WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Conversion of Sign Language to Words

Prof. Ryan Dais[1], Devineni Hemanjan [2], P. Giridhar Reddy[3], Repakula Siva Saikundan[4]

[1]Department of ECE Jain (Deemed to be) University Bengaluru, India ryan.dias@jainuniversity.ac.in
[2]Department of ECE Jain (Deemed to be) University Benguluru, India hemanjandevineni@gmail.com
[3]Department of ECE Jain (Deemed to be) University Benguluru, India giri.parlapalli1@gmail.com
[4]Department of ECE Jain (Deemed to be) University Benguluru, India repakulasivasaikundan11@gmail.com

ABSTRACT—

Human life has always depended heavily on communication. There are many languages used for communication throughout the world Nevertheless, communication can be challenging for those who have lost their capacity to hear and speak due to accidents or genetics. People with hearing loss have found sign language useful for social interaction. People with hearing loss have found sign language useful for social interaction. Social media and the internet can be difficult for people with hearing impairments to use to meet new people and start new relationships. For hearing-impaired people, a freely available video-calling tool that can translate sign language is quite useful. In an effort to bridge the vast communication gap, sign language recognition (SLR) has received a lot of attention. But as compared to other activities, sign language is much more unpredictable and complex, making it difficult to reliably recognise. The Speech-to-Text API enables those who can read but have trouble speaking to understand others. By converting their sign language into text that other people can comprehend, the Sign Language Translation Application (SLTA) enables individuals to communicate. The suggested method identifies the sign motion in real-time by utilising Python, the MediaPipe Framework for gesture data extraction, and the Deep Gesture Recognition (DGR) Model. Using a neural network with Long-Short Term Memory units for sequence detection, the suggested solution has the best accuracy of 98.81%.

Keywords—American Sign Language (ASL), Neural Networks, Deep Learning, Web- Real-Time Communication (RTC), Long-Short Term Memory (LSTM), Gesture Recognition, Sign Language Translation.

I. INTRODUCTION

Human communicate with one another using a variety of gestures and dialects. Between 50,000 and 150,000 years ago, the initial language began to develop. Since that time, voice communication has been essential for human interaction. Not everyone has the ability to hear and speak well enough to understand others. Many people around the world have difficulty hearing and communicating, and they are labelled as deaf, dumb, or hearing impaired. Around 300 million people worldwide, including 18 million Indians, have hearing impairments. Due to their limitations, it is challenging to converse with these individuals. The inability to hear makes it difficult for hearing-impaired people to form new relationships. These people use sign language to communicate with each other. There are many different types of sign language, including American Sign Language (ASL) [20], Chinese Sign Language (CSL), and Arabic Sign Language. The most well-known and popular sign language is American Sign Language (ASL), which is used in many nations. There is a large communication gap when using sign language to interpret and comprehend one another and to communicate with those who have hearing impairments, everyone should be conversant in sign language impairment. The communication gap can be reduced by developing a tool that converts this sign language into the language of choice and vice versa. The goal of this research is to create a system that facilitates communication via video conferencing [7][15]. The present research employs deep learning approaches to gather, train, and test data utilising open-source technologies like TensorFlow, Keras, NumPy, MediaPipe Framework, OpenCV, etc. to construct the Sign Language Recognition (SLR) system.

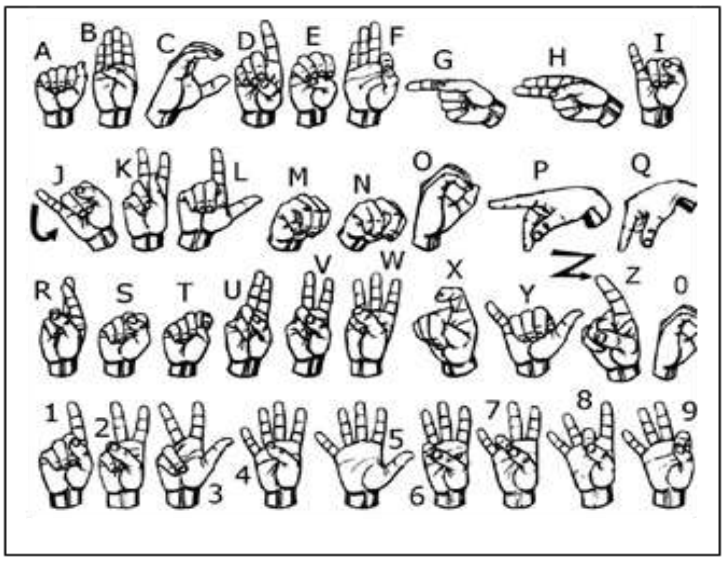

Fig. 1.1. Alpha-Numeric Hand Gestures of American Sign Language

In the first section of the review, the use of sign language is explained, along with how platforms that are open-source use SLR services, the tools employed, research that aids in understanding SLR model development, and conclusions drawn from it.

The definitions and analyses of the requirements necessary for creating the model and the software integration tools utilised in this study are included in the second section of the review, along with an overview of software requirements specifications (SRS).

The final section of the analysis explains the suggested approach and goes over software creation and deployment. Description, performance, test findings, and a plan for system management to meet system requirements are all explained.

II. RELATED WORK

Over the past few years, researchers and computer professionals have thoroughly investigated this problem statement in order to find a solution. All of their solutions range from looking at various gesture recognition techniques and patterns to analysing various sign languages and data collection methods.

The method is said to use "canny edge detection, which yields a more accurate result in identifying edges to determine the edges of hand signals in the frames, according to the paper by researcher Yeresime Suresh [21]. As opposed to using embedded sensors in gloves to gather data for the recognition of gestures, as seen in [3], the recommended system also makes use of the model for prediction of convolutional neural networks' (CNN) translation of spoken words and hand signals, as illustrated in figure 1.1. When trained on a large dataset, the results from both CNN and Canny Edge Detection are trustworthy and accurate.

Karly Kudrinko [14] looked into the viability of employing wearable sensor-based gadgets to recognise hand gestures in a sign language-related application. Her review looks at previous research to find trends and the best methods. The review also draws attention to the challenges and gaps that the sensor-based SLR field is currently facing. Without the need for sensor-based wearables, our study may contribute to the development of improved SLR [17] systems. By examining several study approaches, a standardised data collection protocol and evaluation procedure might also be created for her field.

In order to enhance the continuous SLR model, Mathieu De Coster's [13] study investigated a number of neural network topologies, including Hidden Markov Models (HMM), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN). OpenPose is a framework that collects the skeleton motion of gestures as a feature extractor. OpenPose is the only full-body pose estimation algorithm utilised to estimate gesture action because SLR [17] depends on hand form, placement, orientation, and non-manual elements like mouth shape. Other pose estimation techniques exist, however they only pinpoint specific critical spots, such as those on the body or hands (Fang et al., 2017). (2018) Mueller et al. Utilising the OpenPose Framework, he obtained the data, trained the model, and created the model.

Bhushan Bhokse developed a gesture recognition programme in his study that enables a user to display his hand doing a specific move in front of a video camera that is connected to a computer. His movements must be captured on camera so that the computer programme can analyse them and identify the sign. To make the system more manageable, it was decided that identification would comprise counting the user's fingers and identifying the American Sign Language they employ in the input image [12]. In his tests, he employed static images on plain backdrops, extracted the images as grayscale images, and then used the binary data from the photo to detect gestures.

In his research, Bhushan Bhokse created a programme for gesture recognition that allows a user to demonstrate his hand doing a certain motion in front of a camera with video capabilities that is attached to a computer. He must be seen moving on camera for the computer programme to be able to assess and recognise the sign. It was determined that identification would involve counting the user's fingers and determining the American Sign Language they

use in the input image in order to make the system more manageable [12]. Using static photos on plain backgrounds, he retrieved the pictures as grayscale images for his testing, after which he utilised binary information from the photo to detect motions.

In his paper [19], Zhibo Wang reviewed earlier research and system evaluations on a variety of SLRs with wearable sensors, including RF-based [5], PPG-based [9], acoustic-based [6], sensing gloves [8], vision-based [2] [4], EMG-based [9] [10] [3], and SignSpeaker [1]. These tests are compared to his DeepSLR work, which is based on a multichannel CNN architecture and generates findings that require less than 1.1 seconds to identify signals and recognise a sentence with four sign words, illustrating DeepSLR's recognition efficacy and real-time capability in practical settings. For continuous sentence recognition, the average word mistake rate is 10.8%.

In this proposed system, the issues with the current systems, such as model prediction dependence on sensor gloves and static indicators to anticipate the words, as seen in figure 1.1, will be resolved. By adopting an intelligent framework that can extract essential aspects of the gesture skeleton structure using a digital camera typically incorporated in any personal computing (PC) device, this system might lessen its need for sensor gloves. With the MediaPipe Framework created by Google, which can map the skeleton of the human being and retrieve coordinates of those crucial point locations for gesture identification, action holistic (skeletal system) data may be extracted. This can be used in conjunction with several frames to produce a motion sequence that represents a sign language word. LSTM algorithmic neural network topologies might work better for motion recognition, according to the literature review. The LSTM technique is well-known for its memory units, which display neural network nodes and allow them to recall the results of earlier data predictions. Given that motion gestures make up the majority of ASL in SLR applications, the storage barrier in the LSTM provides an important algorithm. The hurdle of using merely static images to predict signs is overcome through the application of the DGR model with LSTM architecture. It entails broadening the vocabulary and improving the model's adaptability and usability under various circumstances.

These anticipated words can be turned into sentences using the NLP model. Sentences will be shown as captions in the UI, and the NLP model will add significance and transfer it to the WebRTC communication channel to converse seamlessly. The spoken-to-text paradigm can assist the hearing impaired in reading and understanding by turning spoken recordings into text.

III. PROPOSED SYSTEM

The proposed system, which is American Sign Language (ASL) for interaction, aids people who struggle with communication by converting ASL for those who don't have access to sign language conversation. WEBRTC protocols are used by the video-conferencing programme SLTA to enable two-way, real-time audio as well as video communication. It is similar to Google Meet, Zoom, and Microsoft Teams, which are embedded with DGR and NLP models that help in SLR [17]. Development of such a system comes with specific challenges like creating visual motion gestures, the vocabulary of a language, training a neural network to accommodate vast vocabulary for prediction, and making a user-friendly interface system to use the ML translator model.

With the aid of the python libraries OpenCV and MediaPipe, video frames from the user camera on the PC, which recognise the positions of the hand, palm, torso, and face over physical establishing points of interest of the individual, and 21 points of each palm that are the coordinates of pixels that allow it to more accurately predict the sign, as shown in figure 3.2, SLTA can record the hand gestures. [18]

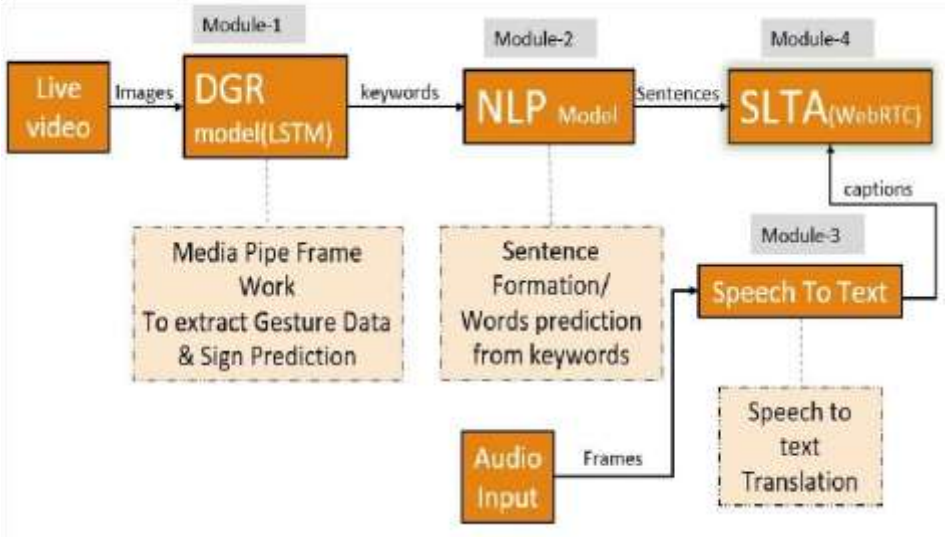

Fig. 3.1. System Architecture of Proposed System SLTA

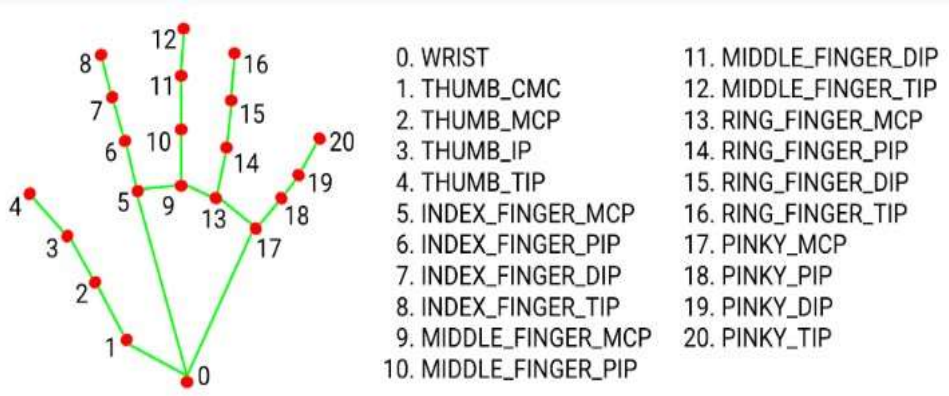

Fig. 3.2. Hand Landmarks detected by Media-Pipe

ASL uses multiple finger combinations to signal each word in its lexicon, often including the locations of the face, hands, and body. The postures of the face, body, and torso matter in a gesture because they help to generate a distinctive lexicon for the activities of the broad sign gesture. The 33 points in the posture holistic landmarks (coordinates) of the entire body may also be detected by MediaPipe Framework, as illustrated in figure 3.3. [18] The majority of ASL language does not refer to the legs or the hip. To record the dataset, just the first 22 points shown in figure 3.3 are used.

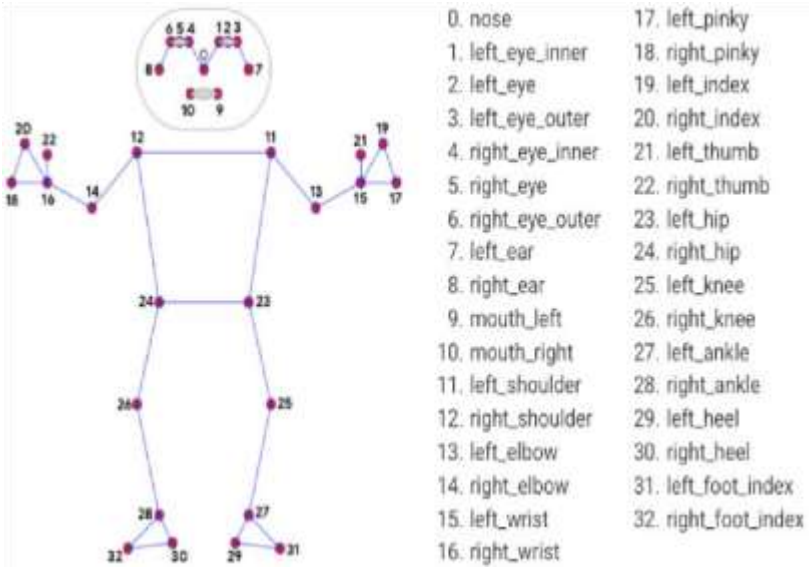

Fig. 3.3. Pose Full Body Landmarks detected By Media-Pipe,

Subject-verb-object (SVO) structure is used in ASL, therefore there are no gaps between sentences. The cognitive perception of humans helps phrases intended to convey to be comprehended by utilising the topic of conversation and verbs connected to the issue. The Deep Gesture Recognition model (DGR), created using a neural network made up of LSTM units, is a module of SLTA that can translate sign language movements into English. The Convolution Neural Network (CNN), which can only anticipate static sign motions, has been demonstrated to have shortcomings in earlier research [20]. Another study on motion gesture identification made use of wearables [8][11] fitted with sensors that gathered acceptable performance spatial 3D data of the hand motion of gestures using LSTM neural networks. However, finding affordable gloves that need less upkeep is difficult. The primary video source for the DGR model is a digital front camera, which is typically found in mobile devices. The OpenCV module, which assists in managing the digital camera data and manipulating it in accordance with the demand, pipes data from a digital camera to the SLTA. The MediaPipe Framework [18] performs gesture identification utilising ML and DL models for the detection of gesture holistic data of the person at each frame of the gesture video, and LSTM neural network is trained using ASL to predict the motion gesture.

The DGR model can recognise sign language and forecast vocabulary in the form of words like "EAT," "DRINK," "HELP," "HELLO," and "THANK YOU," but it is unable to forecast the tenses of these words, such as "EATING," "DRANK," and "HELPED," as well as English articles and prepositions. The Natural Language Processing (NLP) model is able to analyse, forecast, and rebuild words with the meaning that the user intended to convey. In the SLTA application, sentences that are predicted and rebuilt by the NLP model are shown as closed captions (CC). Hearing-impaired persons must learn to grasp the words and phrases of a language they learnt through sign language as communication advances. The speech-to-text paradigm, which translates voice into live captions of the speaker and makes reading the context easier for the hearing impaired, can help the hearing impaired understand the context

of the commoner on the other side of the SLTA. With the help of software engineering approaches and Web Real-Time Communication (WebRTC) technology, it would be feasible to create a user-friendly interface and features for next work.

IV. RESULTS AND DISCUSSIONS

The tests and analyses that were done on the suggested system are briefly discussed in this section. For each move that the system recorded as one data point in the dataset, the video frames must be processed into sequence data of 30 frames in a video. The gesture data is represented as positional coordinates, which MediaPipe uses to represent the gesture holistic of one frame, for thirty frames of each gesture data point. The MediaPipe Framework recognizes the series of frames as a motion gesture since it causes skeletal structures to move. These facts are ASL sign motions that have been converted to words. In order to interpret the 30 frames of sequence gesture from live stream video data and identify motion, LSTM architecture is created based on vocabulary size.

A. Conversion of video frames to sequence data

As seen in figure 4.1, the streaming data is pipelined using OpenCV to the application where MediaPipe Framework's gesture skeletal structure data is extracted from video frames. One point in the ASL dataset of motion gestures is analyzed by MediaPipe [18] to utilize ML and DL models to recognize the skeletal structure in real-time and return collection arrays of each frame as output.

Fig. 4.1. Live Holistic Landmarks detected by MediaPipe

The OpenCV and MediaPipe framework is used to create the ASL dataset for training ML models. Each ASL gesture used to represent a word in the vocabulary was made in front of a camera before being piped into a data-collecting module that turned the video frames into sequence data. Each frame of a 30-frame motion gesture's translated sequence data is kept in a folder, which is used as one data point for the word. 'Auto-py-to-exe' is a library that is used to create an independent executable file (.exe) based on the Python programming language. It doesn't require any Python libraries or frameworks to be installed in order to operate on any Windows PC. By making this system open-source, users may utilise data collector executable files to train their sign motions to function in accordance with the ASL, enabling a broad vocabulary. To create a trustworthy SLR [17] system, experiments with the LSTM architecture and neural network settings are conducted using the ASL dataset.

B. Data Pre-processing

The MediaPipe model will produce data in the form of an object variable. The MediaPipe objects' x, y, z, and visibility variables are used to extract the stance, face, left hand, and right hand (PFLR)landmarks. Only posture landmarks are considered for the visibility variable. Variables from the PFLR landmarks that were extracted are flattened in each category and placed as follows in a two-dimensional (2D) array: stance, face, left and right hands, and feet. So, each frame of a single motion gesture data point is made up of a PFLR sequence of a 1D array, with 30 frames giving us 30 arrays saved in a single folder on the local storage.

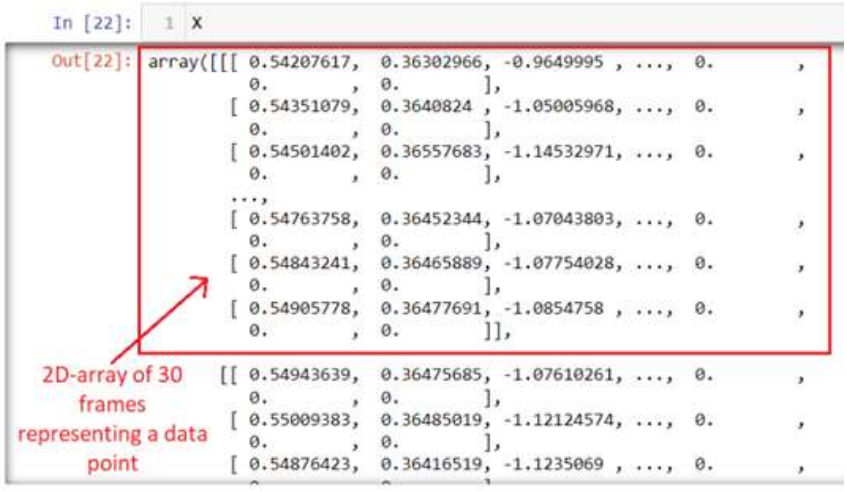

Fig. 4.2. Array representation training dataset

As seen in figure 4.2, the file manager is used to retrieve and aggregate ASL datasets saved in folders into a 2D array to create a data structure that will be fed to the DGR model during training. The training labels for neural network training are created based on the label of the top folder.

C. Building the LSTM architectural Neural Network

ASL dataset used to train the DGR model comprises six words, "HELLO," "EAT," "THANK YOU," "EXCUSE ME," "HELP," and "PLEASE" along with the word "NONE," also added to the vocabulary. The phrase "NONE" denotes the absence of a sign gesture by the user. The MediaPipe Framework, OpenCV, and other libraries were used to create the dataset. In order to extract the landmarks from the gesture data, MediaPipe employs built-in libraries.

Six layers make up the DGR model architecture; the first three are LSTM layers and the final three are completely linked. Thirty frames of gesture data are entered into the DGR model, which outputs seven predicted words, among them "None." Figure 4.3 displays the DGR model's intricate structure.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 30, 64)            442112

lstm_1 (LSTM)                (None, 30, 128)           98816

lstm_2 (LSTM)                (None, 64)                49408

dense (Dense)                (None, 64)                4160

dense_1 (Dense)              (None, 32)                2080

dense_2 (Dense)              (None, 7)                 231

=================================================================
Total params: 596,807
Trainable params: 596,807
Non-trainable params: 0
_________________________________________________________________
```

Fig. 4.3. DGR Model Architecture with LSTM layers

The 360 data points of the dataset provided for training the LSTM neural network, which consists of the four words stated above, form the basis of the training and testing results of the LSTM architecture that are shown. Figure 4.4 illustrates the testing and experimentation that went into the development of the model, which has an accuracy of 98.71 %.

```
optimizer=Adam(decay=1e-4)
# optimizer=SGD(momentum=0.1)
early_stopping = EarlyStopping(monitor='val_loss', mode='min', patience=20, restore_best_weights=True)#monitor=val_acc
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['acc'])
```

```
checkpoints=ModelCheckpoint('Logs\\checkpoints\\' + 'ep{epoch:03d}-loss{loss:.3f}-val_loss{val_loss:.3f}.h5',
                            monitor='val_loss', save_weights_only=True, save_best_only=True)#, period=3)
```

```
model.fit(X, y, batch_size=30, epochs=1000, validation_data=(X_test, y_test), shuffle=True,callbacks=[early_stopping,checkpo
Epoch 22/1000
42/42 [==============================] - 4s 87ms/step - loss: 0.0575 - acc: 0.9841 - val_loss: 0.0548 - val_acc: 0.9841
Epoch 23/1000
42/42 [==============================] - 4s 85ms/step - loss: 0.0854 - acc: 0.9706 - val_loss: 0.0827 - val_acc: 0.9683
Epoch 24/1000
42/42 [==============================] - 4s 85ms/step - loss: 0.0791 - acc: 0.9722 - val_loss: 0.0798 - val_acc: 0.9603
Epoch 25/1000
42/42 [==============================] - 4s 85ms/step - loss: 0.0810 - acc: 0.9770 - val_loss: 0.0760 - val_acc: 0.9802
Epoch 26/1000
42/42 [==============================] - 4s 86ms/step - loss: 0.0673 - acc: 0.9738 - val_loss: 0.0554 - val_acc: 0.9881
Epoch 27/1000
42/42 [==============================] - 4s 87ms/step - loss: 0.0612 - acc: 0.9794 - val_loss: 0.0930 - val_acc: 0.9722
Epoch 28/1000
42/42 [==============================] - 4s 87ms/step - loss: 0.0644 - acc: 0.9825 - val_loss: 0.0448 - val_acc: 0.9921
Epoch 29/1000
42/42 [==============================] - 4s 92ms/step - loss: 0.0526 - acc: 0.9817 - val_loss: 0.0352 - val_acc: 0.9921
Epoch 30/1000
42/42 [==============================] - 4s 95ms/step - loss: 0.0433 - acc: 0.9889 - val_loss: 0.0381 - val_acc: 0.9921

<keras.callbacks.History at 0x17518829eb0>
```

Fig. 4.4. Training results of the DGR Model

D. Sign Gesture Prediction and results

Real-time video stream data from a digital camera is pipelined with OpenCV to the application, where MediaPipe Framework extracts the gesture data. Every video frame's posture, face, left hand, and right-hand coordinates are recorded and saved as sequence data. The DGR model receives this gesture data constantly and uses 30 frames of data to predict each motion. Each frame sequence contains 132 stance landmarks, 1404 face landmarks, 63 left-hand landmarks, and 63 right-hand landmarks, totaling 1662 sequence data points. Figure 4.5 displays the outcomes of the DGR model with a confusion matrix, accuracy, and loss.

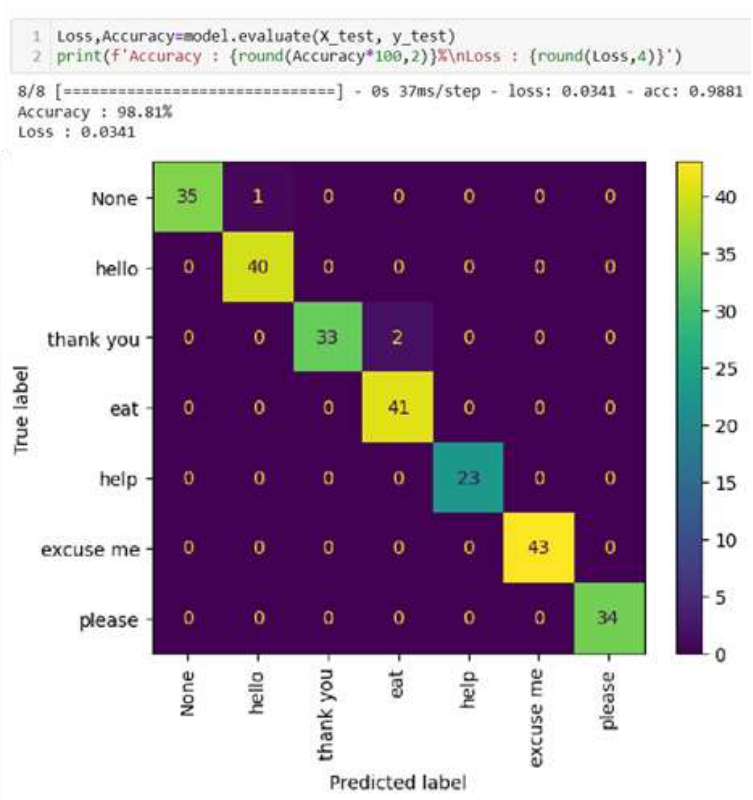

Fig. 4.5. Confusion Matrix, Accuracy and Loss of the DGR Model

After pre-processing and converting the live video data to a data structure in the aforementioned order, the DGR model with LSTM architecture predicts the gesture using the given data. Each gesture is predicted based on the likelihood that a certain gesture will be executed at every 30-frame sequence. The term is predicted using the vocabulary word with the highest likelihood, and that word exceeds the 0.9 threshold probability. According to figure 4.6, the display output receives each predicted word.

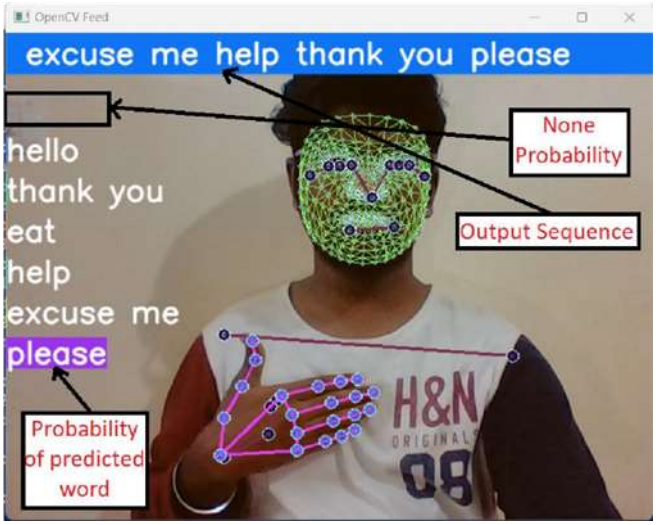

Fig. 4.6. Prediction of ASL with the DGR model

V. CONCLUSION

The major form of communication for those who are deaf or have trouble speaking is sign language. Sign language, particularly American sign language, is not well known. It might be challenging for regular people to comprehend handicapped individuals. Through the use of sign language recognition models, the Sign Language Translation Application (SLTA) seeks to eliminate the communication gap between people with disabilities and able-bodied people. SLTA uses a digital camera to record signed motions and the MediaPipe Framework to retrieve the gesture data. The dynamic motion gestures made by the person in front of the camera may be predicted in real-time with the aid of the DGR model created using LSTM architecture. Using 252 test samples and seven vocabulary terms, an accuracy of 98.81% was attained. By anticipating motion gestures and displacing the use of sensor-based gloves and wearables for gesture motion data gathering, the proposed system also resolves the issues with static gesture recognition utilising LSTM architecture.

This programme may be used to implement video conferencing for those who are deaf or hard of hearing using WebRTC protocols. This system may still be improved and developed further by adding new features like increasing the ASL vocabulary's size, creating an NLP model to anticipate words into sentences, and implementing the WebRTC protocol for video conferencing.

REFERENCES

[1] J. Hou, X.-Y. Li, P. Zhu, Z. Wang, Y. Wang, J. Qian, and P. Yang, "Signspeaker: A real-time, high-precision smartwatch-based sign language translator," in Proc. of ACM MobiCom, 2019.

[2] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," arXiv preprint arXiv:1801.10111, 2018.

[3] J. Wu, Z. Tian, L. Sun, L. Estevez, and R. Jafari, "Real-time American sign language recognition using wrist-worn motion and surface EMG sensors," in Proc. of IEEE BSN, 2015

[4] J. Zang, L. Wang, Z. Liu, Q. Zhang, G. Hua, and N. Zheng, "Attention-based temporal weighted convolutional neural network for action recognition," in Proc. of IFIP INTERACT, 2018, pp. 97–108.

[5] J. Zhang, J. Tao, and Z. Shi, "Doppler-radar based hand gesture recognition system using convolutional neural networks," in International Conference in Communications, Signal Processing, and Systems. Springer, 2017, pp. 1096–1113.

[6] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in Proc. of ACM CHI, 2016, pp. 1515–1525.

[7] Julian Menezes .R, Albert Mayan .J, M. Breezely George," Development of a Functionality Testing Tool for Windows Phones", Indian Journal of Science and Technology,Vol:8,Issue:22,pp: 1-7,September 2015.

[8] T. T. Swee, A. Ariff, S.-H. Salleh, S. K. Seng, and L. S. Huat, "Wireless data gloves malay sign language recognition system," in Information, Communications & Signal Processing, 2007 6th International Conference on. IEEE, 2007, pp. 1–4.

[9] T . Zhao, J. Liu, Y. Wang, H. Liu, and Y. Chen, "Ppg-based finger level gesture recognition leveraging wearables," in Proc. of IEEE INFOCOM, 2018, pp. 1457–1465.

[10] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, vol. 41, no. 6, pp. 1064–1076, 2011.

[11] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices," in IEEE Transactions on Human-Machine Systems, vol. 44, no. 2, pp. 293-299, April 2014, doi: 10.1109/THMS.2014.2302794.

[12] Bhokse, B. (January 1, 2015). ISSN 2348 – 7968 hand gesture recognition using a neural network - IJISET. IJISET - International Journal of Innovative Science, Engineering & Technology. Retrieved October24, 2022,from https://www.ijiset.com/vol2/v2s1/IJISET_V2_I1_01.pdf

[13] Coster(UGent), M. D., & Dambre(UGent), and J. (1970, January 1). Sign language recognition with Transformer Networks. Sign language recognition with transformer networks. Retrieved October 24, 2022, from http://hdl.handle.net/1854/LU-8660743

[14] K. Kudrinko, E. Flavin, X. Zhu, and Q. Li, "Wearable SensorBased Sign Language Recognition: A Comprehensive Review," in IEEE Reviews in Biomedical Engineering, vol. 14, pp. 82-97, 2021, doi: 10.1109/RBME.2020.3019769.

[15] Asha Pandian, Bharathi B, Albert Mayan J, Prem Jacob, Pravin "A Comprehensive View of Scheduling Algorithms for MapReduce Framework in Hadoop," Journal of Computational and Theoretical Nanoscience, Vol.16, No. 8, pp. 3582-3586, 2019

[16] Mitra, S. (2007, May 3). GESTURE RECOGNITION: A survey. IEEE Xplore. Retrieved October 24, 2022, from https://ieeexplore.ieee.org/document/4154947

[17] Razieh Rastgoo, Kourosh Kiani, Sergio Escalera, Sign Language Recognition: A Deep Survey, Expert Systems with Applications, Volume 164, 2021, 113794, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113794. (https://www.sciencedirect.com/science/article/pii/S095741742030614X)

[18] Lugaresi, Camillo. "MediaPipe: A Framework for Building Perception Pipelines." arXiv.org, June 14, 2019. https://arxiv.org/abs/1906.08172.

[19] Wang, Z., Zhao, T., Ma, J., Chen, H., Liu, K., Shao, H., Wang, Q., & Ren, J. (2020). Hear sign language: A real-time end-toend sign language recognition system. IEEE Transactions on Mobile Computing, 1–1. https://doi.org/10.1109/tmc.2020.3038303

[20] Wikimedia Foundation. (2022, October 23). American sign language. Wikipedia. Retrieved October 24, 2022, from https://en.wikipedia.org/wiki/American_Sign_Language

[21] Y. Suresh, J. Vaishnavi, M. Vindhya, M. S. A. Meeran and S. Vemala, "MUDRAKSHARA - A Voice for Deaf/Dumb People," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-8, doi: 10.1109/ICCCNT49239.2020.9225656.

[22] Bazarevsky, V., & Zhang, F. (2019, August 19). Hands. OnDevice, Real-Time Hand Tracking with MediaPipe. Retrieved February 10, 2023, from https://google.github.io/mediapipe/solutions/hands.html

[23] Bazarevsky, V., & Grishchenko, I. (2020, August 13). Pose. On-device, Real-time Body Pose Tracking with MediaPipe BlazePose. Retrieved February 10, 2023, from https://google.github.io/mediapipe/solutions/pose.html

[24] Teak-Wei, C., & Boon Giin, L. (2018, October). The 26 letters and 10 digits of American Sign Language (ASL). American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. Retrieved February 10, 2023, from https://www.researchgate.net/figure/The-26-letters-and-10digits-of-American-Sign-Language-ASL_fig1_328396430 .