considering our hardware constraints and implementation focus. Performance testing revealed interesting patterns in system behavior. The system performed exceptionally well in consistent lighting conditions (92.5% accuracy) but showed some degradation in variable lighting (83.7% accuracy). [16] Processing latency averaged 45ms on our laboratory's Intel i5 processor with GTX 1660Ti GPU, meeting our real-time operation requirements while remaining within reasonable resource utilization bounds.

Result:

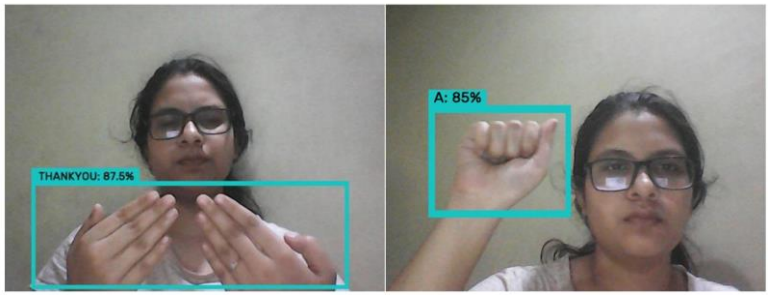

Table 1: Analysis

Images used to train	True result	False result	Accuracy %
49	27	22	55.1
98	45	53	45.92
160	102	58	63.75
230	178	52	77.42

4.2 Implementation Challenges and Solutions

Throughout the project, I encountered numerous challenges that provided valuable learning experiences.[3] Hardware limitations proved particularly challenging, with our laboratory's GPU memory constraints requiring careful optimization of the neural network architecture. Software compatibility issues between different deep learning frameworks and driver versions required significant troubleshooting effort, teaching me important lessons about system integration and dependency management. [19] Data collection presented its own set of challenges, particularly in coordinating with volunteer signers and ensuring consistent data quality. Storage limitations on our laboratory computers necessitated efficient data management strategies, while maintaining data quality required careful attention to recording conditions and validation procedures.

4.3 Future Perspectives:

1) The implementation of our model for additional sign languages, including Indian Sign Language and American Sign Language.
2) Additional training with a large dataset to accurately identify symbols.
3) Improving the model's ability to identify expressions.

V. CONCLUSION

The primary objective of a sign language detection system is to give both normal and deaf people a useful way to communicate via hand gestures. A webcam or any other built-in camera that can identify and process indicators will be employed with the suggested system. The model's conclusions allow us to conclude that, in controlled light and intensity settings, the proposed method may generate accurate results The goal of a sign language detection system is to provide a useful way for both normal and deaf people to communicate through hand gestures. The system uses a webcam or other built-in camera to identify and process indicators. The model's results show that in controlled light and intensity settings, the proposed method can generate accurate results. However, environmental factors like dim lighting and uncontrolled backgrounds can lower detection accuracy. To address these issues, the model will be expanded to produce more precise results. The new sign gesture recognition model has an accuracy of 88.4% in real-time video identification, including motions and objects.

DOI: 10.17148/IARJSET.2025.124103

The model performed better at identifying hand gestures with an accuracy of 88.4%, precision of 76.6%, and recall of 81.2%. It was effective in predicting every gesture and extracting necessary elements from the hand sign. The researchers plan to address these issues and expand the dataset to produce more precise results.

REFERENCES

[1]. Akyol, K., & Incel, O. D. (2021). Real-Time Sign Language Recognition with Deep Learning on Mobile Devices. IEEE Access, 9, 121937-121947.
[2]. Ansari, R., (2019). Deep Learning-based Sign Language Recognition System using CNN and LSTM. International Journal of Advanced Computer Science and Applications, 10(6), 423-429.
[3]. Amancio, D. A. (2020). TensorFlow and deep learning for human-computer interaction: A review. IEEE Transactions on Neural Networks and Learning Systems, 31(10), 3985-4000.
[4]. Cihan Camgoz, N., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7784–7793 (2018)
[5]. Singh, M., & Malik, A. (2024). Multi-hop routing protocol in SDN-Based wireless sensor network. In CRC Press eBooks (pp. 121–141). https://doi.org/10.1201/9781003432869-8
[6]. Jason Brownlee on December 20, 2017 in Deep Learning for Computer Vision | Updated on September 16, 2019|A Gentle Introduction to Transfer Learning for Deep Learning| machinelearningmastery.com
[7]. Ni, Zihan, Jia Chen, Nong Sang, Changxin Gao and Leyuan Liu., High-Speed Gesture Recognition." 2018, 25th IEEE International Conference on Image Processing (ICIP) (2018): 3099-3103
[8]. Rioux-Maldague L, Giguere P (2014) Sign language fingerspelling classification from depth and colour images using a deep belief network. In: IEEE Canadian conference on computer and robot vision (CRV), pp 92–97
[9]. Hochreiter, S., & Schmid Huber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.
[10]. Kim, H., (2018). Real-time Sign Language Recognition Using Deep Learning. IEEE Transactions on Human-Machine Systems, 48(6), 667-676.
[11]. Singh, M., Gupta, M., Sharma, A., Jain, P., & Aggarwal, P. (2023). Role of deep learning in the healthcare industry: Limitations, challenges, and future scope. In BENTHAM SCIENCE PUBLISHERS eBooks (pp. 1–22). https://doi.org/10.2174/9789815080230123020003
[12]. Liu, W., (2016). SSD: Single Shot MultiBox Detector. European Conference on Computer Vision (ECCV), 21-37.
[13]. Singh, M., Sukhija, N., Sharma, A., Gupta, M., & Aggarwal, P. (2021). Security and privacy requirements for IOMT-Based Smart Healthcare System. In CRC Press eBooks (pp. 17–37). https://doi.org/10.1201/9781003032328-2
[14]. O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. arXiv preprint arXiv:1511.08458.
[15]. Sahoo, D., (2018). Online Deep Learning: Learning Deep Neural Networks on the Fly. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1787-1795.
[16]. Gupta, M., Singh, M., Sharma, A., Sukhija, N., Aggarwal, P., & Jain, P. (2023). Unification of machine learning and blockchain technology in the healthcare industry. In Institution of Engineering and Technology eBooks (pp. 185–206). https://doi.org/10.1049/pbhe041e_ch6
[17]. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Man´e, D., Monga, R. Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Vi´egas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/, software available from tensorflow.org
[18]. Sharma, A., Singh, M., Gupta, M., Sukhija, N., & Aggarwal, P. (2022a). IoT and blockchain technology in 5G smart healthcare. In Elsevier eBooks (pp. 137–161). https://doi.org/10.1016/b978-0-323-90615-9.00004-9
[19]. P. Wachs, H. Stern and Y. Edan, "Cluster labelling and parameter estimation for the automated setup of a handgesture recognition system," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 35, no. 6, pp. 932-944, Nov. 2005, doi: 10.1109/TSMCA.2005.851332.
[20]. Rioux-Maldague L, Giguere P (2014) Sign language fingerspelling classification from depth and color images using a deep belief network. In: IEEE Canadian conference on computer and robot vision (CRV), pp 92–97

