iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60302

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Implementation on FingerScripter: Converting Fingerspelling into Text and Speech

Prof. Abhijit Shinde[1], Haridas Pawar[2], Pratham Pardeshi[3], Ashutosh Shirke[4], Udant Udapure[5]
Department of Information Technology, Sinhgad College of Engineering, Pune

Abstract: ***Communication might seem a simple act for normal people, but for the ones who are suffering from physical or psychological disability find it very difficult for communicating with others. This communication barrier needs to be removed for which there are various techniques that are being used, out of which some are verbal, non-verbal, visual and written. For mute and deaf people these communication barrier is a major problem which can adversely affect their lives adding to their already challenging lives. For this sign languages are the best possible option, as they bridge the gap between communication of these deaf and mute people with normal people. We have created a machine learning model in order to solve this issue which will identify these signs and interpret them to detect which alphabet is signified. The model will be detecting the American Sign Language (ASL) which is one of the most widely used sign language and a type of fingerspelling. The model will form words out of the identified alphabets which will also be then converted into speech. The key algorithm which will be used is a special type of Convolutional Neural Network (CNN) for image recognition. All the documentation regarding the model along with its working is provided in this paper. With high precision and accuracy the model aims to ease the communication of deaf and mute people.***

Keywords: Mediapipe Hand Tracking, Convolutional Neural Network, Sign Language Detection, Hand sign, Image classification.

I. INTRODUCTION

Physical and psychological disabilities have made communication very challenging for deaf and mute people. The World Health Organization (WHO) [1] estimates that there are more than 1.5 billion deaf and mute persons in the world. Subsequently, this society uses sign language to communicate with both regular people and each other. However, the most frequent issue that comes up is that most regular people are not aware of this sign language, which effectively isolates the deaf and mute group from the general public. Over 300 different sign languages are presently in use worldwide, according to Sign Solutions [2]. Sign languages such as American Sign Language (ASL), British Sign Language (BSL), French Sign Language (FSL), Japanese Sign Language (JSL), and others differ from a particular nation to the next. If many languages are spoken in the same nation, such as Hindi, Marathi, Gujarati, Malayalam, and many more in India, this might result in the development of numerous sign languages. Sign language can undergo small alterations due to the presence of many regional accents, even in places where only one language is spoken. The American Sign Language is the most commonly used sign language among all of these. This is mostly due to two factors: first, it is a fingerspelling language, which is the foundation of sign language and is thus utilized in the educational process. Second, it is simple to learn since it is a one-handed sign language. The deaf and mute population now needs interpreters in order to connect with the general public. Interpreters are those who can read sign language, interpret it, and explain its meaning to non-deaf people. They are necessary in legal, medical, and educational settings, but due of their high expense, they may be an overhead.

We have developed a machine learning model that translates American Sign Language into text and speech in order to address all of these problems. The device's camera is used by the model to capture photos in real time while the user makes hand gestures. The Mediapipe Hand Landmark model based on Convolutional Neural Network (CNN) is being used to identify and detect these hand signals. The CNN is a multi-layered deep learning system with an emphasis on image processing. The model interprets these pictures for the alphabet. After that, a word is created by combining these alphabets. We have also offered the customer with three word ideas from which to choose, to help and expedite this process. Next, words are translated into voice and gives output through the device's speaker. A full discussion of the existing work on sign language identification can be found in the literature review. To solve these machine learning difficulties, they mostly employ deep learning (DL). A kind of machine learning called "deep learning" is based on the principle of "learning by example." This approach to artificial intelligence (AI) trains machines to handle data in a manner that is modelled after the human brain. One such deep learning system is CNN, which processes images with exceptional accuracy and whose description is provided in full below.

The paper consists of following contents: Related Works were we have discussed research done on the topic till now. Open Issues showcasing a detailed description of drawbacks in the currently used technologies. Objectives of the project and Methodology were we have provided a clear explanation of our model's approach. Algorithm gives a detailed explanation regarding the algorithm used in the project. Flowchart depicting the flow of processes in the model. Acknowledgement as a vote of thanks. Implementation specify the execution of project and Technologies used discusses all the various technologies used. Result depicting the precision and accuracy of the model and lastly the Conclusion presenting the closing statements followed by references.

II. RELATED WORK

Given its significant impact on the deaf and mute communities, sign language detection has been thoroughly studied. While these studies focused on different areas, they all aimed to increase the effectiveness of sign language recognition. Five criteria are commonly used in American Sign Language (ASL) [6], which increases detection efficiency. The hand's form, palm orientation, motion, position, and manual expressiveness are among these factors. Some of these settings provide usefulness, but they also complicate detection. Various studies concentrated on creating models for sign language detection using different techniques, which are elaborated upon below.

A.Sharmila Konwar, B.Sagarika Borah,C.Dr. T.Tuithung [3] presented a model for hand gesture detection using HSV color model and canny edge detection algorithm. Rather than emphasizing identification, their study mostly concentrated on hand sign detection. To identify the skin, the skin's border was specified using the HSV (Hue, Saturation, and Luminance) color model. On the other hand, the Canny Edge detection method was further used to identify the skin's edges in order to identify the entire hand sign. By obtaining unique edges through morphological operations, the detection was further enhanced. Lastly, they mentioned that feature extraction and recognition might be accomplished by PCA and ANN. They said that the accuracy of their detecting technology was around 65%.

Yifan Zhang, Ling Long, Diwei Shi, Haowen He, Xiaoyu Liu [4] proposed a sign recognition and detection model based on improved YOLOv5 algorithm. They instructed their model to recognize motions in Chinese Sign Language. Object identification, instance segmentation, and picture classification tasks are performed with YOLOv5. Its structure is comprised of CSP and Focus structures for slicing, as well as Mosaic data improvement for input. For feature extraction, it also includes CBS, Upsample, Concat, and CSP. To enhance the model's capacity for detection and identification, CBAM attention mechanism and EIOU_Loss were added in addition to YOLOv5. The accuracy rate recorded by the model was 96.05%.

Mihir Deshpande, Vedant Gokhale, Adwait Gharpure, Ayush Gore [5] their paper proposed the use of LSTM Deep Learning Model and Media Pipe. Media Pipe is used to map out 21 spots on the user's hand and detect hand motions. Furthermore, for sign identification and prediction, LSTM (Long Short Term Memory), an artificial neural network with four layers of artificial neurons, is also utilized. In addition, OpenCV is used to acquire data by capturing picture frames from a video stream. During testing, the accuracy was 99%, and more gains might be made by applying sophisticated methods such as transfer learning, data augmentation, and hyper parameter adjustment for a wider range of datasets.

III. OBJECTIVES

1) *Develop Robust Sign Language Detection:* Implement advanced computer vision and machine learning algorithms to accurately recognize and interpret a wide range of sign language gestures.
2) *Real Time Translation:* Enable real-time translation of sign language gestures into text and audible speech to facilitate immediate and seamless communication.
3) *User-Friendly Interfaces:* Develop intuitive and user-friendly interfaces that are accessible to both deaf and mute individuals and those who may not be familiar with sign language.
4) *Collaboration with Deaf Community:* Collaborate closely with the deaf community, involving them in the development process to ensure that the technology aligns with their needs and preferences.

IV. METHODOLOGY

The implemented system not only functions well and is user-friendly, but it also addresses all of the shortcomings of the previous system. Therefore, we created an application for this technology that has an interface and focuses mostly on translating fingerspelling movements into alphabets and subsequently words. Convolutional neural networks, together with a few machine learning and deep learning techniques, were utilized in the development of the program. In addition, other Python libraries will be utilized for distinct tasks.

The created solution circumvents all of the shortcomings of the existing system and is easy to use and perform effectively. So, we developed an application with a user interface that is mainly concerned with converting finger spelling gestures into alphabets, which are then translated into words. The program is developed using convolutional neural networks, a few machine learning and deep learning techniques, and a range of Python modules for different activities.

Here, we have created a user-friendly interface that can recognize motions and convert them into text. We've also automated a rectification function that can instantly correct misspelled words with the use of Python tools. It records every motion, converts it into an alphabet, and then blends all the letters to produce a word, allowing individuals who are not acquainted with sign language to comprehend exactly what the dumb and deaf say. As a result, neither having to spend money nor remembering every meaning of a certain gesture is required in order to buy those devices.

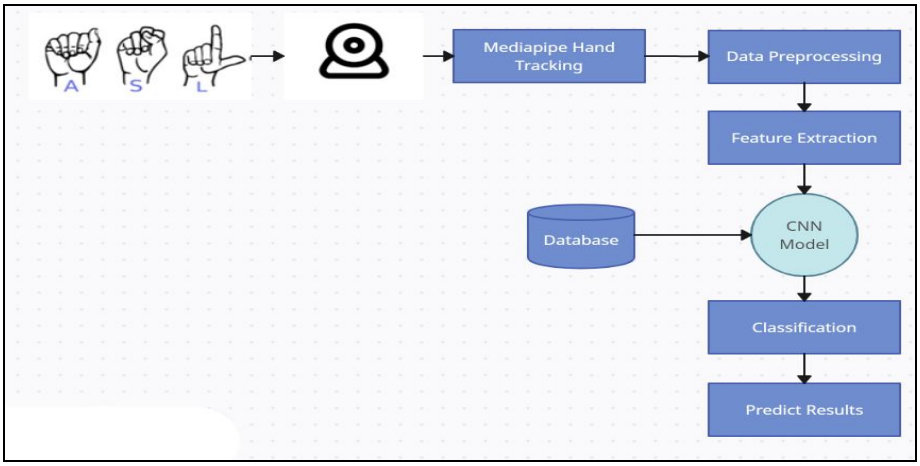


Fig. 1 System Architecture

A. *Dataset Generation*

We are going to collect a range of hand sign images to build our dataset using the OpenCV Python object identification library. We utilize a technique that captures training and test pictures using the system's camera. Despite our initial search for pre-existing datasets, we encountered difficulties because the majority of the datasets that were found didn't fit our requirements. This has led us to create our own dataset.

Throughout the data collection process, every frame the camera takes will be recorded. The Bounding Box Regression technique [11] will be used to define and visually identify a Region of Interest (ROI) inside each frame. Utilizing the Inception model on all of the collected photographs is the most crucial step to do. After all the pictures have been collected, we will use the Inception Model to process them. In order to extract various features and prepare the dataset for our next hand sign recognition projects, this is an essential step.

B. *Feature Extraction*

After capturing frames using OpenCV, we will utilize the Inception Model and the Mediapipe Hand Tracking in our following work to facilitate feature extraction. This phase is critical to reduce noise and highlight significant visual elements in the assembled frames. Next, our Mediapipe Hand Landmark model [12] will utilize the processed picture as input to generate predictions. A letter will be identified, recorded, and taken into account for word creation once the model has been trained to identify hand signals and letters. This process will take longer than 50 frames. In order to enable precise word creation in our future image processing and recognition system, we will utilize a blank sign to represent word space.

C. *Model Implementation*

To create the dataset for American Sign Language (ASL) recognition, hand gestures corresponding to each letter of the alphabet were photographed using the Inception Model. Then, Mediapipe Hand Landmark, a hand-tracking library, made it feasible to quickly and reliably detect hand landmarks and track hand movements in real time. Subsequently, the captured hand gestures were overlaid onto an image of a white background, providing an impartial and consistent background for image processing and classification. With the provided dataset, machine learning models for assistive technology and ASL recognition might be developed.

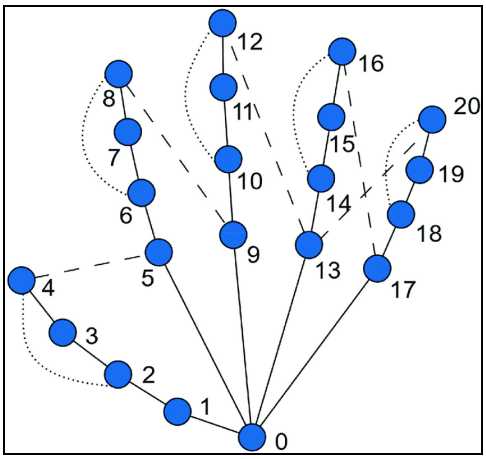


Fig. 2 Mediapipe Hand Landmark

TABLE I
LANDMARKS ON PALM

0	Wrist	10	Middle finger proximal interphalangeal joint
1	Thumb carpometacarpal joint	11	Middle finger distal interphalangeal joint
2	Thumb metacarpophalangeal joint	12	Middle finger tip
3	Thumb interphalangeal joint	13	Ring finger metacarpophalangeal joint
4	Thumb tip	14	Ring finger proximal interphalangeal joint
5	Index finger metacarpophalangeal joint	15	Ring finger distal interphalangeal joint
6	Index finger proximal interphalangeal joint	16	Ring finger tip
7	Index finger distal interphalangeal joint	17	Little finger metacarpophalangeal joint
8	Index finger tip	18	Little finger proximal interphalangeal joint
9	Middle finger metacarpophalangeal joint	19	Little finger distal interphalangeal joint
		20	Little finger tip

A. Text to Speech

Our approach guarantees accessibility and ease of use by converting the final text output to voice or audio format. By using this text-to-speech feature, the system can pronounce the words it has recognized. This enables effective communication without the requirement for specialized libraries and provides an adaptable and entertaining user interface.

V. ALGORITHM

Convolutional neural networks (CNNs) are a regularized kind of feed forward neural networks that use filter optimization to teach themselves feature engineering. It is a particular kind of artificial neural network designed to analyze pixel input and recognize images. A Convolution Neural Network (ConvNet/CNN) is a Deep Learning method that can recognize distinct objects and attributes in an input picture and distinguish between them by assigning weights and biases that may be learned. A ConvNet requires a lot less pre-processing than other classification techniques. ConvNet design was inspired by the structure of the visual cortex and is comparable to the connection pattern of neurons in the human brain. Only in a small area of the visual field known as the Receptive Field do individual neurons react to inputs. The whole visual field is covered by an overlap of these fields. Capturing local spatial patterns is an excellent use of convolutional neural networks, or ConvNets.

They excel at identifying patterns and using those to categorize pictures. ConvNets make the explicit assumption that an image will be the network's input. CNNs classify an image and its rotated version as the same picture because they have pooling layers, which make them insensitive to the translation or rotation of two identical images. We have utilized the TensorFlow library's Inception-v3 model, a deep ConvNet, to extract spatial information from the frames of video sequences because of CNN's significant advantages in extracting spatial features from images. With millions of parameters, Inception is a massive image classification model that can categorize pictures.

Inception-v3 is a pre-trained convolutional neural network that is 48 layers deep, which is a version of the network already trained on more than a million images from the ImageNet database. This pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

We are using it in our model to detect hand signals since it is perfect for identifying and recognizing items. It is made up of several layers, which are as follows:

1) *Input Layer:* The raw images are given as input to the CNN through the input layer. Here, user's hand sign images are given as input to this layer.
2) *Hidden Layers:* In order to learn features specific to the data, these layers carry out operations that modify the data. Convolutional, pooling, and fully connected (FC) layers are among them often. In our scenario, these layers are used to analyze the hand gesture photos.
3) *Convolution Layer:* It is made up of an image convolutional kernel, or matrix of numbers, which is used to extract features from the input pictures. The number of kernels determines the layer's performance. Several characteristics, including color, contours, and edges, will be retrieved in this case.
4) *Pooling Layer:* They are primarily used to shrink the spatial size of the representation, which lowers overfitting (information loss) by lowering the number of parameters and calculations in the network. There are two types of pooling Max pooling and Average pooling.
5) *Fully Connected Layer:* Only a small region is connected to neurons in the convolution layer; in a fully connected region, all inputs are coupled to neurons. Based on the attributes that were retrieved from the earlier levels, images are classified in this layer. The photos will be divided into many classes in this instance.
6) *Dropout Layer:* This layer cancels out certain neurons' contributions to the subsequent layer, in order to prevent overfitting.
7) *Output Layer:* In the convolution layer neurons are connected only to a local region, while in a fully connected region, well connect all the inputs to neurons. Once the completely linked layer's values are obtained, we'll connect them to the last layer of neurons, whose count equals the entire number of classes. This layer will forecast the likelihood that each image belongs to a distinct class.

VI. IMPLEMENTATION

The project is implemented by importing essential libraries, including numpy for numerical computations, cv2 for image processing with OpenCV, pyttsx3 for text-to-speech conversion, keras for loading a pre-trained Convolutional Neural Network (CNN) model, enchant for word suggestions and tkinter for developing the graphical user interface (GUI). These imports ensure that the necessary tools and functionalities are available for the project's implementation.

Upon initialization, the Application class is defined, which serves as the backbone for setting up various components of the application. This includes initializing the webcam to capture video feed, loading a pre-trained CNN model for hand gesture recognition, setting up GUI elements using Tkinter for creating a user-friendly interface, and utilizing the Hand Tracking Module for detecting hands within the captured frames.

The GUI setup involves creating the main window using Tkinter and populating it with labels to display instructions, real time captured images, recognized symbols, and suggestions. Additionally, buttons are incorporated for speaking the recognized text and clearing the input, enhancing user interaction and control over the application's functionalities.

Within the video loop, frames are continuously captured from the webcam, and hand gestures are detected using the Hand Tracking Module. The hand-tracking library provides a fast and reliable way to detect hand landmarks and track hand movements in real time. The captured hand gestures are then mapped onto a white background image, which provides a consistent and neutral background for image processing and classification. The resulting dataset can be used to train machine learning models for ASL recognition.

The library first detects the hand regions in the input image using a bounding box regression algorithm. It then feeds the detected hand regions to the hand landmark model to estimate the landmarks of each hand. The hand landmarks are a set of 21 2D points that represent the joints and fingertips of the hand.

Text manipulation involves constructing a word representing fingerspelled text using the recognized symbols, with various conditions and rules applied to handle different gestures and improve recognition accuracy. This ensures that the recognized text accurately reflects the user's intended message conveyed through fingerspelling.

GUI interaction allows users to view the recognized text in real-time, select suggestions provided by the enchant library, or manually edit the text using buttons, thereby providing flexibility and customization options. Additionally, functionality is implemented for speaking the recognized text and clearing the input, further enhancing user experience and accessibility.

Finally, speech synthesis is achieved using the pyttsx3 library, enabling the conversion of recognized text into speech. This functionality allows users to audibly hear the interpreted text, facilitating communication with individuals who use sign language and ensuring inclusivity and accessibility within the application.

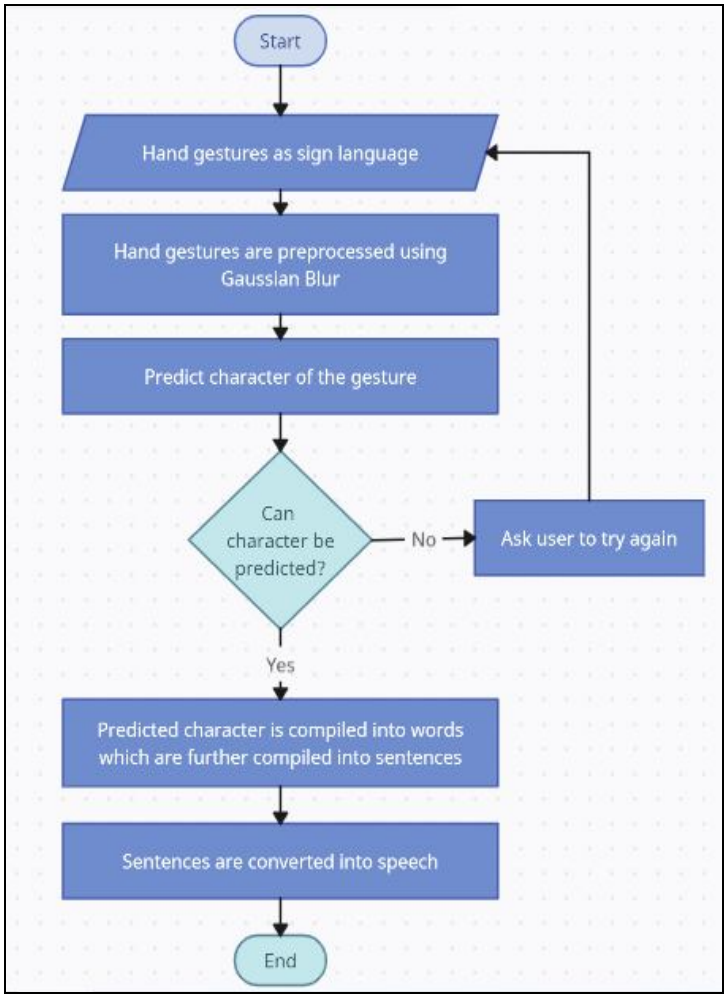


Fig. 3 Flow of the Study

VII. TECHNOLOGY USED

Various types of technologies are used in making of FingerScripter model which are discussed below:

1) *Python:* Used as the core programming language for model building.
2) *OpenCV:* Open Source Computer Vision is a library which is used for performing computer vision task.
3) *Tensorflow and Keras:* Used to execute deep learning, providing intuitive user interfaces, trained models, and effective GPU acceleration.
4) *Mediapipe Hand Landmark:* Library provided by Google which detects the landmark of the hand in an image.
5) *Enchant:* It is a python library which is used for spell checking based on which suggestions are provided in the model.
6) *Pyttsx3:* This library is used for conversion of text to speech even in offline mode, provided by python.
7) *Visual Studio Code:* Chosen as the Integrated Development Environment for the development of model
8) *Tkinter:* A Python library used to create GUIs, or graphical user interfaces, that facilitate communication between the user and the program.
9) *Windows 10:* This operating system have been used because of its reliability, accessibility and ease.

VIII. RESULTS

A variety of libraries and tools have been combined to create a comprehensive model for recognition and interpretation of hand gestures with high efficiency and accuracy.

A. Sign Language Dataset

We have created our own dataset of hand sign images using OpenCV library of the python which consists of 28 classes. Out of which 26 classes are for the 26 alphabets and 2 more classes are for the Next and backspace sign which are used to insert the detected sign into word or to remove it respectively. Each of these classes has around 179 images in them which makes the total number of images in the dataset to be around 5012.

B. Setup

Python programming language is used for experimentation, model evaluations, and data pre-processing. The implemented model have been developed using OpenCV, Tensorflow, Keras, Mediapipe Hand Tracking, enchant, pyttsx3 and tkinter.

C. Evaluation Metrics

A confusion matrix-based method is used to estimate accuracy, precision and recall in order to assess the performance of our implemented model.

1) *Accuracy:* It is a performance metric used in machine learning to evaluate the overall correctness of predictions made by a model.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) *Precision:* It is calculated as the ratio of true positive predictions to the sum of true positives and false positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) *Recall:* It is calculated as the ratio of true positive predictions to the sum of true positives and false negatives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) *F1-Score:* It is a harmonic mean of precision and recall.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

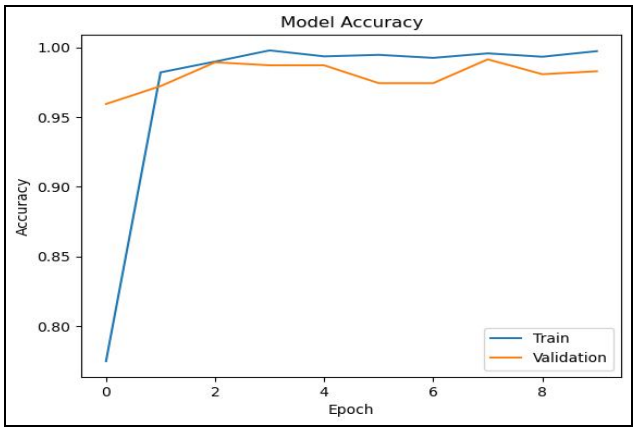


Fig. 4 Model Accuracy

D. Result Analysis

TABLE III
RESULT OBTAINED

Metrics	Model Performance
Accuracy	0.981
Precision	0.971
Recall	0.986
F1 score	0.978

Evaluation metrics [13] such as precision, accuracy, recall and F1 score are used. Model precision is calculated to be 97.1%, followed by accuracy of 98.1%, recall of about 98.6% and F1 score of about 97.8%. To compute accuracy, precision, and recall, we take the values of the true positives, true negatives, false positives, and false negatives from the confusion matrix.

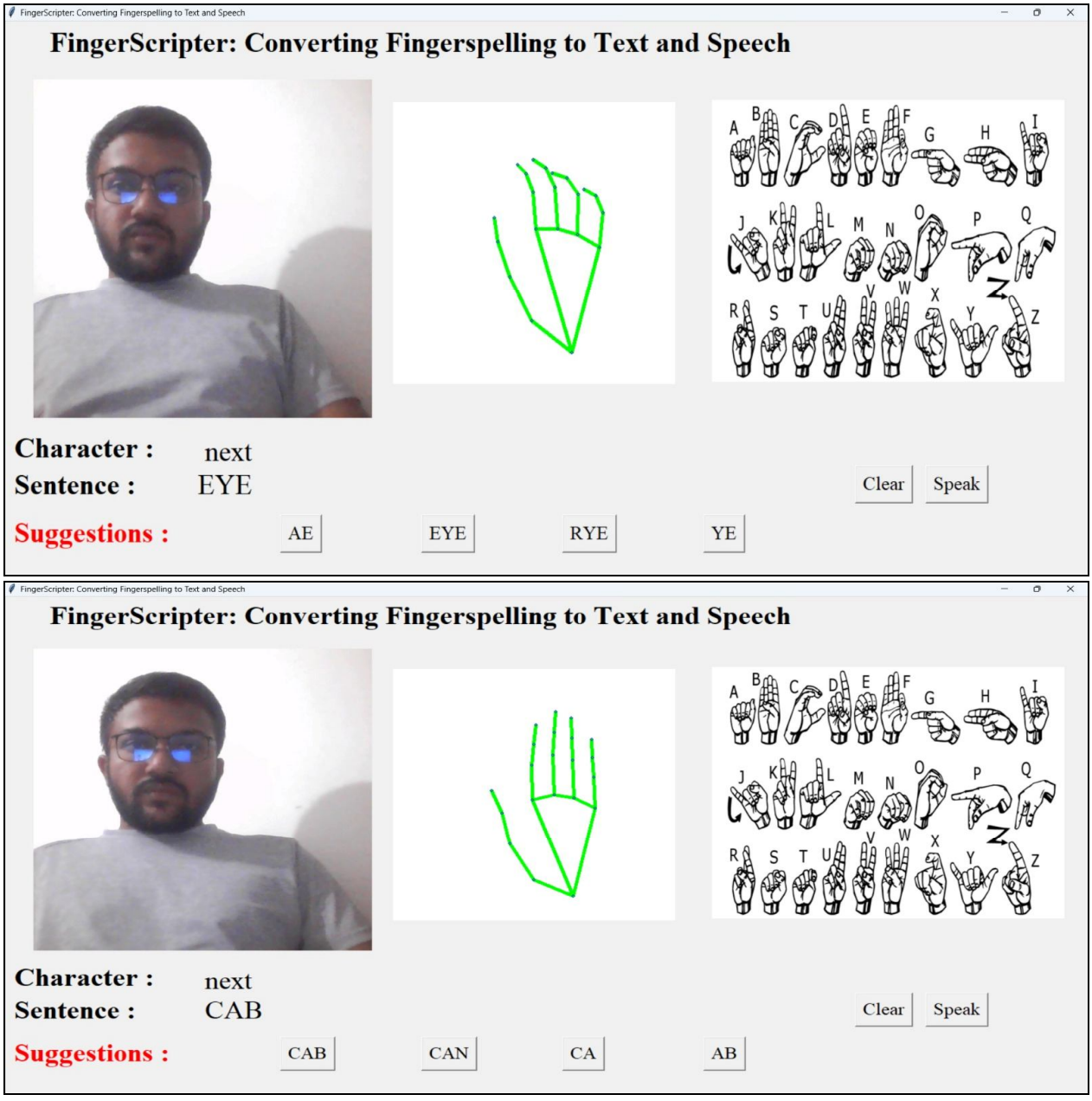


Fig. 5 Model User Interface

IX. CONCLUSION

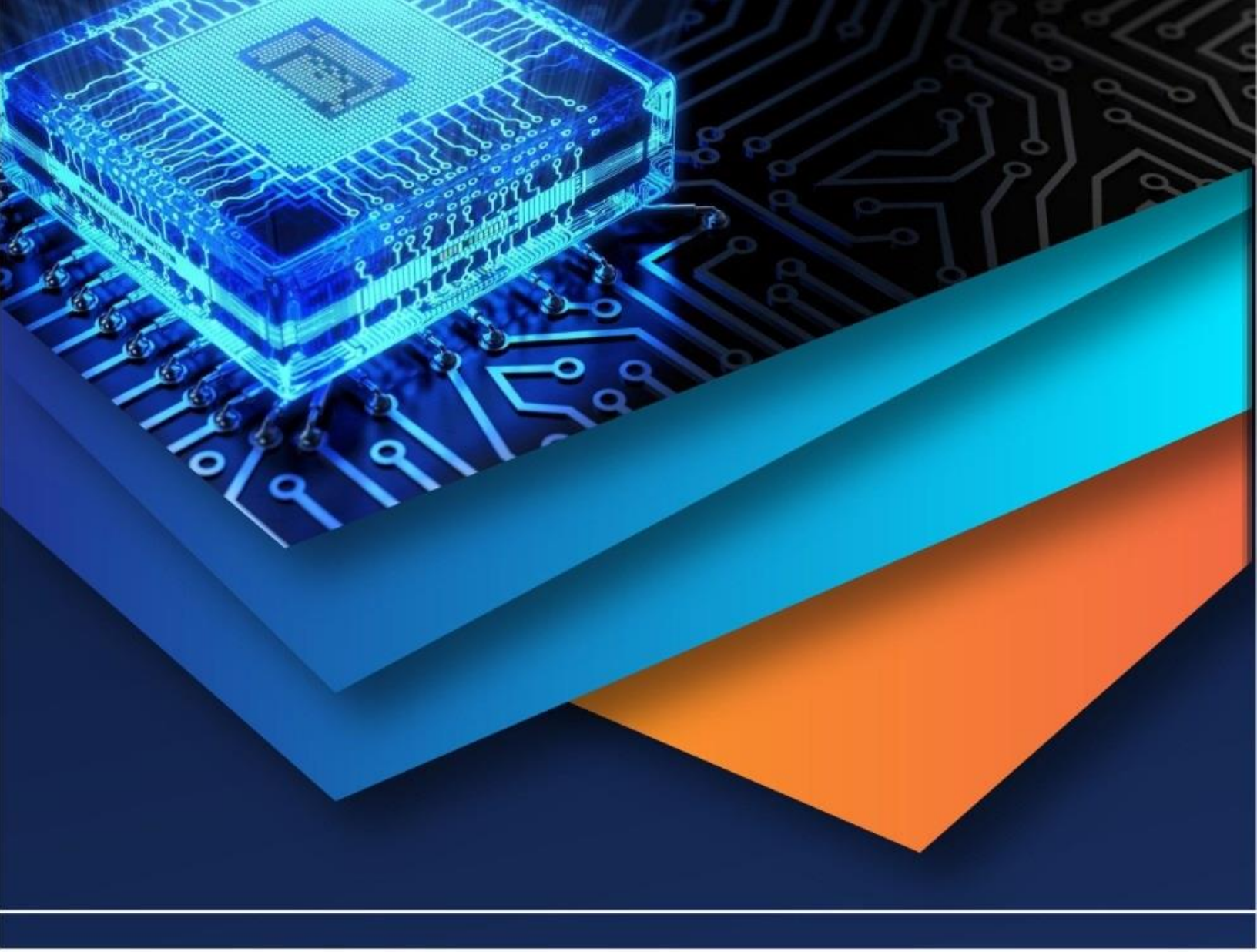
The established model is essential in the process of learning and using American Sign Language, which minimizes the difficulties faced by the deaf and mute people. Our approach distinguishes itself from existing models by enabling real-time detection, interpretation, and voice generation of the interpreted text. In our model, Mediapipe Hand Tracking which is built on a Convolutional Neural Network, provides the highest level of image processing accuracy available. For its additional features, it also uses a variety of Python libraries. This study provides a thorough description of the development, implementation and working of a highly accurate and efficient real-time vision-based sign language recognition model.

X. ACKNOWLEDGMENT

We would like to express our deepest appreciation to Principal Dr. S. D. Lokhande for providing all the facilities necessary for completing this research. This research would not have been possible without the support from our Head of Department Dr. S. R. Ganorkar. We are extremely grateful to our Internal Guide Prof. A. S. Shinde who generously provided knowledge and expertise during the research. We are deeply indebted to our internal Reviewers Prof. A. A. Utikar and Prof. M. S. Bhosale for their valuable guidance and patience. A special thanks to all the staff members of our department teaching as well as non-teaching for their moral support. We would also like to acknowledge all our fellow classmates for their advice and suggestions. Lastly, we would like to thank our families who believed in us and always kept our spirits soaring high.

REFERENCES

[1] World Health Organization (WHO).https://www.who.int/health-topics/hearing- loss#tab tab _ Gives information regarding number of deaf and dumb.
[2] Sign Solutions. https://www.signsolutions.uk.com/what-are-the-different-types-of-sign- language/
[3] A. S. Konwar, B. S. Borah and C. T. Tuithung, "An American Sign Language detection system using HSV color model and edge detection," 2014 International Conference on Communication and Signal Processing, Melmaruvathur, India, 2014, pp. 743-747, doi: 10.1109/ICCSP.2014.6949942.
[4] Y. Zhang, L. Long, D. Shi, H. He and X. Liu, "Research and Improvement of Chinese Sign Language Detection Algorithm Based on YOLOv5s," 2022 2nd International Conference on Networking, Communications and Information Technology (NetCIT), Manchester, United Kingdom, 2022, pp. 577-581, doi: 10.1109/NetCIT57419.2022.00137.
[5] M. Deshpande et al., "Sign Language Detection using LSTM Deep Learning Model and Media Pipe Holistic Approach," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, 2023, pp. 1072-1075, doi: 10.1109/AISC56616.2023.10085375.
[6] Mt. San Antonio College.https://www.mtsac.edu/llc/passportrewards/languagepartners/5ParametersofASL.pdf
[7] Convolutional Neural Network https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural networks/#:~ text=The%20firs t%20fully%20connected% 20 layer ,final%20probabilities%20for%20each%20label.
[8] Dr. J. Suresh Babu, Akumalla Veena, P. Chaturya, “Conversion of Gesture Language to text using OpenCV and CNN” 2022 International Journal of Research in Engineering, IT and Social Sciences.
[9] M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta and M. Aktaruzzaman, "A New Benchmark on American Sign Language Recognition using Convolutional Neural Network," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9067974.
[10] Ghali Upendra, Kokkiligadda Bhuvanendra, D. Gayathri, described in their paper “Sign Language Interpreter using Convolutional Neural Network” 2021 International Research Journal of Engineering and Technology.
[11] Bound Box Regression Information: https://pyimagesearch.com/2020/10/05/object- detection-bounding-box-regression-with-keras-tensorflow-and-deep-learning/
[12] Mediapipe Hand Land marking: https://developers.google.com/mediapipe/solutions/vision/ hand_landmarker
[13] Evaluation Metrics Used in the model are discussed briefly here https://www.evidentlyai.com/classification metrics/accuracy precision recall# :~:t ex t=Accuracy%shows%20how%20often%20a,when%20choosing%20the%20suitable%20metric.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY