International Journal of Artificial Intelligence & Machine Learning (IJAIML)
Volume 2, Issue 01, Jan-Dec 2023, pp. 7-22. Article ID: IJAIML_02_01_002
Available online at https://iaeme.com/Home/issue/IJAIML?Volume=2&Issue=1
Journal ID: 2157-0178, DOI: https://doi.org/10.17605/OSF.IO/KN92F

SCOPE DATABASE INDEXED

OPEN ACCESS

REAL TIME SIGN LANGUAGE TRANSLATOR FOR VIDEO CONFERENCING PLATFORMS

Khan Rehan
Student, Department of Computer Engineering,
Anjuman-i-Islam Kalsekar Technical Campus (AIKTC), New Panvel, Mumbai, India

Mullick Touhid
Student, Department of Computer Engineering,
Anjuman-i-Islam Kalsekar Technical Campus (AIKTC), New Panvel, Mumbai, India

Sayyed Ali
Student, Department of Computer Engineering,
Anjuman-i-Islam Kalsekar Technical Campus (AIKTC), New Panvel, Mumbai, India

Shaikh Danish
Student, Department of Computer Engineering,
Anjuman-i-Islam Kalsekar Technical Campus (AIKTC), New Panvel, Mumbai, India

Prof. Salim G. Shaikh
Assist. Professor, Department of Computer Engineering,
Anjuman-i-Islam Kalsekar Technical Campus (AIKTC), New Panvel, Mumbai, India

ABSTRACT

Sign Language is the method of communication of deaf and dumb people all over the world. There are about 70 Million sign language users around the world. But only a few percent of people who can hear and speak know sign language. This makes it difficult for deaf people to communicate. Computer-based Sign Language Recognition is a breakthrough technology to overcome this problem. After pandemic businesses and organizations have started adapting online video conferencing platforms for carrying out meetings, workshops, interviews, collaborations, etc.. The aim of this paper is to provide a practical solution for sign language interpretation. Here we propose a lightweight real-time and integrable sign language detection application, that can be used in any video conferencing platform such as google meet, microsoft teams, zoom, discord, etc. Here we have used deep learning algorithms, image processing and the concept of virtual cameras to achieve our goal. We describe a desktop application to sign language detection in the browser in order to demonstrate its usage possibility in videoconferencing applications. We use the MediaPipe Holistic pipeline and LSTM for pose detection and to train and predict sign languages. It shows 91%-93% prediction accuracy while the latency is still under 4ms.

Keywords: Sign Language Translation, American Sign Language, LSTM, Virtual Camera, Hand Gesture Recognition, OpenCV, video-conferencing, Sign Language Translator for Meeting Apps, Real Time Sign Language Translation, Google Meet, Microsoft Teams, Zoom.

Cite this Article: Khan Rehan, Mullick Touhid, Sayyed Ali, Shaikh Danish and Prof. Salim G. Shaikh, Real Time Sign Language Translator for Video Conferencing Platforms, International Journal of Artificial Intelligence & Machine Learning (IJAIML), 2(1), 2023, pp. 7-22.
https://iaeme.com/Home/issue/IJAIML?Volume=2&Issue=1

1. INTRODUCTION

Throughout history, individuals who are deaf have communicated using sign language. One of the earliest documented instances of sign language can be found in Plato's Cratylus from the fifth century BC. In the text, Socrates suggests that if people lacked a voice or tongue, they would resort to making gestures with their hands, head, and body to convey meaning, much like individuals who are nonverbal today.[1]

Pedro Ponce de Leon, a Spanish monk who lived from 1520 to 1584,[2] is recognized as the creator of the first manual alphabet. Some or all of the alphabet was influenced by the essential hand gestures used by monks who observed silence.

In 1620, Juan Pablo Bonet released a book titled "Reduction of letters and art for teaching the mute to speak" in Madrid,[3] which is widely recognized as the first contemporary work on sign language phonetics. This publication presents a plan for vocal instruction intended for the hearing-impaired and incorporates a manual alphabet.

Historical records indicate that sign language was utilized for a variety of purposes in Britain, including covert communication.,[4] communication with or by the deaf and public speaking.[5] Today, sign language is one of the most effective means of communication for the global deaf population. Unlike spoken languages, sign languages are complete, natural languages. In addition to using their native sign language, deaf individuals also use an informal form of sign language called international sign language when interacting with each other or when traveling. Although it has a smaller vocabulary than natural sign languages, international sign language is considered a pidgin variety of sign language.

Sign language is acknowledged and encouraged by the Convention on the Rights of Persons with Disabilities. It makes it clear that sign languages have the same status as spoken languages and obliges the state's parties to support the Deaf community's linguistic identity and make sign language learning easier.

In order to increase public awareness of the value of sign language in the full realisation of deaf people's human rights, the UN General Assembly has declared September 23 as the International Day of Sign Languages.

A World Federation of the Deaf study estimates that sign language is the first language of about 70 million Deaf people, in addition to many hearing people. In developing nations, they make up more than 80% of the population. More than 300 different sign languages are used by the group as a whole.

American Sign Language (ASL) is the primary sign language used by Deaf communities in the United States of America and most English-speaking regions of Canada. ASL emerged at the American School for the Deaf (ASD) in West Hartford, Connecticut in the early 1800s due to a language contact situation. Ever since, deaf community organizations and schools for the deaf have widely encouraged the use of ASL.

Technology advancements have made it possible to translate sign languages automatically using machines. Without the aid of a human interpreter, these technologies translate written or spoken language into sign language and sign language into written or spoken language. Hardware innovations like robotic hands with finger-spelling capabilities paved the way for automatic sign language translation.

The robotic hand known as RALPH, short for "Robotic Alphabet," was developed in 1977 and can translate alphabets into finger spellings.[6] Later, the widespread use of motion-detecting gloves led to the creation of projects like the CyberGlove and VPL Data Glove. With the aid of the computer software, the wearable hardware allowed for the capture of the signers' hand motions and shapes. However, as computer vision technology advanced, cameras took their place because they were more effective and placed fewer physical constraints on signers.[7]

Developers have faced challenges because spoken languages and sign languages have different phonological features. These days, developers use computer vision and machine learning to recognise particular phonological parameters and epenthesis[8] unique to sign languages, and speech recognition and natural language processing enable interactive communication between hearing and deaf people.

The reason we are using ASL for this paper is that it is widely used across various countries and there are a lot of datasets available to train our model.

With this increased hardware capabilities in computation power of computers and by using deep learning algorithms, image processing and the concept of virtual cameras it is possible to develop an application that has practical applications in real world scenarios.

1.1. Problem Statement and Objective

There are about 70 million sign language users around the world. However, only a small percentage of people who can hear and speak are familiar with sign language. It makes it difficult for deaf people to communicate. Computer-based Sign Language Recognition is a breakthrough technology to overcome this problem, but this technology has not been utilized properly based on opportunities available today. Technology has been instrumental in providing innovative solutions to address the problems caused by the COVID-19 pandemic. Trends in video conferencing clearly illustrate this point. After the pandemic, businesses and organizations have started adapting to online video conferencing platforms for carrying out meetings, workshops, interviews, collaborations, etc. This shift in technology has provided an opportunity to integrate these people with the rest of the world. A lot of research has been done to interpret sign language by utilizing computer technologies such as artificial intelligence. But none of the products available today provide any actual use case when it comes to real-world scenarios. The objective of this paper is to provide a practical solution for sign language interpretation. Here we propose a lightweight real-time and integrable sign language detection application that can be used in any video conferencing platform such as Google Meet, Microsoft Teams, Zoom, Discord, etc. Here we have used deep learning algorithms, image processing, and the concept of virtual cameras to achieve our goal. We describe a desktop application for sign language detection in the browser to demonstrate its usage possibility in videoconferencing applications.

1.2. Paper Scope

This paper's future potential is quite broad and has vast avenues for expansion. Creating a more comprehensive and user-friendly desktop application that gives users an easier accessible and user-friendly UI is one of the conceivable advancements.

Deploying the deep learning model on the cloud will enable us to train and update it more frequently without the need for users to download any update package. Also, because real-time data processing will be done in the cloud rather than on user devices, the application will be quicker and more effective. Furthermore, it can be used to provide a user authentication system for the login purpose of the user. Another fascinating feature is adding a text-to-speech module with a voice selection feature to improve communication between hearing-impaired and other people. This system has many use cases as it can also be used as a general translator. Different models can be added to this system to translate not only sign language but also spoken language as well.

It is difficult for people around the globe speaking different languages to communicate as traditionally it is not possible. But thanks to development in technology, we can provide them a tool that can be used to break this communication barrier at least in virtual meetings on different video conferencing platforms. However, video conferencing platforms do not have any language translators. There are many software available that provide translation facilities in virtual meeting apps, but none of them support most of the other platforms. This paper can emerge as a solution recommendation for all-language translator supporting most of the platforms available, as it already uses the concept of virtual cameras and with the help of virtual audio drivers we can achieve this goal using the same concept. It can be used as an all-language translator translating to the desired language that each user wants according to their preference.

For example, in an online meeting, the speaker is English-speaking, but most of the people are French-speaking. The speaker can use this app to output its voice to French in real-time, and suppose we have a participant who doesn't know English, they can use this app to hear this speech in their desired language in real-time.. It has a wide range of applications; it can be used in virtual meetings, conferences, seminars, e-learnings, collaborations, etc., where people around the globe gather together.

Overall, the future scope and real-world applications of this paper are very vast and diverse, with various opportunities for extension and expansion.

1.3. Technical Details

In this project we predominantly use the python programming language to develop this system. The reason we use python is because python offers concise and readable code. It also has many frameworks, and different pre-written libraries that can be used in the project. Here we use opencv to capture the video frames. We use the MediaPipe Holistic pipeline for pose, face and hand components detection and key points extraction. These key points are passed to a trained LSTM model for sign language detection. Text will be generated from the sign language. Text and video will be passed to the video conferencing platforms using virtual cameras. We manage to do all these processes while still working under 4ms.

1.4. Innovativeness Usefulness

This paper is a breakthrough concept to integrate impaired people to the rest of the world and to overcome language barrier problems at least in virtual meetings. Human Interpreters charge high fees and they are always not available and it is also not feasible to use them in virtual meetings . As of now there are no such tools available in the market that provide automatic translation services in video conferencing platforms and that have universal support for most of the platforms. This paper is very cost effective and is very simple to use. The user can use this tool to automatically translate sign language in real time without the need of any human interpreter. This tool universally supports most of the video conferencing platforms or any other live platforms that use cameras and microphones to connect people.

2. LITERATURE SURVEY

Paper Title 1: Real-Time Sign Language Detection using Human Pose Estimation.

The central theme of this paper is to develop a sign language recognition model that can operate in real-time directly from the web browser, specifically for video conferencing software. This model uses unidirectional LSTM with one layer and 64 hidden neurons on the DGS corpus dataset with 301 videos. It is observed that using a more diverse set of landmarks with a single point per hand, the system performs much better. Here, the parameters are increased by calculating 51, 842 features, with an accuracy of 91.53% and latency of 3537 ms. Both wrists have higher weights and are asymmetric, concluding that the selection is based on the user using the dominant hand for the gesture. 4,138 signing sequences totaling 11.35 seconds in duration make up the test set. Many mistakes might occur when testing the model. Using the Pose-All model resulted in fewer errors of most types except the bridge and skipped. Here the processing is done on the device, which is faster than processing on the videoconferencing servers. This paradigm has drawbacks since it can be too sluggish to use on mobile devices or devices without hardware acceleration like a GPU. According to tests, Google Meet, Zoom, and Slack can be tricked into thinking a person is speaking by broadcasting audio at 20KHz, which is inaudible to people, but if the audio is cropped to a range that is audible to humans, the system may fail.[9]

Paper Title II: Real-Time American Sign Language Recognition with Faster Regional Convolutional Neural Networks.

The article describes the creation and application of convolutional neural network-based ASL translators that can categorize spatial data. The proposed system contains a pre-trained GoogLeNet architecture that was developed using the ASL datasets from Surrey University and Massey University as well as the ILSVRC 2012 dataset. The pre-trained model is modified for this purpose via transfer learning to accommodate more specialized or niche data. CNNs and RNNs are used in the system to extract spatial and temporal characteristics, and a CNN is used to categorize ASL letters. To extract more pertinent characteristics from the frame, backdrop, and other body elements are eliminated after extracting the frames for each gesture from many video sequences. This method can be extended to sentence-level sign language translation, and future work can focus on combining the CNN and RNN models into a single model for better performance. The system shows promising results in classifying ASL letters with high accuracy, and further development could improve the system's overall performance.[10]

Paper Title III: Relevant Features for Video-Based Continuous Sign Language Recognition.

This study describes the development of a continuous sign language recognition system utilizing Hidden Markov Models (HMM). A vocabulary of 97 German Sign Language (GSL) signals was used in the system's architecture to enable it to detect phrases in sign language. The system employs feature vectors that represent manual sign settings as input for training and recognition, and beam search is used to reduce computing complexity while performing the recognition job. A monochrome video camera was integrated into the device to capture images, and the device utilized plain-colored cotton gloves to accomplish real-time picture segmentation and feature extraction. The researchers studied the effects of various characteristics on the identification outcomes. The experiments conducted revealed that the system's precision rate was 94%, utilizing a lexicon of 52 gestures and all available characteristics.

A unique Hidden Markov Model (HMM) represented each signal in the system. A language model will be incorporated into the system's recognition process in the future, and it is anticipated that the incorporation of syntactic and semantic data would further enhance recognition performance. The findings of this study are encouraging because they may allow deaf and hard-of-hearing people to continuously and automatically recognise phrases written in sign language, boosting their capacity to interact with the hearing community.[11]

Paper Title IV: Glove Based Sign Language Interpreter for Deaf and Aphonic Peoples

The aim of this study is to create a system utilizing a glove that can effectively translate communication for individuals who are deaf and mute, allowing them to interact with others. The user's hands and fingers may be tracked as they move and are positioned using the accelerometers included into the glove. The gadget can understand hand gestures and translate them into words by measuring these motions. Both hearing and speaking persons and deaf-mute people may communicate more affordably and conveniently with the help of this method. The initiative focuses especially on Indian Sign Language (ISL), where names or English phrases without a conventional sign are written out using fingerspelling. This system identifies the user's gestures as they are being made since it employs real-time signal input and is data-dependent. In this system, a microcontroller, an accelerometer, a speech module, an LCD, a speaker, and a regulated power supply are all employed as hardware elements. Communication between persons who are deaf-mute and those who are not can be facilitated by the glove-based interpretation system created as part of this research. The system's ability to detect complicated motions and signals can be expanded in the future, and it can also be integrated with other communication technologies.[12]

Paper Title V: Sign Language Translator

The approach outlined in this paper is intended to get over the communication difficulty experienced by deaf and mute people who are unable to verbally communicate. Instead, they use non-verbal communication techniques like sign language. Nevertheless, not everyone is able to comprehend sign language, which can make communication challenging. The technology can transform sign language fingerspelling into a common language that is simple for others to comprehend in order to solve this problem. When communication must happen fast if there is no translator present, this can be quite useful. The finger spells may be swiftly and correctly translated into words by the system, improving communication. Moreover, the system can fingerspell words that have been typed or printed. This function can be particularly helpful when conversing with deaf and mute people who are conversant in sign language but are unable to read. Effective communication between deaf and mute people and those who don’t understand sign language is made possible by this two-way communication channel.[13]

SR No	Author	Title	Method	Result/Accuracy
1.	Amit Moryossef (Google), Ioannis Tsochantaridis (Google), Roee Aharoni (Google)	Real-Time Sign Language Detection using Human Pose Estimation.	LSTM with one layer and 64 hidden neurons on the DGS corpus dataset with 301 videos.	91% accuracy using Pose-All Method with a latency of 4 ms.
2.	S. Dinesh, S. Sivaprakash, M. Keshav, K. Ramya	Real-Time American Sign Language Recognition with Faster Regional Convolutional Neural Networks.	Spatial features are extracted for individual frames using Convolutional Neural Network and temporal features using Recurrent Neural Network.	CNN was used to classify spatial features which obtained an accuracy of 93.33 %.
3.	Britta Bauer and Hermann Hienz	Relevant Features for Video-Based Continuous Sign Language Recognition.	Continuous density based Hidden Markov Models (HMM) with one model for each sign by using simple colored cotton gloves.	recognize continuous sign language with an accuracy of 94% based on a lexicon of 52 signs
4.	G R Meghana, Harshitha H S, L Shyla, Madan Kumar S M	Glove Based Sign Language Interpreter for Deaf and Aphonic Peoples	glove based deaf-mute communication interpreter system with glove internally equipped with accelerometer	Using Glove based accelerometer the accuracy is 98%
5.	Sujay R, Somashekar M, Aruna Rao B P	Sign Language Translator.	CNN and Computer Vision	CNN based model for classifying ASL with accuracy above 90%

Table 2.1 Literature Survey

3. PROPOSED SYSTEM

3.1. System Architecture

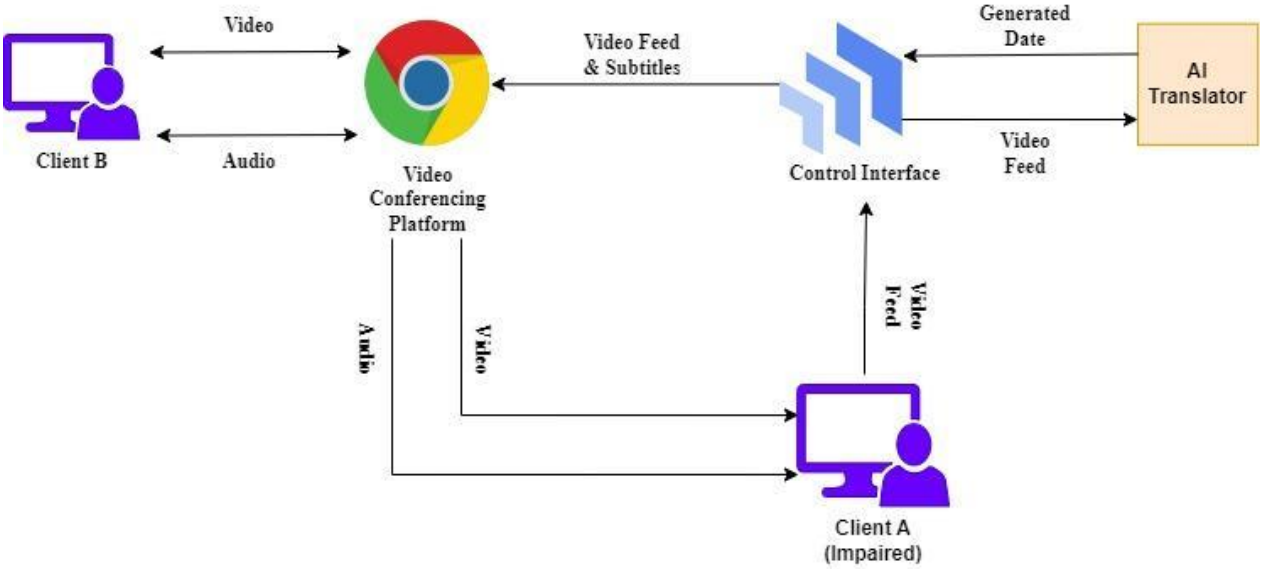

Figure 3.1 System Architecture

1. Select the virtual camera in the video conferencing app as input source.
2. Initialize the program when the user clicks start.
3. Program will read the video frames from the physical camera.
4. Predict the continuous sign language and generate sentences.
5. Draw the predicted action and sentence buffer onto the video frame.
6. Output the annotated video frame to the virtual camera, which will stream it to the video conferencing platform..
7. Continue the loop until the program is stopped or interrupted.

3.2. Methodology

Virtual camera driver

We utilize OpenCV for input video sources since it is a large library that provides many methods for image and video processing. We capture video from the camera using OpenCV. We designed a virtual camera. A virtual camera is utilized as an input source in the video conferencing platform. Our virtual camera's primary role is to integrate with any video conferencing platform.

Media-pipe Holistic

Sign language recognition simultaneously recognizes a person's posture, facial features, and real-time hand tracking [14]. Holistic is one of the pipelines with enhanced face, hand, and posture elements, providing holistic tracking where the model can recognize hand and position of the body alongside facial landmarks at the same time.

MediaPipe Holistic is used in the code to recognize and track multiple human body parts and their movements in real-time video streams. This enables the tool to capture hand gestures and movements during sign language and pass them through the neural network to identify the corresponding sign language phrase. By using MediaPipe Holistic, the tool is able to accurately capture the movements of the hands and other body parts, which is essential for accurate sign language recognition.

Algorithm for MediaPipe Holistic:

1. Import the necessary libraries, including MediaPipe and OpenCV.
2. Create a VideoCapture object to capture video frames from the webcam.
3. Create a PyVirtualCam object to send the processed video frames as virtual webcam output.
4. Create a MediaPipe Holistic object with specified minimum detection and tracking confidence thresholds.
5. Begin a loop to capture video frames from the webcam until the video stream is closed.
6. Within the loop, read a frame from the video stream using the VideoCapture object.
7. Pass the captured frame to the MediaPipe Holistic object for detection and tracking.
8. Extract the hand landmarks from the detected results.
9. Append the extracted hand landmarks to a sequence of the last 30 hand landmarks.
10. Once the sequence has 30 landmarks, pass it as input to a trained LSTM model for classification.
11. Obtain a probability distribution over all possible actions from the LSTM model.
12. If the probability of the predicted action is above a certain threshold, add it to a sentence buffer.
13. Keep the sentence buffer length at 5 or fewer words.
14. Draw the hand landmarks and any additional visualizations onto the captured frame.
15. Send the processed video frame as virtual webcam output using the PyVirtualCam object.
16. Continue the loop until the video stream is closed.
17. Release the resources and close all windows when finished.

LSTM model

A type of recurrent neural network (RNN) that can easily remember past inputs that is useful for processing sequential data. LSTM artificial neural network is mostly utilized in artificial intelligence and deep learning [15].

For LSTM we use tensorflow and keras. We import dependencies for LSTM that are sequential models, LSTM layer and dense layer. After building LSTM we train our LSTM Learning Model. Then we make sign language predictions to check the performance of a model after that we save our model weights.

Algorithm for LSTM model:

1. Define the model architecture as a Sequential model.
2. Add the LSTM layers to the model with the following parameters:

- LSTM(64, return_sequences=True, activation='relu', input_shape=(30, 1662))

- LSTM(128, return_sequences=True, activation='relu')

- LSTM(64, return_sequences=False, activation='relu')

3. Add two fully connected Dense layers with 64 and 32 units respectively and 'relu' activation function to the model.
4. Add the final Dense layer with a softmax activation function and the number of units equal to the number of actions in the dataset.
5. Compile the model with Adam optimizer, categorical_crossentropy loss function, and categorical_accuracy metric.
6. Train the model using the fit() method with the training data and labels, number of epochs, and TensorBoard callback.
7. Evaluate the model on the test data using the evaluate() method to get the test accuracy and loss.
8. Use the trained model to make predictions on new data.

Evaluation using a Confusion Matrix

After completion of our model, we evaluate the performance of our model by using matrices from Scikit learn that is a multi-label confusion matrix which is going to give us a confusion matrix for each one of our different labels and these allow us to evaluate what is being detected as a true positive and a true negative and what is been detected as false positive and a false negative. After evaluation the accuracy score of the model is between 91%-93%.

Algorithm for Evaluating a Model using Confusion Matrix and Accuracy:

- Import required libraries.
  - from sklearn.metrics import multilabel_confusion_matrix, accuracy_score
  - import numpy as np
- Predict the output using the trained model:
  - yhat = model.predict(X_test)
- Convert the true labels of the test data into a list:
  - ytrue = np.argmax(y_test, axis=1).tolist()
- Convert the predicted labels of the test data into a list:
  - yhat = np.argmax(yhat, axis=1).tolist()
- Compute the confusion matrix using the true labels and predicted labels:
  - cm = multilabel_confusion_matrix(ytrue, yhat)

- Compute the accuracy score using the true labels and predicted labels:
    - accuracy = accuracy_score(ytrue, yhat)
- Return the confusion matrix and accuracy score.

User interface & control panel

We made a control panel in the form of a desktop app to organize and handle streaming which is made by python Tkinter Gui. This app is responsible for controlling all user related settings including the different model selection.

3.3. Analysis

3.3.1. Framework

Media-pipe

A cross-platform library that delivers great ready-to-use Machine Learning and Deep Learning solutions for computer vision problems [16]. It is an open source framework by Google for processing time-series data such as video, audio, images and so on. It functions as an intermediary in the execution of system models.

TensorFlow

TensorFlow, a deep learning framework, was used to create, test, and operate our LSTM layers. TensorFlow is a prominent open-source toolkit for constructing machine learning and deep learning models that was launched in 2015 by the Google Brain team. It is based on the Python programming language and builds models by doing numerical computations on data flow graphs [17]. These are the key characteristics of TensorFlow. It performs well with multidimensional arrays. It enables computing scalability across machines and massive data volumes.

It allows for quick debugging and model creation. It has a big community and includes a Tensor-board to display the model [17]. Face identification, language translation, fraud detection, and video detection are some of its uses [17].

3.3.2 Algorithm

1. Import necessary libraries and modules (OpenCV, pyvirtualcam, mediapipe, etc.).
2. Set up the camera input stream and the virtual camera output stream.
3. Set up the mediapipe Holistic model for pose estimation.
4. Enter a loop to continuously read frames from the camera input stream.
5. For each frame:
    a. Make detections using the Holistic model on the frame.
    b. Extract the pose keypoints from the detection results.
    c. Add the key points to a sequence of the last 30 frames' keypoints.
    d. If the sequence now contains 30 frames:
        i. Use the trained model to predict the action being performed based on the sequence of keypoints.
        ii. If the predicted action is above a threshold probability:
            1. Add the action to a sentence buffer.
            2. If the sentence buffer is longer than 5 words, remove the oldest words.
        iii. Draw the current sentence buffer as text on the video frame.
    e. Send the video frame to the virtual camera output stream for display.

6. Exit the loop when the user presses stop on the control interface.
7. Release the camera input stream and destroy any OpenCV windows.

4. RESULTS

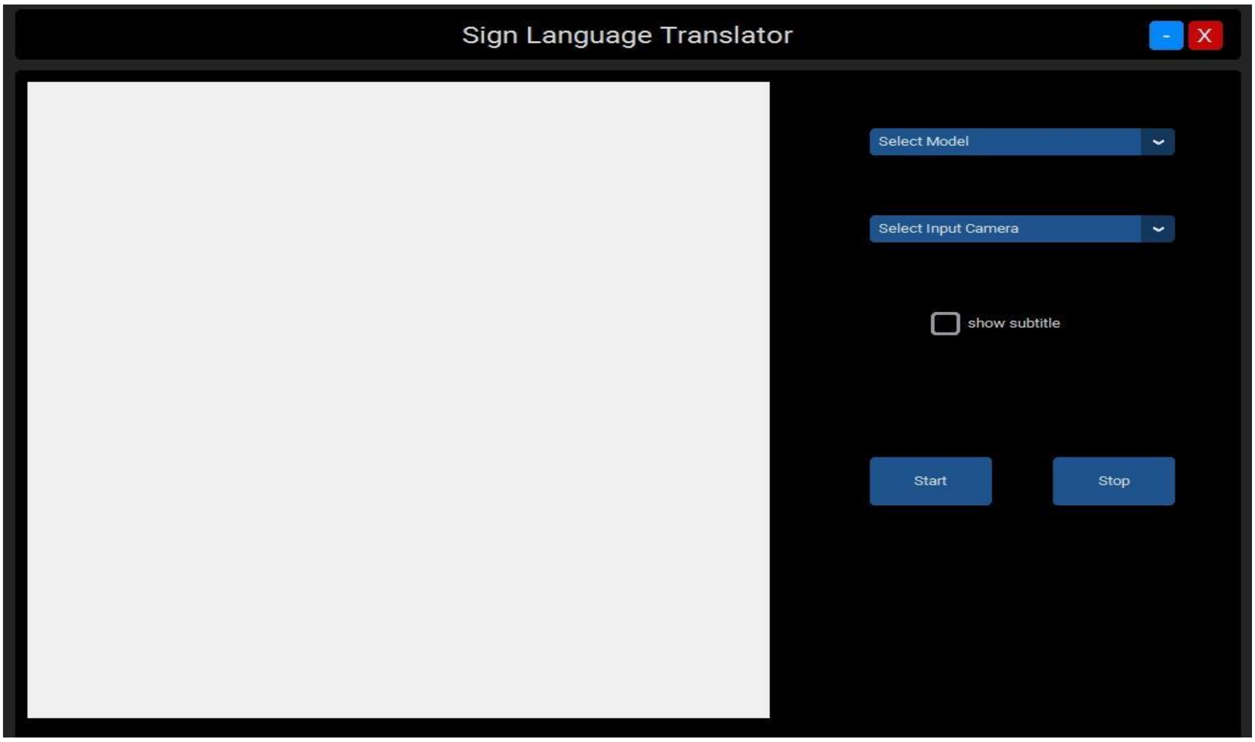

Figure 4.1 Control Interface

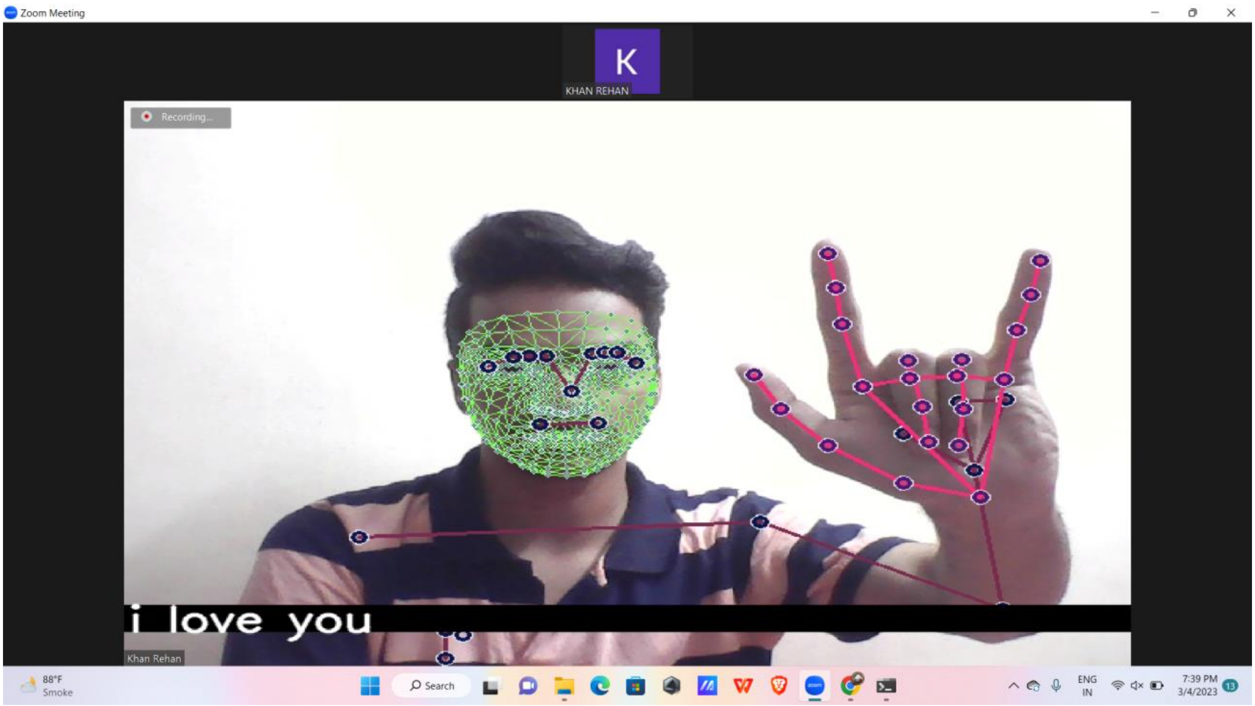

Figure 4.2 Hand gesture conveying “i love you” message at Zoom

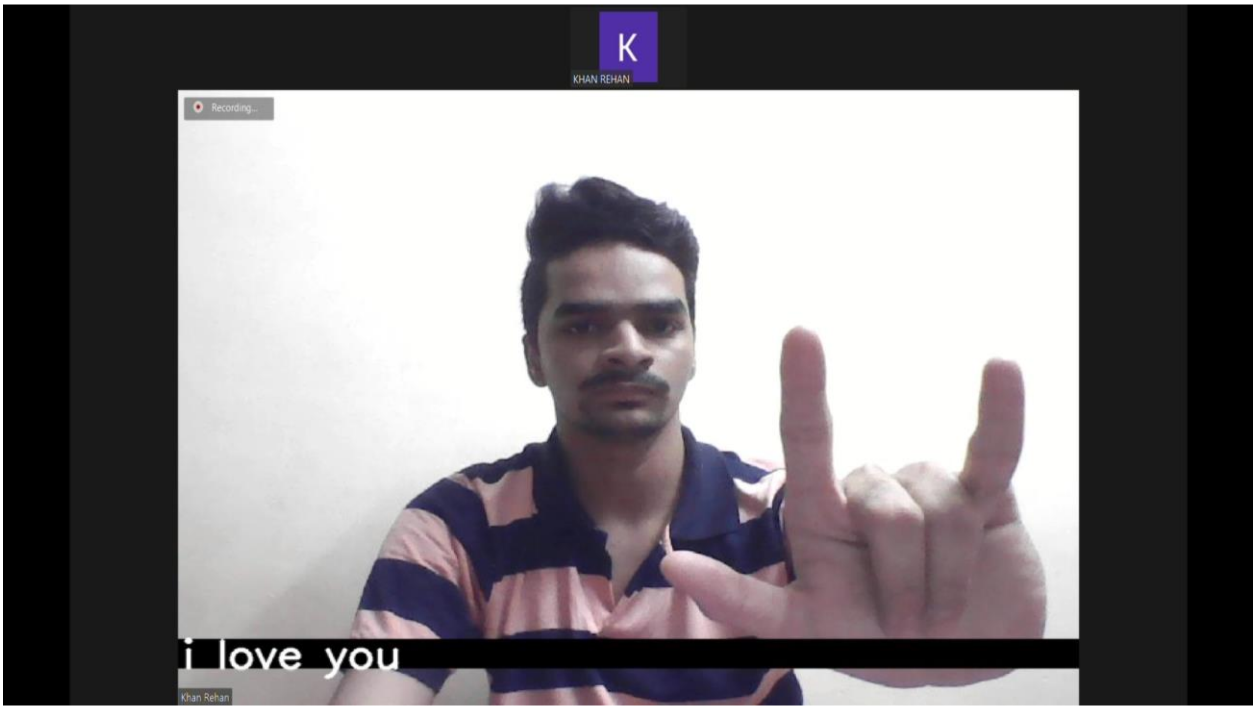

Figure 4.3 Hand gesture conveying "i love you" message at Zoom without landmarks

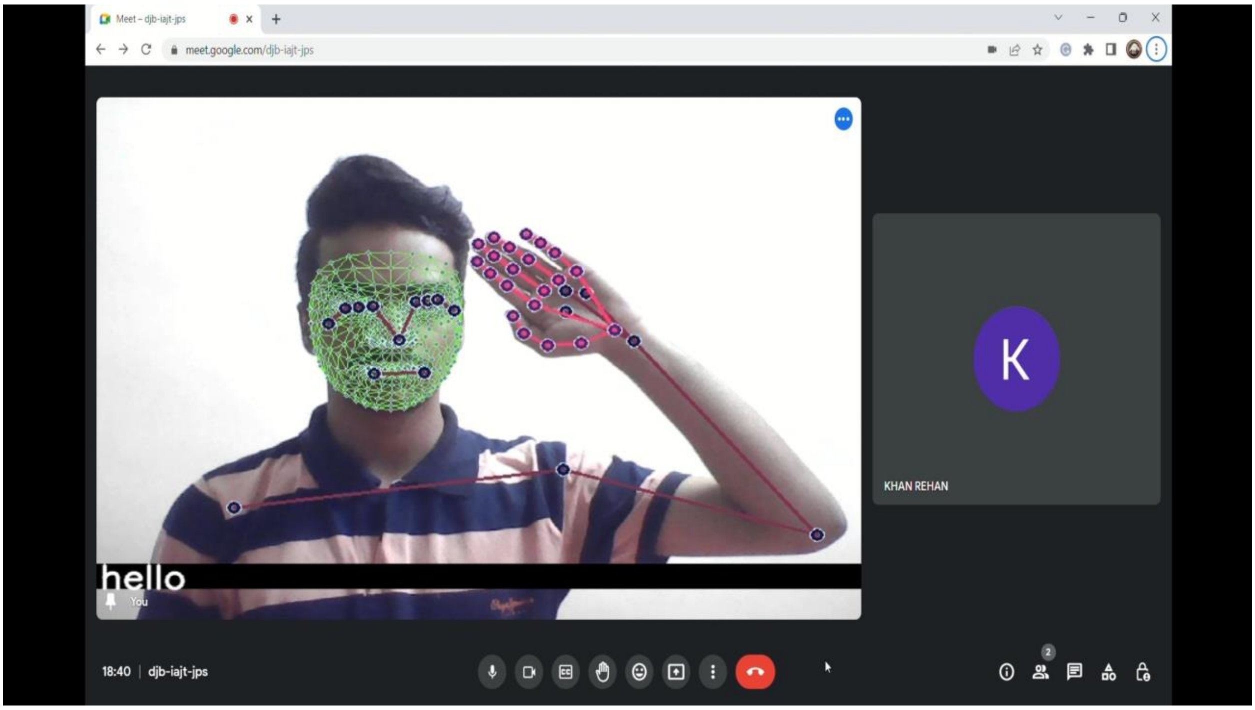

Figure 4.4 Hand gesture conveying "hello" message at Google Meet

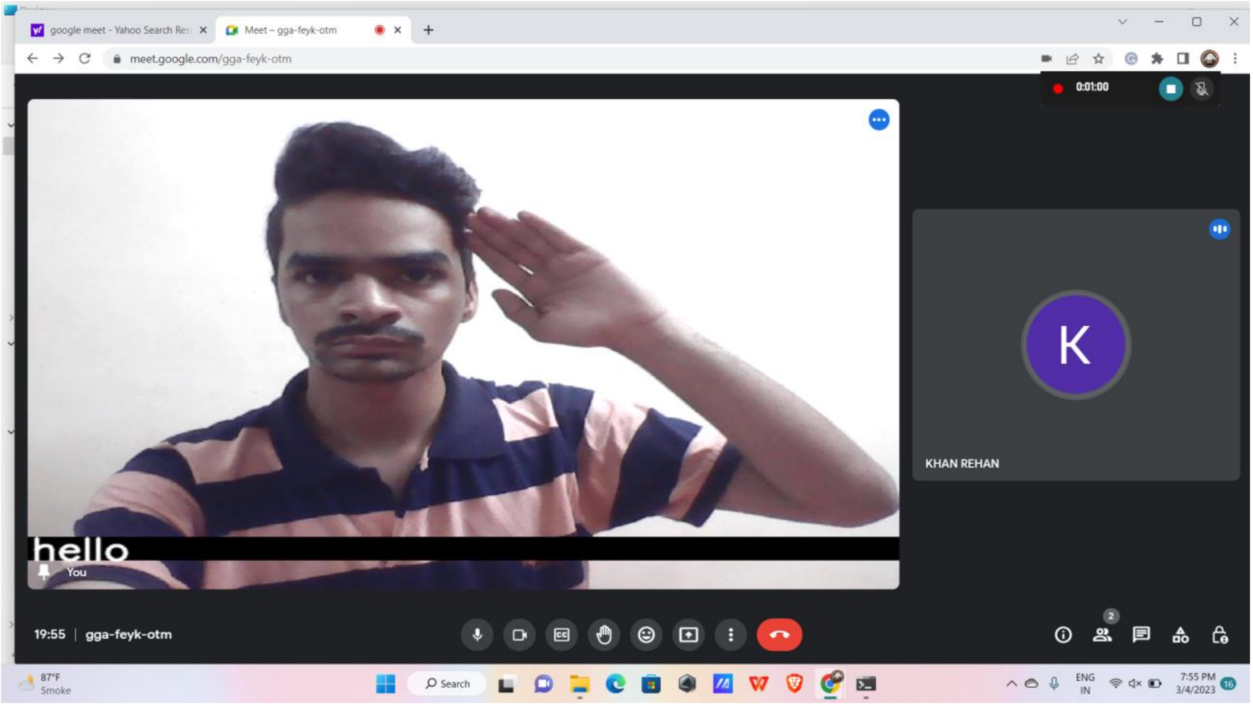

Figure 4.5 Hand gesture conveying "hello" message at Google Meet without landmarks

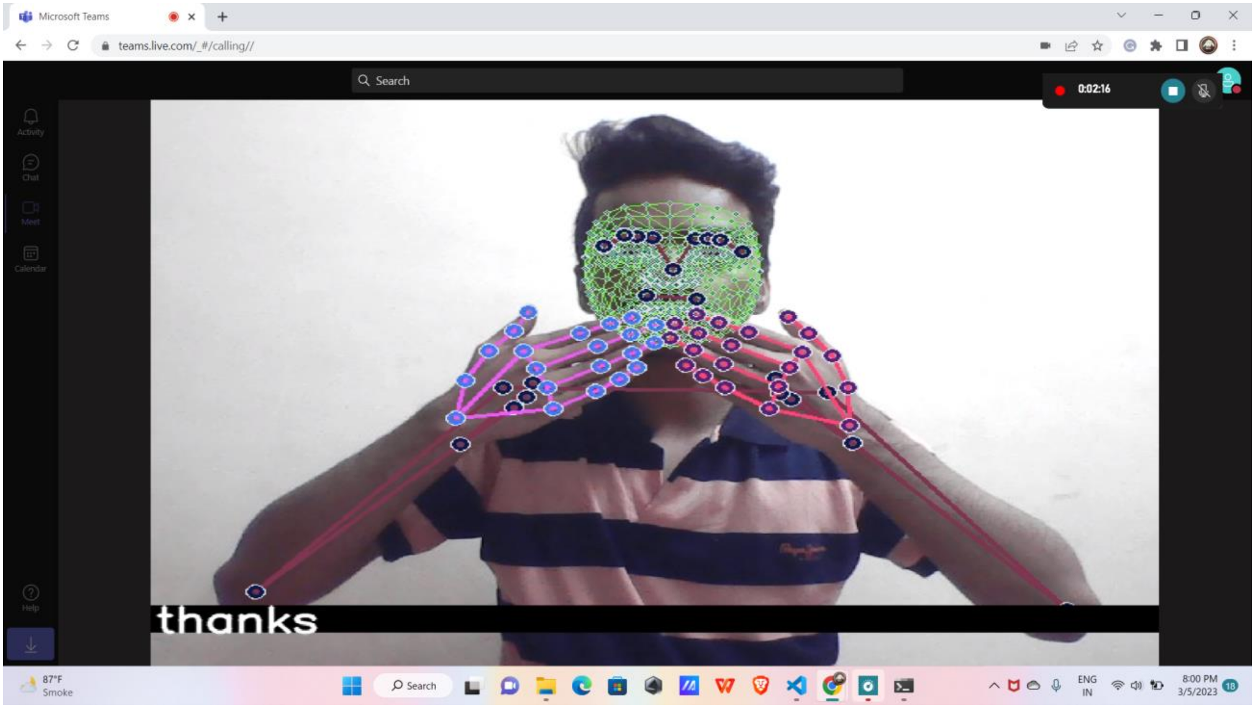

Figure 4.6 Hand gesture conveying "thanks" message at Microsoft teams

Figure 4.7 Hand gesture conveying "thanks" message at Microsoft teams without landmarks

5. CONCLUSION

In this paper we proposed an idea to implement real time sign language translation for video conferencing platforms using the concept of virtual camera. We have successfully implemented this idea of sign language translator in real time video conferencing platforms. We have integrated and tested this translator with different platforms such as Google Meet, Zoom, Microsoft Teams, Discord, Skype, etc. By adding different translation models this system can be used for spoken language translation also. We use the MediaPipe Holistic pipeline for pose, face and hand components detection and key points extraction. We use LSTM deep neural networks to train and predict sign languages based on actions of the user. It shows results of 91%-93% prediction accuracy depending on the input.

All components are working correctly and the end result can be obtained still under 4ms. However, latency depends upon the internet speed of the user and the system should have minimum processing power to process these real time data unless deployed in a cloud server.

6. ACKNOWLEDGEMENT

This research work was supported by our guide Prof. S.G.Shaikh. The authors wish to thank him for his kind support and valuable suggestions. We would like to show our gratitude to Prof. S.G.Shaikh for their assistance.

REFERENCES

[1] "Open your eyes: Deaf studies talking," Dirksen Bauman (2008). This publication is available through the University of Minnesota Press and has an ISBN of 978-0-8166-4619-7.

[2] Kim Nielsen (2012) published "A Disability History of the United States," a book that examines the history of disability in America. The book is available through Beacon Press and has an ISBN of 978-080702204-7.

[3] "Reduction de las letras y Arte para enseñar á ablar los Mudos" this book was written by J. de Pablo Bonet in 1620. This book is regarded as the first modern treatise on sign language phonetics and features a manual alphabet for the hearing-impaired. A facsimile of the original book is available online (in Spanish) through the “Miguel de Cervantes" Virtual Library.

[4] In his work on cryptography titled "Mercury, the Swift and Silent Messenger" (1641), John Wilkins discussed fingerspelling as a method of secret communication using sign language. Wilkins also provided an example of a fingerspelling system.

[5] "Chirologia: or the natural language of the hand," a book by John Bulwer, was published in London in 1644. Bulwer mentions in this book that deaf individuals use alphabets, although he presents a different system that focuses on public speaking.

[6] Jaffe, DL (August 1994). "Evolution of mechanical fingerspelling hands for people who are deaf-blind". This paper was published in the Journal of Rehabilitation Research and Development. 31 (3): 236–244. PMID 7965881.

[7] "Sign Language Recognition and Translation: A Multidisciplined Approach from the Field of Artificial Intelligence". This paper was written by Parton, B. S. (12 October 2005) and was published in the Journal of “Deaf Studies and Deaf Education”. 11 (1): 94–101. doi:10.1093/deafed/enj003. PMID 16192405.

[8] Boris Mocialov, Helen Hastie, Graham Turner, Katrin Lohan (2017). "Towards Continuous Recognition of Sign Language Using Deep Learning" (PDF). Robot Assistants and Meaning: The Gap Smart Devices Leave" (IEEE-RAS International Conference on Humanoid Robots). S2CID 5525834.

[9] Moryossef, A., Tsochantaridis, I., Aharoni, R., Ebling, S. and Narayanan, S., (2020), August. Real-time sign language detection using human pose estimation. In the European Conference on Computer Vision (pp. 237-248). Springer, Cham.

[10] S. Dinesh, S. Sivaprakash, M. Keshav, K. Ramya (2018). Real-Time American Sign Language Recognition with Faster Regional Convolutional Neural Networks.

[11] Bauer, B., & Hienz, H. (2000). Relevant features for video-based continuous sign language recognition. In Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (pp. 440-445). doi: 10.1109/AFGR.2000.840672.

[12] G R Meghana, Harshitha H S, L Shyla & Madan Kumar S M (2020). Glove-Based Sign Language Interpreter for Deaf and Aphonic Peoples.

[13] Sujay R, Somashekar M & Aruna Rao B P (2022). Sign Language Translator.

[14] Holistic," mediapipe. https://google.github.io/mediapipe/solutions/holistic.html

[15] At the 2017 IEEE (WACV), a team consisting of H. Gammulle, S. Denman, S. Sridharan, and C. Fookes introduced a deep fusion framework named "Two Stream LSTM". This framework was specifically developed to recognize human actions. http://dx.doi.org/10.1109/wacv.2017.27

[16] "Mediapipe OpenCV," GeeksforGeeks, Sep. 16, 2021. https://www.geeksforgeeks.org/face-and-hand-landmarks-detection-using-python-mediapipe-opencv/

[17] A. Biswal, "What Is TensorFlow 2.0: The Best Guide to Understand TensorFlow," Simplilearn, Dec. 10, 2020. https://www.simplilearn.com/tutorials/deep-learning-tutorial/tensorflow-2

Citation: Khan Rehan, Mullick Touhid, Sayyed Ali, Shaikh Danish and Prof. Salim G. Shaikh, Real Time Sign Language Translator for Video Conferencing Platforms, International Journal of Artificial Intelligence & Machine Learning (IJAIML), 2(1), 2023, pp. 7-22

https://doi.org/10.17605/OSF.IO/KN92F

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJAIML/VOLUME_2_ISSUE_1/IJAIML_02_01_002.pdf

editor@iaeme.com