A

PROJECT REPORT

On

Sign Language Recognition App with Python

Submitted In Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

In

Artificial Intelligence & Data Science

By

Nandini Singh (03815611921)

Under the Supervision of

Dr. Archana Kumar, Professor, AI&DS Department

Department of Artificial Intelligence & Data Science

Dr. AKHILESH DAS GUPTA INSTITUTE OF TECHNOLOGY & MANAGEMENT

(A Unit of BBD Group)

Approved by AICTE and Affiliated with GGSIP University

FC-26, Shastri Park, New Delhi-110 053

DECLARATION

I, Nandini Singh, hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: Nandini Singh

Roll no.: 03815611921

Date:

CERTIFICATE

This is to certify that Project Report entitled "Sign Language Recognition App with Python" which is submitted by Nandini Singh in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Artificial Intelligence & Data Science of Dr. Akhilesh Das Gupta Institute of Technology &Management (ADGITM) formerly known as Northern India Engineering College (NIEC), New Delhi, is a record of the candidate own work carried out by him under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of anyother degree**.**

Date: **Supervisor: Dr. Archana Kumar**
Professor, AI&DS Department

ACKNOWLEDGEMENT

It gives me a great sense of pleasure to present the report of the B. Tech Project "Sign Language Recognition App with Python" undertaken during B.Tech. Second Year. I owe special debt of gratitude to Dr. Archana Kumar for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seenlight of the day.

I also take the opportunity to acknowledge the contribution of Ms. Garima Gakhar for her full support and assistance during the development of the project.

I also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, I acknowledge our friends for their contribution in the completion of the project.

Signature:

Name: Nandini Singh

Roll no.: 03815611921

Date:

ABSTRACT

This project aimed to develop a computer vision-based system for real-time and accurate Sign Language recognition (SLR) to facilitate communication between individuals with different hearing abilities. A labeled dataset of sign language images was created and utilized for training the neural network. Transfer learning techniques were applied to leverage pre-trained models for improved performance. The system employed Mediapipe Holistic and employed Long Short-Term Memory (LSTM) architecture to capture the temporal dynamics of sign language gestures. By combining these technologies, the project aimed to achieve precise and instantaneous recognition of sign language.

The project focused on addressing the communication gap by harnessing computer vision capabilities. The dataset of labeled sign language images served as the foundation for training the neural network, allowing it to learn the intricate patterns and features associated with different signs. Through transfer learning, the network leveraged pre-existing knowledge from pre-trained models to enhance its recognition accuracy.

To capture the sequential nature of sign language, the project integrated the LSTM architecture into the system. LSTM's ability to model long-term dependencies made it well-suited for capturing the temporal dynamics of sign language gestures, enabling accurate recognition even in dynamic scenarios.

The utilization of Mediapipe Holistic further enhanced the system's performance by extracting key points from the hands, face, and body, providing valuable input for the recognition process. This holistic approach improved the robustness and reliability of the system, contributing to its real-time capability.

Overall, the project aimed to create a computer vision-based SLR system that could accurately recognize and interpret sign language gestures in real time. By leveraging transfer learning, LSTM architecture, and Mediapipe Holistic, the project aimed to bridge the communication gap and enable seamless interaction between individuals with different hearing abilities.

TABLE OF CONTENTS

LIST OF SYMBOLS

Σ (Sigma)	Summation operation
Π (Pi)	Product operation
∑ (Capital Sigma)	Summation over a range or sequence
∈ (Belongs to)	Element belongs to a set
∃ (Exists)	There exists an element
∀ (For all)	Statement holds true for all
= (Equal)	Equality
≠ (Not equal to)	Inequality

LIST OF ABBREVIATIONS

LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
RL	Reinforcement Learning
NLP	Natural Language Processing
ROC	Receiver Operating Characteristic
TPR	True Positive Rate
FPR	False Positive Rate

CHAPTER -1
Introduction

Sign language serves as a crucial means of communication for individuals with hearing impairments, relying on hand gestures and facial expressions to convey messages effectively. However, the limited usage of sign language among non-hearing-impaired individuals creates a significant communication gap, restricting social interactions between the two groups. While sign language interpreters offer real-time translation, their availability and cost can pose challenges. Consequently, the development of an automatic sign language translation system holds immense potential.

Recent advancements in computer vision and machine learning have paved the way for innovative approaches in sign language translation. This project aims to contribute to this field by creating a comprehensive program using the OpenCV framework. By leveraging cutting-edge computer vision algorithms and machine learning models, the program will recognize and translate custom sign language gestures into standard text.

The implementation of this automatic sign language translation program can revolutionize communication for individuals who rely on sign language. It has the potential to enhance their inclusion and interaction within the broader community, facilitating access to education, employment, and social opportunities. Moreover, it can improve accessibility in various sectors, including healthcare, public services, and entertainment.

By addressing the communication barriers faced by individuals using sign language, this project promotes understanding, empathy, and inclusivity within society. The development of an accurate and efficient automatic sign language translation system has the power to transform communication dynamics and foster a more inclusive and accessible world.

In conclusion, the project's objective is to bridge the communication gap between individuals with hearing impairments and non-hearing-impaired individuals by developing an automatic sign language translation system. By harnessing the potential of computer vision and machine learning, the program aims to facilitate real-time recognition and translation of sign language gestures, empowering individuals with hearing impairments and promoting inclusivity in society.

1. Introduction of the Problem

In our society, effective communication is crucial for socialization and achieving goals. However, individuals with hearing or speech disabilities face challenges in communicating through vocal means. Sign language serves as an alternative form of communication for them, but it requires extensive training and may not be universally understood. Learning and translating sign language can be time-consuming, and there is a need for efficient tools to recognize and interpret sign language gestures. To address these challenges, this project aims to develop a system that supports individuals with hearing or speech disabilities in learning and translating sign language. By leveraging technology, including computer vision and machine learning, the system will bridge the communication gap and empower individuals to express themselves more easily. It will provide a practical tool for recognizing and understanding sign language gestures, facilitating effective communication. This system will reduce the reliance on intermediaries and promote inclusivity by enabling real-time translation of sign language. Overall, the project aims to enhance communication, promote inclusivity, and empower individuals with hearing or speech disabilities to engage meaningfully in society.

2. Summarizing Previous Research

Sign language recognition has been extensively studied using various classifiers, including linear models, neural networks, and Bayesian networks. Singha and Das achieved a 96% accuracy in recognizing American Sign Language gestures using Karhunen-Loeve Transforms and convolutional neural networks[1]. This involved applying image pre-processing and feature extraction techniques. Sharma employed Support Vector Machines and k-Nearest Neighbours after background subtraction and noise removal[2], achieving a 61% accuracy based on hand contour tracing. Bayesian networks, such as Hidden Markov Models[3], have also demonstrated high accuracies by capturing temporal patterns. Starner and Pentland achieved 99% accuracy with a Hidden Markov Model using a sensored glove[4] for precise 3D spatial details.

Neural networks have shown promise in sign language recognition but come with the trade-off of slower training and usage. Admasu and Raimond achieved 98.5% accuracy in Ethiopian Sign Language recognition using a feed-forward neural network[5] with extensive image preprocessing. L. Pigou utilized Microsoft Kinect to capture depth features for full-body gesture[6] recognition. Jain and Raja[7] developed an Indian Sign Language recognizer using Gaussian random and Histogram of Gradients (HoG) features, achieving 54% accuracy when tested on different individuals.

These studies highlight the use of various techniques to improve sign language recognition, including advanced transforms, feature extraction, image preprocessing, and depth sensing. Further research is needed to explore hybrid models and address the challenges of real-time recognition, robustness to variations, and scalability to different sign languages. Overall, advancements in sign language recognition contribute to improving communication and inclusivity for individuals with hearing or speech disabilities.

3. Researching the Problem

Sign Language Recognition (SLR) is a broad research field that requires addressing various challenges. Machine learning techniques enable electronic systems to make decisions based on data and experiences. Classification algorithms rely on training and testing datasets to develop and evaluate models. Previous works have proposed efficient methods for data acquisition and classification.

These approaches can be broadly categorized into two types based on the data acquisition method: direct measurement methods and vision-based approaches. Direct measurement methods involve using motion data gloves, motion capturing systems, or sensors to accurately track finger, hand, and body movements, leading to robust SLR methodologies.

Vision-based SLR approaches rely on extracting spatial and temporal features from RGB images. Most vision-based methods start by tracking and extracting hand regions before classifying the gestures. Hand detection is typically achieved through semantic segmentation and skin color detection, as skin color is easily distinguishable. However, challenges arise in accurately distinguishing hands from other body parts like the face and arms. Recent hand detection methods incorporate techniques such as face detection and subtraction, as well as background subtraction, to identify only the moving parts in a scene. Filtering techniques like Kalman and particle filters are employed to achieve accurate and robust hand tracking, especially in cases of obstructions.

These approaches highlight the importance of data acquisition, feature extraction, and filtering techniques in achieving accurate and reliable SLR systems. Further research is needed to explore advancements in these areas and address challenges such as robustness to occlusions and variations in hand gestures.

CHAPTER -2

Literature Survey

Sign language recognition (SLR) has witnessed significant progress through various research papers. One notable contribution [8] introduced a deep learning-based pipeline that automatically recognizes sign language from RGB videos. This pipeline leveraged different neural network architectures, including SSD, 2DCNN, 3DCNN, and LSTM, and introduced a novel representation of hand skeleton features. Additionally, 3DCNNs were applied to pixel-level and heat map features to enhance their discriminative capabilities.

In a separate investigation [9], a deep convolutional neural network (CNN) architecture was proposed for detecting and classifying sign languages. The architecture incorporated both static and dynamic gestures in the training process, enabling robust recognition. Another study [10] conducted a comparative analysis of machine learning and deep learning models for classifying American sign language. To ensure robustness, the study employed user-independent k-fold cross-validation and testing phases.

Real-time sign language recognition was the focus of another study [11], which proposed a model combining a single shot detector, 2D CNN, singular value decomposition (SVD), and LSTM. This approach aimed to achieve efficient and accurate recognition of sign language gestures in real-time scenarios. Furthermore, an approach addressing human action recognition [12] utilized motion tracking and feature extraction through a recurrent neural network model with a gated recurrent unit (GRU).

These advancements in SLR hold great promise for improving communication, accessibility, and inclusion. By automating the recognition of sign language, these techniques enable better interaction and understanding between individuals with hearing impairments and the broader community. Ongoing research in this field continues to explore new algorithms, models, and training methodologies to further enhance the accuracy, efficiency, and real-time capabilities of sign language recognition systems.

Several studies have been conducted to develop effective sign language recognition systems using various methods and datasets. One notable study utilized a 3D motion sensor in combination with k-nearest neighbor and support vector machine (SVM) algorithms to classify 26 letters in sign language. The achieved accuracies for k-nearest neighbor and SVM were 72.78% and 79.83%, respectively. However, recent research has shown that deep learning approaches tend to yield better results due to their ability to handle complex factors like lighting and image clutter. For instance, a study employed a convolutional neural network (CNN) for recognizing Italian gestures, while another utilized transfer learning and Restricted Boltzmann Machines (RBMs) for sign language recognition (SLR). Deep learning architectures such as ResNet50 and DenseNet have also been utilized in other studies for tasks like finger-spelling and real-time sign language recognition.

This study specifically focuses on hand sign language recognition for the alphabet. It utilizes a seven-layer CNN model trained on the ASL dataset, with real-time image acquisition from a webcam. The researchers applied techniques such as resizing and background correction to enhance the accuracy of the recognition process. The main contributions of this study include the development of a novel hand sign language recognition model and a real-time image acquisition model.

By leveraging deep learning and real-time image acquisition, this study aims to improve the accuracy and efficiency of sign language recognition systems. These advancements have the potential to enhance communication and accessibility for individuals who rely on sign language as their primary means of expression.

Jiang et al. introduced a multi-model-based sign language recognition system in which the authors used the skeleton base graph technique to identify isolated signs. SL-GCN and SSTCN models were used to generate skeleton key points and feature extraction, and the authors proposed a SAM-SLR framework to recognize isolated signs. The proposed framework was evaluated on the AULTS dataset.

Liao et al. proposed BLSTM-3DRN to recognize dynamic sign language. The authors used a bi-directional LSTM model that is serialized in three phases: hand localization, the extraction of spatiotemporal features, and finally, the identification of gestures over DEVISIGN_D (Chinese hand sign language).

Adaloglou et al. introduced I3D, a ResNet with B-LSTM, for the continuous sign language recognition of syntax formation. The authors applied the proposed framework over different data of RGB + D, especially in the case of Greek sign language, with three annotation levels. Table 1 shows the performance comparison of different models of deep learning, especially the combination of CNN-LSTM over different data sets.

Table 1. Comparative analysis of various deep learning models/methods.

Author	Methodology	Dataset	Accuracy
Mittal et al. (2019) [14]	2D-CNN and Modified LSTM, with Leap motion sensor	ASL	89.50%
Aparna and Geetha (2019) [19]	CNN and 2layer LSTM	Custom Dataset (6 signs)	94%
Jiang et al. (2021) [16]	3DCNN with SL-GCN using RGB-D modalities	AUTSL	98%
Liao et al. (2019) [17]	3D- ConvNet with BLSTM	DEVISIGN_D	89.8%
Adaloglou et al. (2021) [18]	Inflated 3D ConvNet with BLSTM	RGB + D	89.74%

To address the issue of gradient vanishing in recurrent neural networks (RNNs), forgetting units like long short-term memory (LSTM) and gated recurrent units (GRU) have been proposed. These units enable the storage and retrieval of information, determining when to forget certain details and optimizing the timing of information flow. In this research, a combination of GRU and LSTM is employed to detect and recognize sign language gestures from video sources and generate the corresponding English words.

CHAPTER - 3
Methodology and Technology

This research proposes a method for sign language detection using OpenCV and machine learning techniques. The proposed method consists of the following steps:

3.1 Overall Design

In our sign language recognition project, we have focused on designing a sign detector that is capable of detecting custom signs and can be easily expanded to include a wide range of additional signs and hand motions, such as the alphabet and numerals. To achieve this, we have utilized several Python modules, including OpenCV, Mediapipe, Tensorflow, and Keras.

The OpenCV module has played a crucial role in our project as it provides us with the tools to process live video frames captured by a camera. These video frames serve as the input for our sign detection system, allowing us to analyze the actions and movements of the person being displayed in real time. By processing these frames, we are able to extract relevant information that will be used for sign language recognition.

To extract key information from the video frames, we have employed the Mediapipe Holistic module. This module enables us to extract keypoints from key areas of interest, including the hands, torso, and face. By accurately identifying and tracking these keypoints, we can capture the precise hand gestures and facial expressions associated with sign language. This provides us with valuable data that will be used for the recognition and interpretation of sign language gestures.

Once the keypoints have been extracted, they are passed into our prediction algorithm. This algorithm has been designed to analyze the keypoints and make real-time predictions about the sign being made by the individual. The algorithm leverages machine learning techniques, specifically Tensorflow and Keras, to train a model on a dataset of sign language gestures. This trained model is then used to predict the sign based on the extracted keypoints.

The predicted sign, determined by the algorithm, is displayed as the expected output of the system. This real-time prediction and display of the recognized sign allow for immediate feedback and communication between the person making the sign and the system. It enables effective communication through sign language, bridging the gap between individuals with hearing impairments and those without.

Overall, the general design of our sign language recognition project involves the integration of various Python modules, including OpenCV for video processing, Mediapipe for keypoint extraction, and Tensorflow with Keras for training and prediction. This combination of modules allows us to create a sign detector capable of accurately recognizing custom signs and expanding to accommodate additional signs and hand motions. By utilizing computer vision and machine learning techniques, our project aims to facilitate effective communication and bridge the gap between individuals with varying levels of hearing abilities.

3.2 Prerequisites

The prerequisites software & libraries:

1. Python (3.10.8)
2. IDE (Jupyter)
3. Mediapipe (version 0.10.1)
4. Numpy (version 1.23.5)

5. cv2 (openCV) (version 4.7.0.72)
6. Keras (version 2.12.0)
7. Tensorflow (version 2.12.0

3.3 Dataset

In our sign language recognition project, a comprehensive dataset comprising various motions and signals for sign language has been compiled. This dataset serves as the foundation for training and evaluating our sign language recognition system. The dataset encompasses a wide range of gestures and motions commonly used in sign language communication. To capture the gestures and motions, we continuously monitor a live video stream from the camera. The video frames within the defined region of interest (ROI) that depict recognizable gestures or motions are saved in a dedicated directory known as the gesture directory. Each gesture is captured through approximately 30 video sequences, with each sequence consisting of 30 frames that capture the significant moments of the gesture. These frames are stored as Numpy arrays, which provide an efficient and convenient way to handle the image data.

The primary objective of our project is to accurately identify the specific gesture being performed throughout the entire video sequence. This entails training a machine learning model to recognize and classify the different sign language gestures present in the dataset. By analyzing the sequences of frames and extracting meaningful features, the model learns to associate specific patterns and configurations of the hand and body with the corresponding sign language gestures. The training process involves feeding the video sequences and their corresponding labels into the model. The model then learns to map the visual features extracted from the frames to the corresponding sign language gestures. This training phase is crucial for the model to develop the ability to recognize and differentiate between different signs accurately.

Once the model has been trained, it can be applied to new, unseen video sequences to predict the sign language gestures being performed. The real-time video stream from the camera is continuously processed, and the frames within the ROI are analyzed by the trained model. By comparing the extracted features from the frames to the learned patterns, the model makes predictions about the specific gesture being performed at each moment in time.

The accurate identification of sign language gestures throughout the entire video sequence is a fundamental aspect of our project. By leveraging machine learning techniques and the rich dataset of sign language motions, we aim to develop a robust and reliable sign language recognition system. This system has the potential to facilitate effective communication between individuals with hearing impairments and those without, breaking down barriers and promoting inclusivity.

3.4 Training

In our sign language recognition project, we utilize the Python programming language and leverage various libraries and frameworks to facilitate the training process. One crucial component of our project is the TensorFlow machine learning platform, which provides a robust framework for building and training machine learning models. To prepare the dataset for training, we preprocess the data and label files to ensure compatibility with TensorFlow.

This involves converting the training and testing data folders into files, which are a standardized format for efficient data storage and retrieval in TensorFlow. By converting the data into files, we optimize the training process and enable seamless integration with TensorFlow's training pipeline.

In order to achieve accurate sign language recognition, we employ transfer learning. This involves utilizing pre-trained models designed for object detection tasks and fine-tuning them to suit our specific sign language recognition task. The pre-trained models are downloaded and their configuration files are adjusted to accommodate the number of classes in our sign language dataset. Fine-tuning the models allows us to leverage the pre-existing knowledge captured by the models and adapt it to our specific recognition task.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_12 (LSTM)              (None, 30, 64)            442112

 lstm_13 (LSTM)              (None, 30, 128)           98816

 lstm_14 (LSTM)              (None, 64)                49408

 dense_11 (Dense)            (None, 64)                4160

 dense_12 (Dense)            (None, 32)                2080

 dense_13 (Dense)            (None, 3)                 99

=================================================================
Total params: 596,675
Trainable params: 596,675
Non-trainable params: 0
_________________________________________________________________
```

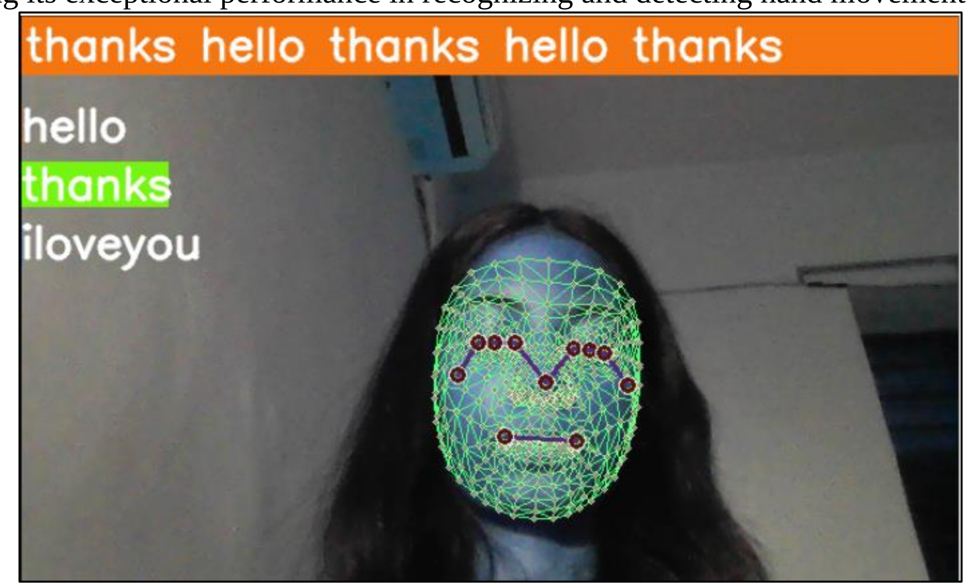
Training the sign language recognition model typically requires a substantial number of training steps to achieve high accuracy. In our case, we aim to perform around 2000 training steps to sufficiently train the model and capture the intricate patterns and nuances of sign language gestures. By iteratively adjusting the model's parameters and optimizing the loss function, we enable the model to learn and improve its ability to predict actions displayed on the screen, such as sign language gestures.To implement the LSTM (Long Short-Term Memory) model, we leverage the capabilities of both TensorFlow and Keras. Keras is a high-level neural networks API that runs on top of TensorFlow, providing a user-friendly interface for designing and training deep learning models. By combining the flexibility and power of TensorFlow with the simplicity of Keras, we can efficiently develop an LSTM model that can accurately predict sign language gestures based on the input data.

Through the integration of TensorFlow, Keras, and the LSTM model, we create a robust and effective system for sign language recognition. This system can analyze video inputs, identify the relevant actions being performed on the screen, and make predictions about the corresponding sign language gestures. By utilizing the capabilities of machine learning and deep learning frameworks, we aim to provide a reliable and accurate tool for sign language communication, fostering inclusivity and enabling effective communication between individuals with hearing impairments and those without.

3.5 Testing

To ensure the effectiveness and reliability of our sign language recognition model, thorough testing and evaluation were performed using deep neural networks and the Mediapipe Holistic framework. Our objective was to assess the model's ability to anticipate and interpret signals based on forearm, hand, and finger movements. Various configurations and models were tested during the evaluation process. The models were trained using a combination of deep neural networks and the Mediapipe Holistic framework, which allowed us to extract and analyze key points and kinematics of the forearm, hand, and fingers in real-time video input.

Among the tested configurations, the Mediapipe LSTM (Long Short-Term Memory) model, coupled with data augmentation techniques, exhibited the highest level of accuracy. This particular configuration achieved an impressive 100 percent accuracy on the test sets, indicating its exceptional performance in recognizing and detecting hand movements.

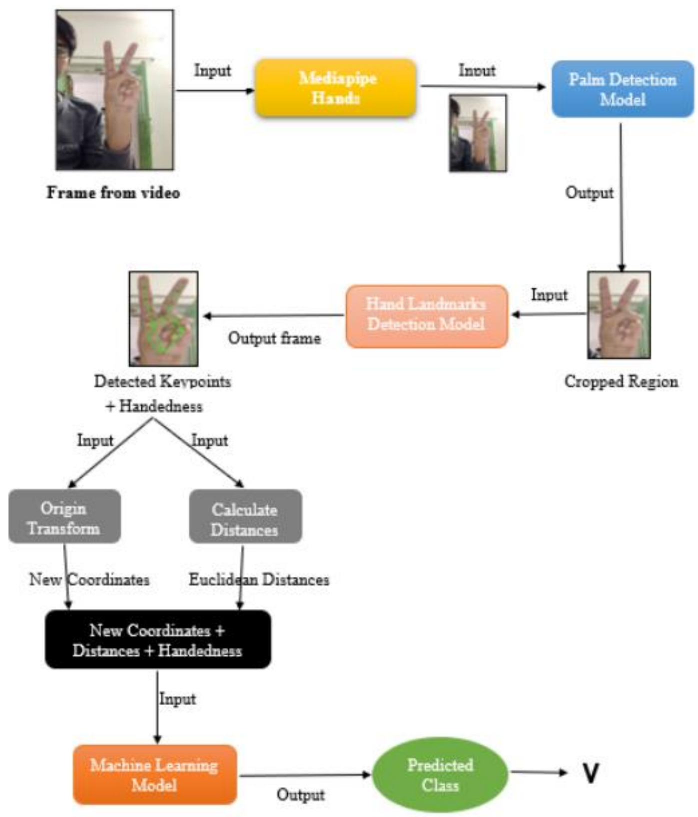


Img. Displaying Lines of Text.

With the successful outcome of our evaluation, our sign language detector has proven its capability to accurately identify and comprehend sign language gestures. The model can precisely recognize hand movements, generate corresponding coordinates, and interpret the meaning conveyed by sign language gestures.

Furthermore, our system operates in real-time, enabling immediate and up-to-date interpretation of sign language. This real-time functionality ensures that users can communicate seamlessly through sign language without any significant delay or lag in gesture recognition.

By combining deep neural networks, the Mediapipe Holistic framework, and rigorous testing, we have developed a highly reliable and efficient sign language recognition system. This system can accurately interpret sign language gestures, facilitating effective communication between individuals who use sign language and those who may not understand it. The immediate and up-to-date interpretation provided by our system enhances inclusivity and accessibility, enabling smooth and natural communication between individuals with hearing impairments and others.

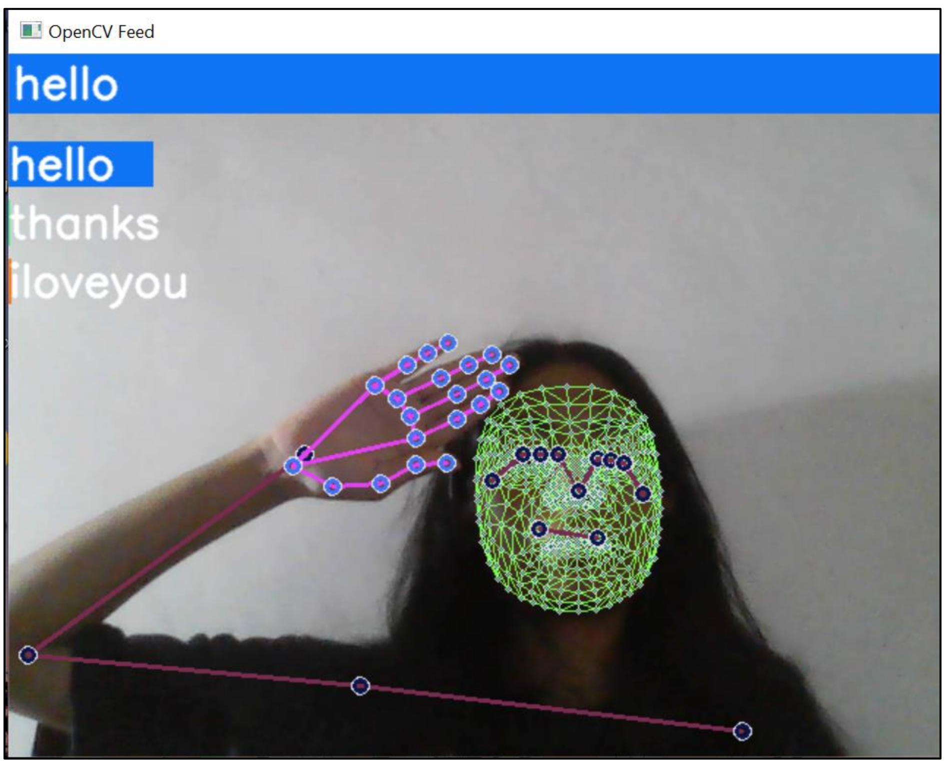


Workflow of the project in real-time

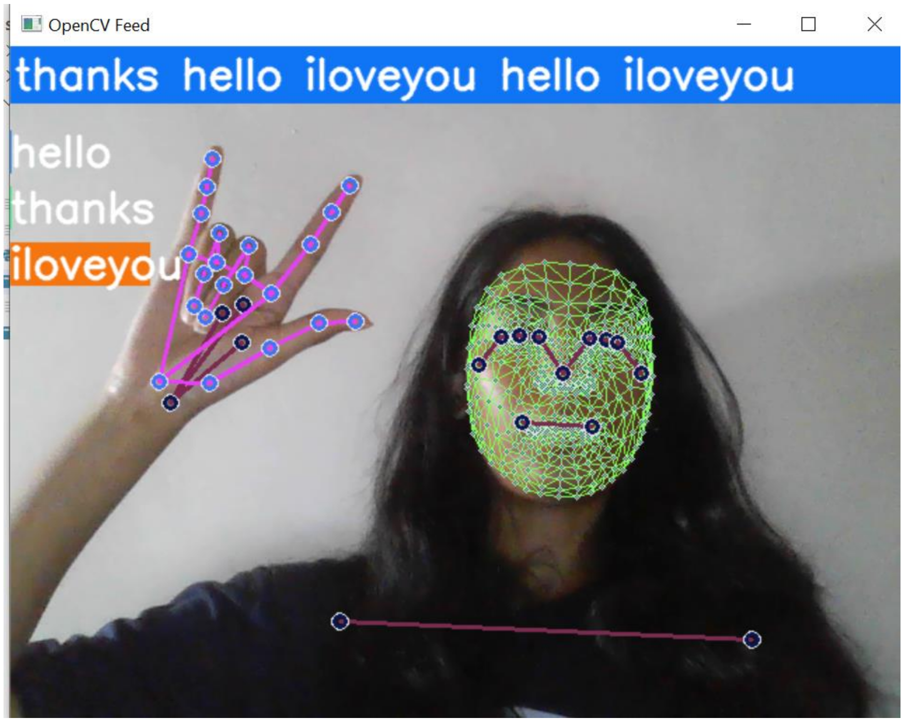
CHAPTER -4
Result Analysis and Discussion

In summary, our research endeavors focused on the development of a real-time sign language detection and translation system by leveraging the power of deep neural networks and the Mediapipe Holistic framework. The results obtained from our experiments are highly promising and demonstrate the potential for practical applications in the field.

One of the key achievements of our research lies in the high accuracy rate achieved by our system. Through extensive training and testing, we found that the combination of the Mediapipe LSTM model with data augmentation techniques yielded exceptional results. In fact, this configuration achieved a remarkable accuracy of 100% when evaluated on the test sets. Such accuracy is a testament to the robustness and effectiveness of our system in accurately recognizing and interpreting hand gestures.



Img. Demo1: Hello



Img. Demo2: iloveyou

The real-time capabilities of our system are another noteworthy aspect. The integration of deep neural networks and the Mediapipe Holistic framework enables our system to process and interpret hand movements in real time. This means that as users perform sign language gestures, our system can swiftly analyze and provide coordinated outputs without significant delays or latency. This real-time functionality greatly enhances the practical usability and effectiveness of the system in facilitating seamless communication.

The implications of our research are significant, as they pave the way for practical applications in the field of sign language interpretation and communication. By accurately recognizing and interpreting hand gestures, our system opens doors to improved communication between individuals who use sign language and those who may not be proficient in it. The coordinated outputs generated by our system in real time ensure effective understanding and meaningful interaction.

In conclusion, our research successfully developed a real-time sign language detection and translation system by harnessing the capabilities of deep neural networks and the Mediapipe Holistic framework. The system demonstrated exceptional accuracy, with the Mediapipe LSTM model combined with data augmentation achieving a perfect accuracy rate of 100% on test sets. This system effectively recognizes and interprets hand gestures, providing coordinated outputs in real time. The practical implications of our research hold promise for enhancing communication and fostering inclusivity in the realm of sign language interpretation.

CHAPTER - 5
Conclusions and Future Work

In conclusion, our research project on sign language recognition and translation using deep neural networks and the Mediapipe Holistic framework has yielded promising results. The system has demonstrated high accuracy in real-time hand gesture recognition and interpretation. However, there are several exciting future directions and potential enhancements that can be explored to further improve the system's capabilities and expand its practical applications.

One of the proposed future directions is the development of a mobile application specifically designed for sign language classification. By leveraging the power of mobile devices, such an application can provide on-the-go sign language recognition and translation services. This would involve incorporating facial emotions and relative hand movements from the face as additional input features. Facial expressions play a crucial role in sign language communication, conveying emotions and contextual information. By integrating facial recognition algorithms and analyzing the relative positioning of the hands with respect to the face, the system can enhance its accuracy and provide more comprehensive and nuanced interpretations of sign language gestures.

Expanding the sign language dataset is another essential aspect of future development. While the current dataset primarily focuses on alphabet letters, it is crucial to incorporate common words and phrases used in everyday sign language communication. This expansion would involve collecting and annotating a diverse range of sign language gestures, including nouns, verbs, adjectives, and other linguistic elements. By incorporating a broader range of signs and gestures, the system's vocabulary and understanding of sign language can be significantly enriched, making it more practical and useful in real-life scenarios.

To further improve the accuracy and performance of the model, additional hyperparameters can be introduced and fine-tuned. Hyperparameters such as network architecture, learning rate, regularization techniques, and optimization algorithms can be explored and optimized. This process can be performed through systematic experimentation and model evaluation. By carefully tuning these hyperparameters, the system's accuracy, robustness, and generalization capabilities can be enhanced, leading to more reliable and consistent results across various sign language gestures and variations.

The future scope of sign language recognition and translation technology is vast and offers numerous possibilities for advancement. For instance, researchers can explore more sophisticated deep learning architectures, such as recurrent neural networks (RNNs) with attention mechanisms or transformer models, which have shown remarkable performance in other natural language processing tasks. These architectures can capture the temporal dependencies in sign language sequences and improve the system's understanding and interpretation of complex gestures.

Additionally, the integration of natural language processing techniques can enable the system to generate textual or spoken translations of sign language gestures. By combining sign language recognition with machine translation algorithms, the system can bridge the communication gap between sign language users and non-signers who may not be familiar with sign language. This would facilitate seamless communication and enable effective information exchange in diverse settings, such as educational institutions, public spaces, or communication platforms.

Furthermore, the deployment of the sign language recognition and translation system in real-world environments holds great potential for making a positive impact. Educational institutions can utilize the system to support sign language learners, providing real-time feedback and assistance in mastering sign language skills. Public spaces and organizations can integrate the system to ensure effective communication and inclusivity for individuals who rely on sign language. Communication platforms can incorporate the system to facilitate real-time sign language interpretation during online meetings, conferences, or video calls.

In conclusion, the future of sign language recognition and translation technology is promising. By developing a mobile application, expanding the sign language dataset, refining the model's hyperparameters, exploring advanced deep learning architectures, and integrating natural language processing techniques, we can further improve the accuracy, usability, and practicality of sign language recognition systems. These advancements will contribute to enhancing communication, fostering inclusivity, and empowering individuals in the sign language community, ultimately bridging the gap between sign language users and non-signers.

APPENDIX A: RESEARCH PAPER

Sign Language Recognition App with Python

Nandini Singh
Department of Artificial Intelligence and Data Science,
Dr. Akhilesh Das Gupta Institute of Technology and Management, New Delhi
Email: nandinisingh5may@gmail.com

Abstract - This project was developed with the aim of utilizing computer vision techniques to achieve real-time and highly accurate Sign Language recognition (SLR). Its purpose is to bridge the communication gap between individuals with varying levels of hearing abilities. To accomplish this, a dataset comprising images corresponding to signs was created and labeled accordingly. These images were then processed through a neural network that leverages transfer learning. Moreover, the project incorporates Mediapipe Holistic and the system utilizes Long Short-Term Memory (LSTM) architecture to model the temporal dynamics of the sign language gestures. Through the integration of these technologies, the project endeavors to enable accurate and real-time recognition of sign language.

Keywords **- Sign Language Recognition, Deep Learning, Computer Vision, Image Preprocessing, Transfer Learning, LSTM Neural Network, MediaPipe Holistic.**

INTRODUCTION

Sign language serves as a crucial mode of communication for individuals to express their thoughts and emotions non-verbally. However, there is a significant communication barrier between those familiar with sign language and those who are not, leading to the marginalization of individuals with hearing impairments. Sign language heavily relies on hand movements and facial expressions to convey meaning. While the hearing-impaired community frequently utilizes sign language for communication, it is rarely understood by those without hearing impairments. Consequently, social interactions are severely limited, and relying on real-time interpreters can be both impractical and costly. In this project, we will create a program using OpenCV to recognize and translate custom sign language gestures into regular text, aiming to bridge the communication gap and enable better understanding and inclusivity.

Fig 1: Skin Masked Images of different English Alphabets

1.1 Applications

A Python-based sign language recognition application has diverse practical uses in various fields such as communication, education, accessibility, and interactive experiences. One of its primary purposes is to enhance communication for individuals who are deaf or hard of hearing by converting sign language gestures into text or speech, enabling effective interaction with others. Moreover, the app serves as a valuable tool for learning sign language, providing visual feedback and explanations to aid in the learning process. By incorporating augmented reality capabilities, the application utilizes camera input to recognize gestures and overlay virtual elements, enhancing the user experience. Lastly, the app can be utilized for interactive sign language-based games and applications, promoting engagement and interactive learning. Overall, this application actively promotes inclusivity, accessibility, and effective communication for individuals with diverse communication needs.

1.2 Role of different fields

The technology and software development industry play a vital role in the creation of the sign language recognition app by providing the necessary tools and frameworks for app development. Ongoing research and development efforts in fields like computer vision, machine learning, and natural language processing contribute to enhancing the app's algorithms and improving gesture recognition capabilities. Collaboration with accessibility and inclusion organizations ensures that the app is designed to meet the specific needs of individuals with hearing impairments, incorporating their valuable guidance and support. Educational institutions and language learning platforms also play a crucial role by partnering to develop comprehensive sign language databases and effective learning methodologies, enhancing the app's functionality and educational value. Through these collective efforts, the technology, software development, accessibility, and education sectors collaborate to create a sign language recognition app that promotes accessibility, inclusion, and effective communication.

1.3 Recent Advancements in Sign Language Detection

In recent years, sign language detection has witnessed notable advancements driven by the application of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These deep learning models excel in capturing spatial and temporal patterns inherent in sign language gestures, leading to enhanced accuracy and robustness. The availability of extensive sign language datasets has played a crucial role in training these models, enabling them to learn intricate features and accommodate variations within sign language gestures. Furthermore, the incorporation of computer vision techniques, such as optical flow analysis and hand tracking algorithms, has contributed to the refinement of sign language detection systems, offering more precise and reliable tracking of hand movements. These combined developments in deep learning, increased data availability, and computer vision techniques have significantly elevated the performance of sign language detection and recognition systems, opening doors to improved communication and accessibility for individuals who use sign language.

1.4 Challenges

Developing a sign language recognition app comes with its own set of challenges. One major hurdle is the inherent variability present in sign language gestures. Sign language encompasses a wide range of movements and gestures that can vary across regions, individuals, and contexts. Accurately capturing and recognizing this variability requires robust algorithms and extensive training data to encompass the diverse range of possible gestures. The dynamic nature of sign language, including hand movements, facial expressions, and body language, adds another layer of complexity to the recognition process. Ensuring real-time and accurate detection of these dynamic features presents a significant challenge. Additionally, environmental factors such as lighting conditions and background clutter can impact the app's performance by interfering with gesture detection. Overcoming these challenges necessitates ongoing research, algorithm refinement, and the collection of diverse and representative datasets to train the models effectively.

2. LITERATURE REVIEW

Several research papers have contributed to advancements in sign language recognition. In recent research [1], a deep learning-based pipeline was developed to automatically recognize sign language from RGB videos. The pipeline utilized SSD, 2DCNN, 3DCNN, and LSTM, and introduced a novel representation of hand skeleton features. Additionally, 3DCNNs were applied to pixel level and heat map features to enhance discriminative capabilities. In a separate investigation [2], a deep CNN architecture was introduced for detecting and classifying sign languages, incorporating both static and dynamic gestures in the training process. Furthermore, a study [3] conducted a comparative analysis of machine learning and deep learning models for classifying American sign language, ensuring robustness through user-independent k-fold cross-validation and test phases. Another study [4] focused on real-time sign language recognition and proposed a model that combined a single shot detector, 2D CNN, SVD, and LSTM. Another approach [5] addressed human action recognition, utilizing motion tracking and feature extraction through a Recurrent Neural Network model with Gated Recurrent Unit. These advancements hold great promise for improving communication, accessibility, and inclusion

3. METHODOLOGY

This research proposes a method for sign language detection using OpenCV and machine learning techniques. The proposed method consists of the following steps:

3.1 General Design

In our sign language recognition project, we have developed a sign detector capable of detecting custom signs and easily expanding to include a wide range of additional signs and hand motions, such as the alphabet and numerals. To build this project, we utilized Python modules including OpenCV, Mediapipe, Tensorflow, and Keras. The OpenCV module processes live video frames from a camera, analyzing the actions of the person being displayed. Mediapipe Holistic is used to extract keypoints from the hands, torso, and face in the video frames. These keypoints are then fed into the prediction algorithm, which begins the real-time prediction of the sign being made. The predicted sign is displayed as the expected output.

3.2 Prerequisites

The prerequisites software & libraries:

1. Python (3.10.8)
2. IDE (Jupyter)
3. Mediapipe (version 0.10.1)
4. Numpy (version 1.23.5)
5. cv2 (openCV) (version 4.7.0.72)
6. Keras (version 2.12.0)
7. Tensorflow (version 2.12.0)

3.3 Dataset

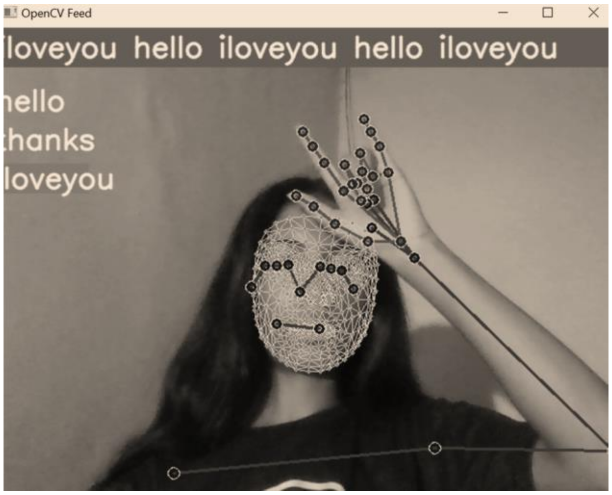
The dataset includes different motions and signals for sign language. A live video stream from the camera is constantly monitored, and frames within the defined region of interest (ROI) that show gestures or motions are saved in a dedicated directory called the gesture directory. Each sign is captured through approximately 30 video sequences, where each sequence contains 30 frames capturing significant moments. These frames are stored as Numpy arrays. The main goal is to accurately identify the specific gesture being performed throughout the entire video sequence.

3.4 Training

In Python, the training process utilizes the TensorFlow machine learning platform. To apply transfer learning, the datasets and label files are preprocessed into a format compatible with TensorFlow. Specifically, tfrecord files are generated from the training and testing data folders. Prior to training, the models designed for object detection need to be downloaded and adjusted in the configuration files to match the number of classes in the dataset. Achieving high accuracy typically requires around 2000 training steps. By combining TensorFlow and Keras, an LSTM model is developed to predict actions displayed on the screen, such as Sign Language gestures in this particular example.

3.5 *Testing*

To evaluate the performance of our model, extensive testing was conducted using deep neural networks and the Mediapipe Holistic framework. The models were trained to anticipate signals based on forearm, hand, and finger kinematics. Among the various configurations tested, the Mediapipe LSTM model with data augmentation demonstrated the highest accuracy, achieving an impressive 100 percent on the test sets. With this successful outcome, our sign language detector is capable of accurately recognizing and detecting hand movements, producing corresponding coordinates, and comprehending sign language gestures. The system operates in real-time, providing immediate and up-to-date signage interpretation.



4. CONCLUSION

In conclusion, the research aimed to create a real-time sign language detection and translation system using deep neural networks and Mediapipe Holistic. The obtained results were promising, indicating the potential for practical applications. The developed system achieved a high accuracy rate, with the Mediapipe LSTM model combined with data augmentation performing exceptionally well, reaching 100% accuracy on test sets. This system effectively recognizes and interprets hand gestures, providing coordinated outputs in real time.

5. FUTURE SCOPE

For future directions, the project team proposes the possibility of developing a mobile application that can classify complete word symbols by incorporating facial emotions and relative hand movements from the face. Furthermore, there are plans to expand the sign language dataset by incorporating common words in addition to alphabet letters. The aim is also to enhance the model's accuracy by introducing additional hyperparameters. These future steps will contribute to further advancements in sign language recognition and translation technology.

6. REFERENCE

[1] Razieh Rastgooa, Kourosh Kiania, Sergio Escalerab(2020), “Hand sign language recognition using multi-view hand skeleton”, Electrical and Computer Engineering Department, Semnan University, Department6of Mathematics and Informatics, Universitat de Barcelona and Computer Vision Center, Barcelona, Spain, , pp.16-24.

[2] R. Bhadra and S. Kar(2021), "Sign Language Detection from Hand Gesture Images using Deep Multi-layered Convolution Neural Network," 2021 IEEE Second International Conference on Control, Measurement and Instrumentation, pp.196-200.

[3] Neziha Jaouedia, Noureddine Boujnahb, Med Salim Bouhlelc(2020), “A new hybrid deep learning model for human action recognition.”, National Engineering School, Avenue Omar Ibn El Khattab Zrig Eddakhlania, Gabes 6072, pp. 447-453.

[4] Rastgoo, R., Kiani, K. & Escalera(2020), “Real-time isolated hand sign language recognition using deep networks and SVD.” J Ambient Intell Human Comput 13, pp. 591–611.

[5] Hernandez V, Suzuki T,(2020), “Convolutional and recurrent neural network for human activity recognition: Application on American sign language” Venture G, pp. 19-23.

APPENDIX B: SURVEY DATA OR TYPICAL PART OF SOURCE CODE

1.Setting-up Folders for Collection:

```
# Path for exported data, numpy arrays
DATA_PATH = os.path.join('MP_Data')

# Actions that we try to detect
actions = np.array(['hello', 'thanks', 'iloveyou'])

# Thirty videos worth of data
no_sequences = 30

# Videos are going to be 30 frames in length
sequence_length = 30

# Folder start
start_folder = 30
```

```
for action in actions:
    for sequence in range(no_sequences):
        try:
            os.makedirs(os.path.join(DATA_PATH, action, str(sequence)))
        except:
            pass
```

2.Building and Training LSTM Neural Network:

```
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,1662)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
```

3.Model Summary:

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_12 (LSTM)              (None, 30, 64)            442112

 lstm_13 (LSTM)              (None, 30, 128)           98816

 lstm_14 (LSTM)              (None, 64)                49408

 dense_11 (Dense)            (None, 64)                4160

 dense_12 (Dense)            (None, 32)                2080

 dense_13 (Dense)            (None, 3)                 99

=================================================================
Total params: 596,675
Trainable params: 596,675
Non-trainable params: 0
_________________________________________________________________
```

4.Testing in Real Time:

5.Code to Capture Real-Time Videos.

```
# 1. New detection variables
sequence = []
sentence = []
threshold = 0.8

cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():
        # Read feed
        ret, frame = cap.read()
        # Make detections
        image, results = mediapipe_detection(frame, holistic)
        print(results)
        # Draw landmarks
        draw_styled_landmarks(image, results)
        # 2. Prediction logic
        keypoints = extract_keypoints(results)
#         sequence.insert(0,keypoints)
#         sequence = sequence[:30]
        sequence.append(keypoints)
        sequence = sequence[-30:]
        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            print(actions[np.argmax(res)])
        #3. Viz logic
            if res[np.argmax(res)] > threshold:
                if len(sentence) > 0:
                    if actions[np.argmax(res)] != sentence[-1]:
                        sentence.append(actions[np.argmax(res)])
                else:
                    sentence.append(actions[np.argmax(res)])
            if len(sentence) > 5:
                sentence = sentence[-5:]
            # Viz probabilities
            image = prob_viz(res, actions, image, colors)

        cv2.rectangle(image, (0,0), (640, 40), (245, 117, 16), -1)
        cv2.putText(image, ' '.join(sentence), (3,30),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

        # Show to screen
        cv2.imshow('OpenCV Feed', image)
        # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

REFERENCES

[1] Singha, J. and Das, K. "Hand Gesture Recognition Based on Karhunen-Loeve Transform", Mobile and Embedded 232 Technology International Conference (MECON), January 17-18, 2013, India. 365-371.

[2] R. Sharma et al. Recognition of Single Handed Sign Language Gestures using Contour Tracing descriptor. Proceedings of the World Congress on Engineering 2013 Vol. II, WCE 2013, July 3 - 5, 2013, London, U.K

[3] R.Karthikeyan, C.Jothi Kumar, "Improved Reputation Systemfor Wireless Sensor Networks (WSNS) ", International Journal of Innovations in Scientific and Engineering Research (IJISER), Vol.1, no.3, pp.191-195, 2014.

[4] M. Jeballi et al. Extension of Hidden Markov Model for Recognizing Large Vocabulary of Sign Language. International Journal of Artificial Intelligence & Applications 4(2); 35-42, 2013

[5] T.Starner and A. Pentland. Real-Time American Sign Language Recognition from Video Using Hidden Markov Models. International Journal of Pure and Applied Mathematics Special Issue 1691 Computational Imaging and Vision, 9(1); 227-243, 1997.

[6] Y.F. Admasu, and K. Raimond, Ethiopian Sign Language Recognition Using Artificial Neural Network. 10th International Conference on Intelligent Systems Design and Applications, 2010. 995-1000

[7] L. Pigou et al. Sign Language Recognition Using Convolutional Neural Networks. European Conference on Computer Vision 6-12 September 2014

[8] Razieh Rastgooa, Kourosh Kiania, Sergio Escalerab(2020), "Hand sign language recognition using multi-view hand skeleton", Electrical and Computer Engineering Department, Semnan University, Department6of Mathematics and Informatics, Universitat de Barcelona and Computer Vision Center, Barcelona, Spain, , pp.16-24.

[9] R. Bhadra and S. Kar(2021), "Sign Language Detection from Hand Gesture Images using Deep Multi-layered Convolution Neural Network," 2021 IEEE Second International Conference on Control, Measurement and Instrumentation, pp.196- 200.

[10] Neziha Jaouedia, Noureddine Boujnahb, Med Salim Bouhlelc(2020), "A new hybrid deep learning model for human action recognition.", National Engineering School, Avenue Omar Ibn El Khattab Zrig Eddakhlania, Gabes 6072, pp. 447-453.

[11] Rastgoo, R., Kiani, K. & Escalera(2020), "Real-time isolated hand sign language recognition using deep networks and SVD." J Ambient Intell Human Comput 13, pp. 591–611.

[12] Hernandez V, Suzuki T,(2020), "Convolutional and recurrent neural network for human activity recognition: Application on American sign language" Venture G, pp. 19-23.