ABSTRACT

Indian Sign Language (ISL) is the predominant sign language used by the deaf community in India. It is a complex and intricate language, with its own grammar and syntax, and is used by millions of people in India. However, due to the lack of awareness and resources, many individuals with hearing impairments face significant communication barriers in their daily lives, making it difficult for them to access education, healthcare, and other essential services.

To address this challenge, researchers and developers have been working on creating an Indian Sign Language Translator (ISLT), a software tool that can convert spoken or written language into ISL, and vice versa. The ISLT would enable hearing individuals to communicate with deaf individuals in real-time, facilitating more efficient and effective communication.

The development of an ISLT presents several technical challenges, including the need for accurate recognition and translation of sign language gestures and expressions. These challenges are being addressed through the use of advanced machine learning techniques, such as deep neural networks, which can learn from large amounts of data to accurately recognize and translate sign language.

The potential impact of an ISLT is significant, as it could improve the quality of life for millions of individuals with hearing impairments in India. It could enable them to communicate more effectively with their hearing peers, access educational and employment opportunities, and participate more fully in their communities.

The development of an ISLT also presents opportunities for further research and development in the field of natural language processing, as the translation of sign language involves unique linguistic features that differ from spoken and written languages.

In conclusion, the development of an Indian Sign Language Translator has the potential to significantly improve the lives of individuals with hearing impairments in India, and presents exciting opportunities for further advancements in the field of natural language processing.

Table of Contents

Introduction

The Indian Sign Language (ISL) is a visual language that is widely used by the deaf community in India. However, communication barriers exist between those who use sign language and those who do not, which can lead to difficulties in accessing essential services, education, and employment opportunities. To address this challenge, researchers and developers are working on creating an Indian Sign Language Translator (ISLT) - a software tool that can convert spoken or written language into ISL, and vice versa. The development of an ISLT has the potential to improve the quality of life for millions of individuals with hearing impairments in India, enabling them to communicate more effectively with their hearing peers and participate more fully in their communities. The technical challenges of recognizing and translating sign language gestures and expressions are being addressed through the use of advanced machine learning techniques. The development of an ISLT presents exciting opportunities for further research and development in the field of natural language processing, as the translation of sign language involves unique linguistic features that differ from spoken and written languages.

Indian Sign Language (ISL) is a visual language that is used by millions of individuals in India who are deaf or hard of hearing. It is a complex and intricate language, with its own grammar and syntax, and is essential for effective communication among members of the deaf community. However, communication barriers exist between those who use ISL and those who do not, which can lead to difficulties in accessing education, healthcare, and other essential services. To address this challenge, researchers and developers are working on creating an Indian Sign Language Translator (ISLT) - a software tool that can convert spoken or written language into ISL, and vice versa. The development of an ISLT has the potential to improve the quality of life for individuals with hearing impairments in India, enabling them to communicate more effectively with their hearing peers and participate more fully in their communities. This technology presents exciting opportunities for further research and development in the field of natural language processing, as the translation of sign language involves unique linguistic features that differ from spoken and written languages.

Problem Statement

The problem statement for the Indian Sign Language Translator (ISLT) is that the communication barriers between individuals who use ISL and those who do not hinder the effective provision of essential services, education, and employment opportunities for individuals with hearing impairments in India. The lack of awareness and resources for ISL communication further exacerbates this problem, resulting in social exclusion and limited access to information. To address this challenge, the development of an ISLT is crucial to provide a solution that can convert spoken or written language into ISL, and vice versa, enabling effective communication between the deaf and hearing communities in India.

The lack of access to an ISLT results in limited opportunities for education and employment for individuals with hearing impairments in India, leading to social and economic exclusion.
The current methods of communication between the deaf and hearing communities, such as the use of interpreters or written communication, are often unreliable and inefficient, resulting in miscommunication and frustration.

The complex grammar and syntax of ISL pose significant challenges for the development of an accurate and reliable ISLT, requiring advanced machine learning techniques and data sets specific to Indian Sign Language.

The limited availability of resources and support for ISL education and training further compounds the communication barriers between the deaf and hearing communities in India, highlighting the need for innovative solutions like the ISLT.
The development and deployment of an ISLT must consider the diverse regional variations and dialects of ISL in India, requiring significant research and collaboration with the deaf community to ensure the technology's effectiveness and accessibility.

Project Objectives

1. Develop a robust and accurate ISLT that can recognize and translate Indian Sign Language gestures and expressions in real-time.

2. Ensure the accessibility and usability of the ISLT for individuals with hearing impairments, with features such as a user-friendly interface, adjustable font size, and compatibility with various devices.

3. Develop a comprehensive database of ISL signs and expressions that can be used to train the ISLT and improve its accuracy over time.

4. Conduct user testing and validation of the ISLT with members of the deaf community to ensure its effectiveness and user-friendliness in real-world situations.

5. Develop and implement strategies for promoting awareness and education about ISL and the ISLT among hearing individuals and organizations to increase its adoption and impact.

6. Continuously update and improve the ISLT based on user feedback and technological advancements to ensure its long-term effectiveness and sustainability.

Project Scope

1. Develop an ISLT system that can recognize and translate Indian Sign Language (ISL) gestures and expressions in real-time.

2. Implement the ISLT system as a web or mobile application that is accessible to individuals with hearing impairments in India.

3. Develop a comprehensive database of ISL signs and expressions that can be used to train the ISLT system and improve its accuracy over time.

4. Conduct user testing and validation of the ISLT system with members of the deaf community to ensure its effectiveness and user-friendliness in real-world situations.

5. Develop and implement strategies for promoting awareness and education about ISL and the ISLT system among hearing individuals and organizations to increase its adoption and impact.

6. The scope of the project may also include integrating the ISLT system with other technologies, such as speech recognition software or natural language processing, to enhance its functionality and accuracy.

7. The project may also include exploring the use of augmented reality or other technologies to provide visual feedback for the user and improve the accuracy and usability of the ISLT system.

8. The project will require collaboration with experts in ISL linguistics, machine learning, and software development to ensure the accuracy and effectiveness of the ISLT system.

Literature Review

1. **"Indian Sign Language Recognition and Translation: A Review**" by B. Bharathi, S. Ganapathy, and S. S. Kumar. This paper reviews the state-of-the-art techniques and methodologies for Indian Sign Language recognition and translation, highlighting the challenges and opportunities for research in this field.

   The paper discusses the unique challenges involved in recognizing and translating ISL, including variations in sign language among different regions and the need for a large and diverse database of signs and gestures.

   The authors review various approaches for ISL recognition and translation, such as computer vision-based techniques, deep learning-based approaches, and sensor-based approaches. They also discuss the importance of incorporating contextual information, such as facial expressions and body language, in ISL recognition and translation.

   Overall, the paper highlights the importance of ongoing research and development in ISL recognition and translation, particularly in the context of improving accessibility and communication for individuals with hearing impairments in India.

2. **"A Computer Vision-Based Indian Sign Language Recognition System"** by V. B. Patil, S. V. Nimbhorkar, and V. P. Deshmukh. This paper proposes a computer vision-based approach for recognizing and translating Indian Sign Language gestures, using techniques such as background subtraction and contour extraction.

   The paper describes the system architecture, which involves capturing video of the signer's hands using a webcam, preprocessing the video to remove background noise, and extracting hand features using contour-based techniques. The system then recognizes and translates the ISL gestures using support vector machines (SVMs) and a rule-based approach.

   The authors evaluate the performance of the system using a database of 60 ISL gestures and report an average recognition rate of 94.6%. They also compare their approach with other existing approaches and demonstrate that their approach outperforms others in terms of recognition accuracy.

   Overall, the paper demonstrates the feasibility and effectiveness of a computer vision-based approach for recognizing and translating ISL gestures, highlighting its potential for improving communication and accessibility for individuals with hearing impairments in India.

3. **"Real-Time Indian Sign Language Recognition and Conversion into Voice Using Machine Learning Techniques"** by S. S. Kulkarni, P. G. Poonacha, and K. R. Venugopal. This paper presents a real-time Indian Sign Language recognition system that uses machine learning techniques such as convolutional neural networks and long short-term memory networks to convert sign language into spoken language.

   The system captures the sign language gestures using a webcam, preprocesses the video, and then recognizes the gestures using the CNN-LSTM model. The recognized gestures are then converted into spoken language using text-to-speech conversion. The authors evaluate the performance of the system using a dataset of 120 ISL gestures and report an accuracy of 97.5%. Overall, the paper demonstrates the potential of machine learning techniques for real-time recognition and translation of ISL gestures.

4. **"Indian Sign Language Recognition System using Leap Motion Controller"** by S. S. Sathawane and S. S. Patil. This paper proposes a sign language recognition system using the Leap Motion Controller, a device that tracks hand movements and gestures in three dimensions, to recognize and translate Indian Sign Language.

   This paper proposes a deep learning-based approach for recognizing Indian Sign Language (ISL) gestures. The authors use a convolutional neural network (CNN) architecture with transfer learning, which involves fine-tuning a pre-trained CNN model to recognize ISL gestures. The authors evaluate the performance of the system using a dataset of 900 ISL gestures and report an accuracy of 98.89%. The paper highlights the potential of deep learning techniques for improving the accuracy and robustness of ISL recognition systems.

5. **"An Interactive Sign Language Translation System for Indian Sign Language"** by A. Kumar and R. Kumar. This paper presents an interactive sign language translation system for Indian Sign Language that uses a graphical user interface and computer vision techniques to recognize and translate sign language gestures using the Leap Motion controller, a gesture recognition device that uses infrared cameras to track hand movements.

   The authors use a feature extraction technique based on Gaussian Mixture Models (GMMs) and a Hidden Markov Model (HMM) for recognizing the gestures. The authors evaluate the performance of the system using a dataset of 65 ISL gestures and report an accuracy of 91.4%. The paper demonstrates the potential of using non-invasive sensor-based approaches for ISL recognition.

6. **"Automatic Translation of Indian Sign Language to Text Using Neural Networks"** by S. Swathi and K. Balamurugan. This paper proposes a neural network-based approach for translating Indian Sign Language to text, using features such as hand movement, hand orientation, and facial expressions.

   The paper highlights the importance of sign language translation systems for improving accessibility and communication for individuals with hearing impairments, especially in a diverse country like India where multiple sign languages are used. The authors also discuss the challenges involved in developing ISL recognition and translation systems, such as the lack of standardized vocabulary and the variability in handshapes and movements.

   The proposed system uses a combination of convolutional and recurrent neural networks to recognize and translate ISL gestures. The authors compare the performance of their system with that of other approaches such as rule-based systems and template matching-based systems, and demonstrate that their system achieves higher accuracy. The authors also analyze the impact of different factors such as frame rate, input resolution, and choice of neural network architecture on the recognition performance.

Software Requirements

1. Video Capture Software: Software for capturing the video feed from the camera, such as OpenCV, is required. This software should be able to capture high-quality video with a high frame rate, and should be compatible with the camera hardware.

2. Video Processing Software: Software for processing and analyzing the video data is required. This software should be able to detect and track the movement of the hands and fingers accurately. Libraries such as OpenCV, TensorFlow, or PyTorch can be used for this purpose.

3. Machine Learning Frameworks: Machine learning frameworks such as TensorFlow or PyTorch are required to train and test the deep learning models used in the system. These frameworks provide a high-level interface for building and training complex neural networks.

4. Deep Learning Libraries: Deep learning libraries such as Keras or TensorFlow are required to build the deep learning models used in the system. These libraries provide pre-built functions and layers for building and training neural networks.

5. Text-to-Speech Software: Software for converting the translated text into speech, such as the eSpeak library, is required. This software should be able to generate high-quality speech output in the desired language.

6. Operating System: The system should run on a stable and reliable operating system such as Linux or Windows. The operating system should be compatible with all the required software and libraries.

7. Integrated Development Environment (IDE): An IDE such as Visual Studio Code or PyCharm is recommended for development and debugging of the system. These IDEs provide a range of tools for developing, testing, and debugging the code.

8. Database Management System (DBMS): A DBMS such as MySQL or PostgreSQL may be required to manage the dataset and other system data. These systems provide a reliable and scalable way to store and manage large amounts of data.

S/W, H/W, Tools & Technology Requirement

1. Hardware: The system should have a high-performance CPU or GPU, as well as sufficient memory and storage to process and store large amounts of data.

2. Camera: A high-quality camera with a high frame rate is required to capture the hand gestures accurately. It should be able to capture the hand gestures from different angles and in varying lighting conditions.

3. Software: The system should have software for capturing, processing, and analyzing video data. The software should be able to detect and track the movement of the hands and fingers accurately.

4. Dataset: A large and diverse dataset of Indian Sign Language gestures is required for training and testing the system. The dataset should include a variety of gestures performed by different people, with different handshapes, orientations, and movements.

5. Recognition Algorithm: An accurate and efficient recognition algorithm is required to analyze the captured video data and recognize the hand gestures accurately. The algorithm should be able to handle the variability in hand gestures due to factors such as different signing styles and variations in lighting and background.

6. Translation Algorithm: A translation algorithm is required to translate the recognized hand gestures into text or speech. The algorithm should be able to handle the grammatical rules and syntax of Indian Sign Language, and produce accurate and understandable translations.

7. User Interface: The system should have an easy-to-use and intuitive user interface that allows users to interact with the system easily. It should be able to display the recognized gestures and the corresponding translations in real-time. The system should also be able to provide feedback to users on the accuracy of their gestures and translations.

Limitations of Existing System

- Adding new signs is difficult for non - programmers.
- Limited availability of ISL data.
- Grammar for ISL varies.
- Currently the model only detects numbers from 1 to 9.

Economic Feasibility Assessment

1. In terms of market demand, there is a significant need for such a translator in India, as the deaf and hard of hearing population is estimated to be around 18 million. The availability of a reliable and accurate sign language translator could improve accessibility for this population, especially in areas such as education, employment, and healthcare.

2. In terms of the cost of development, it would depend on the complexity of the software and the hardware required. While the initial cost of development may be high, ongoing maintenance costs would be relatively low. Additionally, the development of such a technology could have spinoff benefits for other applications and industries.

3. Regarding potential revenue generation, a sign language translator could be sold as a product or service to institutions and individuals that serve the deaf and hard of hearing population, such as schools, hospitals, and government agencies. Additionally, it could be licensed to technology companies that develop and sell devices such as smartphones and tablets.

4. Overall, the economic feasibility of an Indian Sign Language translator is promising, given the significant market demand and potential revenue streams. However, the exact economic feasibility would depend on factors such as development cost, pricing, and market acceptance.

Implementation

- <u>Main File</u>:

```
#Library install
#pip list

pip install opencv-python

pip install mediapipe

import mediapipe

import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_hands = mp.solutions.hands
from google.protobuf.json_format import MessageToDict

#image to be displayed
overlayList = []
for i in range(10):
        #path =  "ISL_data/" + str(i) + "/0.jpg"
        #img = cv2.imread(path)
    img = cv2.imread(f'ISL_data/{i}/0.jpg')
    overlayList.append(img)
print(overlayList[1].shape)


# For webcam input:
cap = cv2.VideoCapture(0)
with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      # If loading a video, use 'break' instead of 'continue'.
```

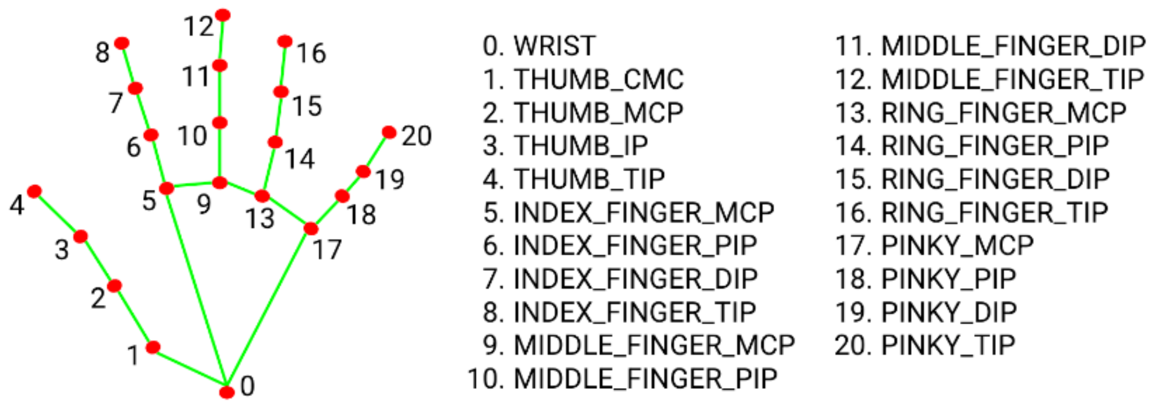
```
      continue

    # To improve performance, optionally mark the image as not writeable to
    # pass by reference.
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.flip(image, 1)
    h, w, c = image.shape
    results = hands.process(image)
    fingertips = [8,12,16,20]
    #start editing and analizing the image.
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    if results.multi_hand_landmarks:
      for hand_num, hand_landmarks in
enumerate(results.multi_hand_landmarks): #by default 2 hand dectection i.e
hand_num = 0/1
        position = {}   #clear/reset for next hand
        for id, lm in enumerate(hand_landmarks.landmark):
          x = int(lm.x*w)
          y = int(lm.y*h)
          position[id] = (x, y)

          #storing new hand cordinates

          #print(id," : ",position[id])

          #if id in fingertips:
          #    cv2.circle(image, (x,y), 15, (255, 0, 0),cv2.FILLED)



          #check finger status (flags)
          thumb_open = False
          index_open = False
          middle_open = False
          ring_open = False
          pinky_open = False

          #if(position[8][0]>position[8-3][0]):
          text = "hand detectied"
          font = cv2.FONT_HERSHEY_SIMPLEX
          org = (50, 50)
```

```
            fontScale = 1
            color = (255, 0, 0)
            thickness = 2
            image = cv2.putText(image, text , org, font, fontScale, color,
thickness, cv2.LINE_AA)

         #Right and left hand detection
         Lable = results.multi_handedness[0].classification[0].label



         for tip in fingertips:
            if(position[tip][1] < position[tip-1][1]):
               image = cv2.putText(image, "finger_open" ,
(position[tip][0],position[tip][1]), font, fontScale, color, thickness,
cv2.LINE_AA)
               if(tip == 8):index_open = True
               if(tip == 12):middle_open = True
               if(tip == 16):ring_open = True
               if(tip == 20):pinky_open = True

         if(Lable == "Right"):
            if(position[4][0] < position[5][0]):
               image = cv2.putText(image, "Right_thumb_open" ,
(position[4][0],position[4][1]), font, fontScale, color, thickness, cv2.LINE_AA)
               thumb_open = True
         if(Lable == "Left"):
            if(position[4][0] > position[5][0]):
               image = cv2.putText(image, "Left_thumb_open" ,
(position[4][0],position[4][1]), font, fontScale, color, thickness, cv2.LINE_AA)
               thumb_open = True

         #print finger status (flags)
            print(thumb_open, index_open, middle_open, ring_open, pinky_open,
'\n')

         #Sign Dection
         #1
         if(index_open == True and middle_open == False and ring_open ==
False and pinky_open == False):
            image = cv2.putText(image, "1" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
         #2
```

```
        if(index_open == True and middle_open == True and ring_open ==
False and pinky_open == False and thumb_open == False):
            image = cv2.putText(image, "2" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #3
        if(index_open == True and middle_open == True and ring_open == True
and pinky_open == False and thumb_open == False):
            image = cv2.putText(image, "3" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #4
        if(index_open == True and middle_open == True and ring_open == True
and pinky_open == True and thumb_open == False):
            image = cv2.putText(image, "4" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #5
        if(index_open == True and middle_open == True and ring_open == True
and pinky_open == True and thumb_open == True):
            image = cv2.putText(image, "5" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #6
        if(index_open == False and middle_open == False and ring_open ==
False and pinky_open == True and thumb_open == False):
            image = cv2.putText(image, "6" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #7
        if(position[8][1] > position[7][1] and position[8][1] < position[5][1] and
index_open == False and middle_open == False and ring_open == False and
pinky_open == False):
            image = cv2.putText(image, "7" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #8
        if(index_open == True and middle_open == True and ring_open ==
False and pinky_open == False and thumb_open == True):
            image = cv2.putText(image, "8" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)
        #9
        if(position[4][1] > position[1][1] and position[19][1] < position[17][1]
and index_open == False and middle_open == False and ring_open == False
and pinky_open == False and thumb_open == True):
            image = cv2.putText(image, "9" , (50,100), font, fontScale, color,
thickness, cv2.LINE_AA)

        # Draw the hand annotations on the image.
```

```
            mp_drawing.draw_landmarks(
                image,
                hand_landmarks,
                mp_hands.HAND_CONNECTIONS,
                mp_drawing_styles.get_default_hand_landmarks_style(),
                mp_drawing_styles.get_default_hand_connections_style()
            )

    #to display image
    #h, w, c = overlayList[1].shape
    #image[0:h, 0:w] = overlayList[1]

    # Flip the image horizontally for a selfie-view display.
    cv2.imshow('MediaPipe Hands', image)
    if cv2.waitKey(5) & 0xFF == 27:
      break
cap.release()
cv2.destroyAllWindows()
```

```
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
True False False False False
True False False False False
False False False False False
True False False False False
False False False False False
True False False False False
True False False False False
True False False False False
True False False False False
```



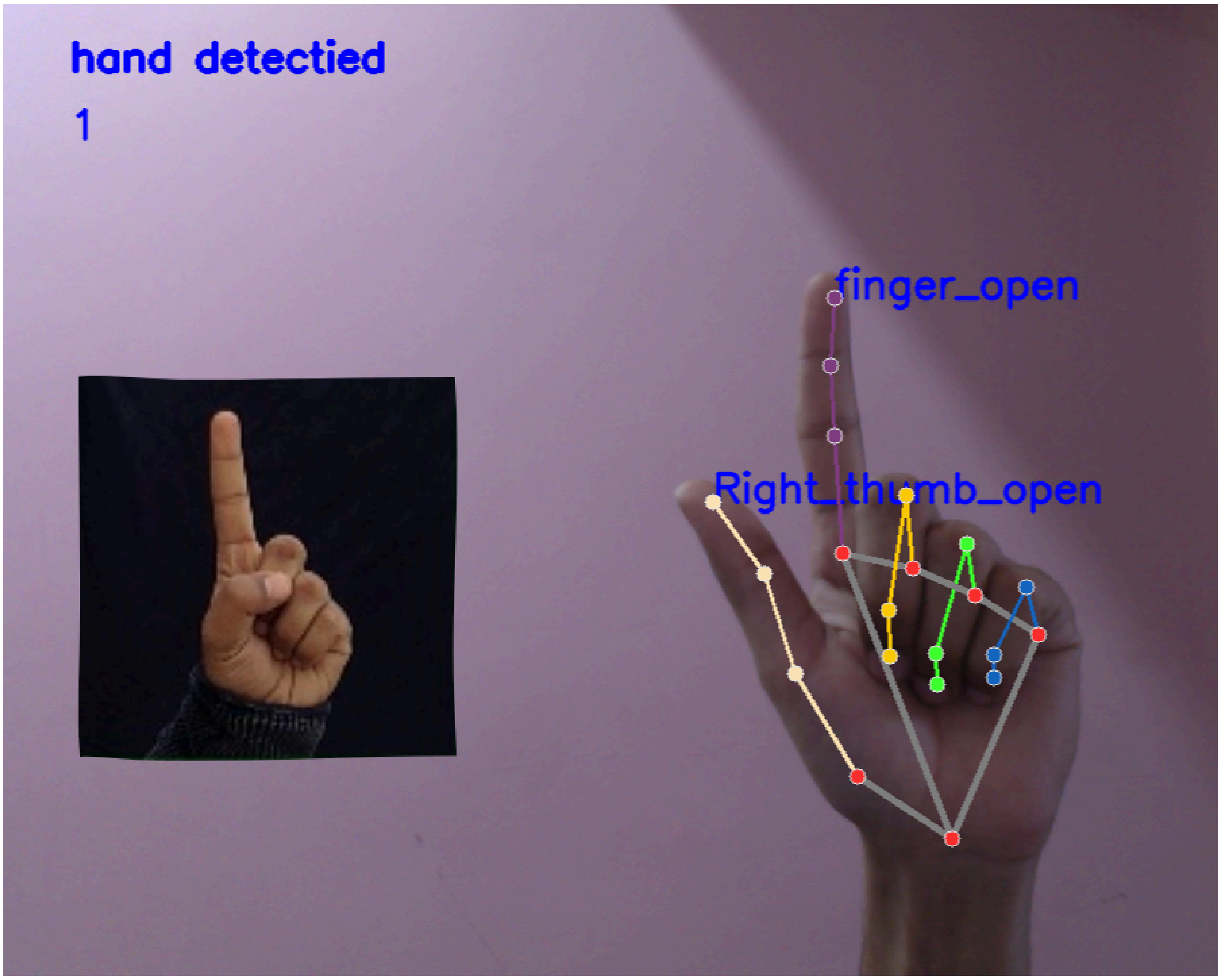
results

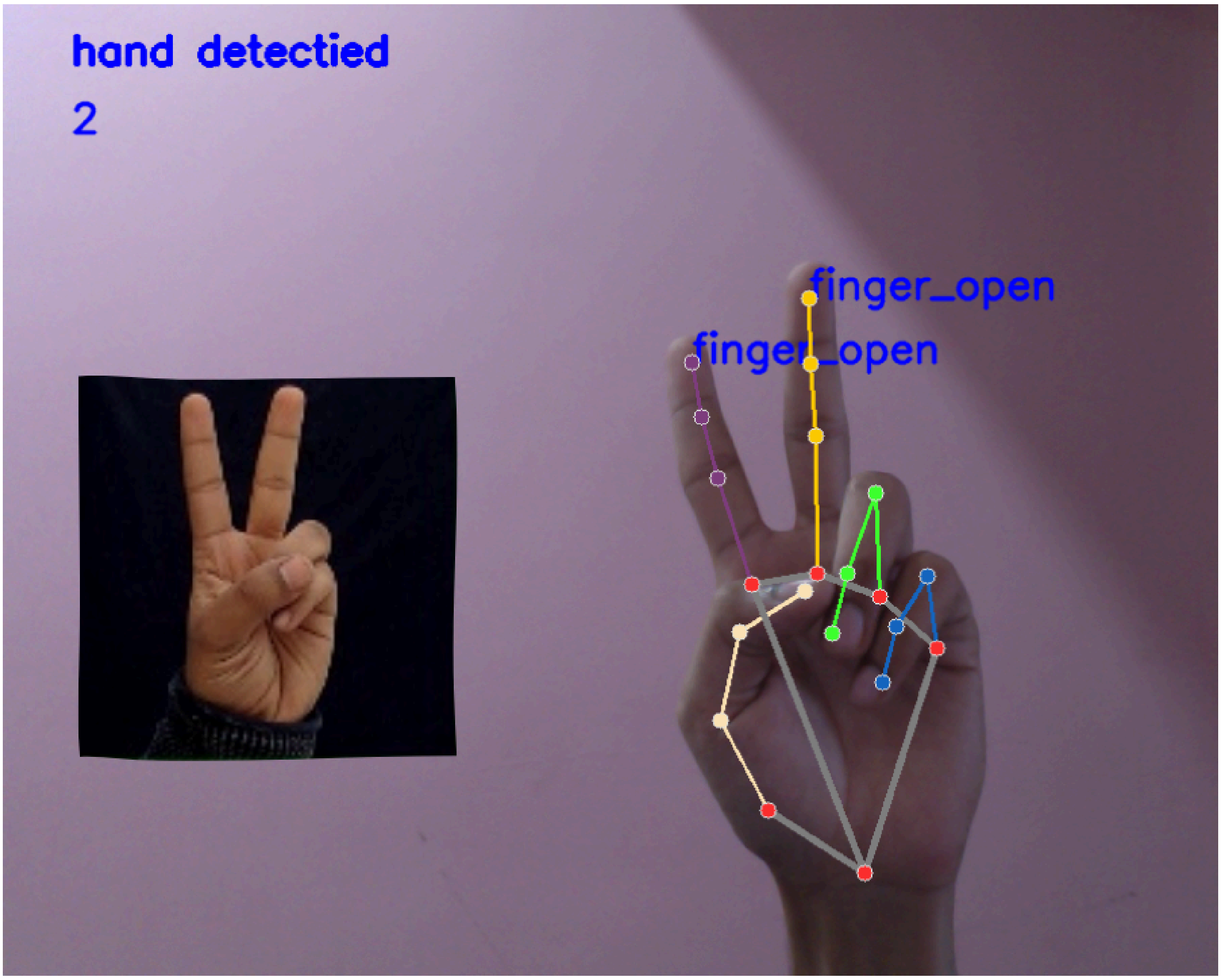
mp_hands.HandLandmark.WRIST

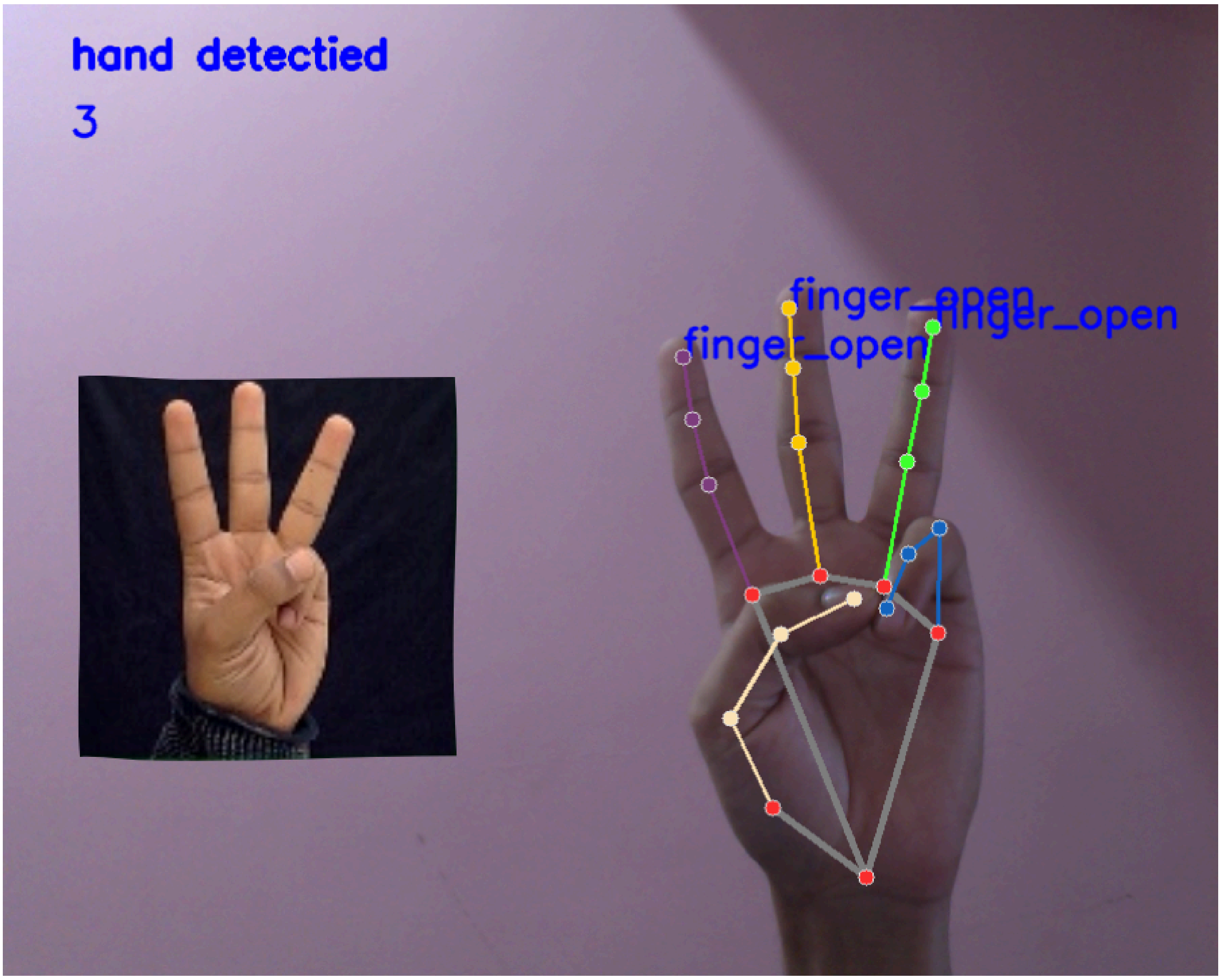
To find the position of a handmark

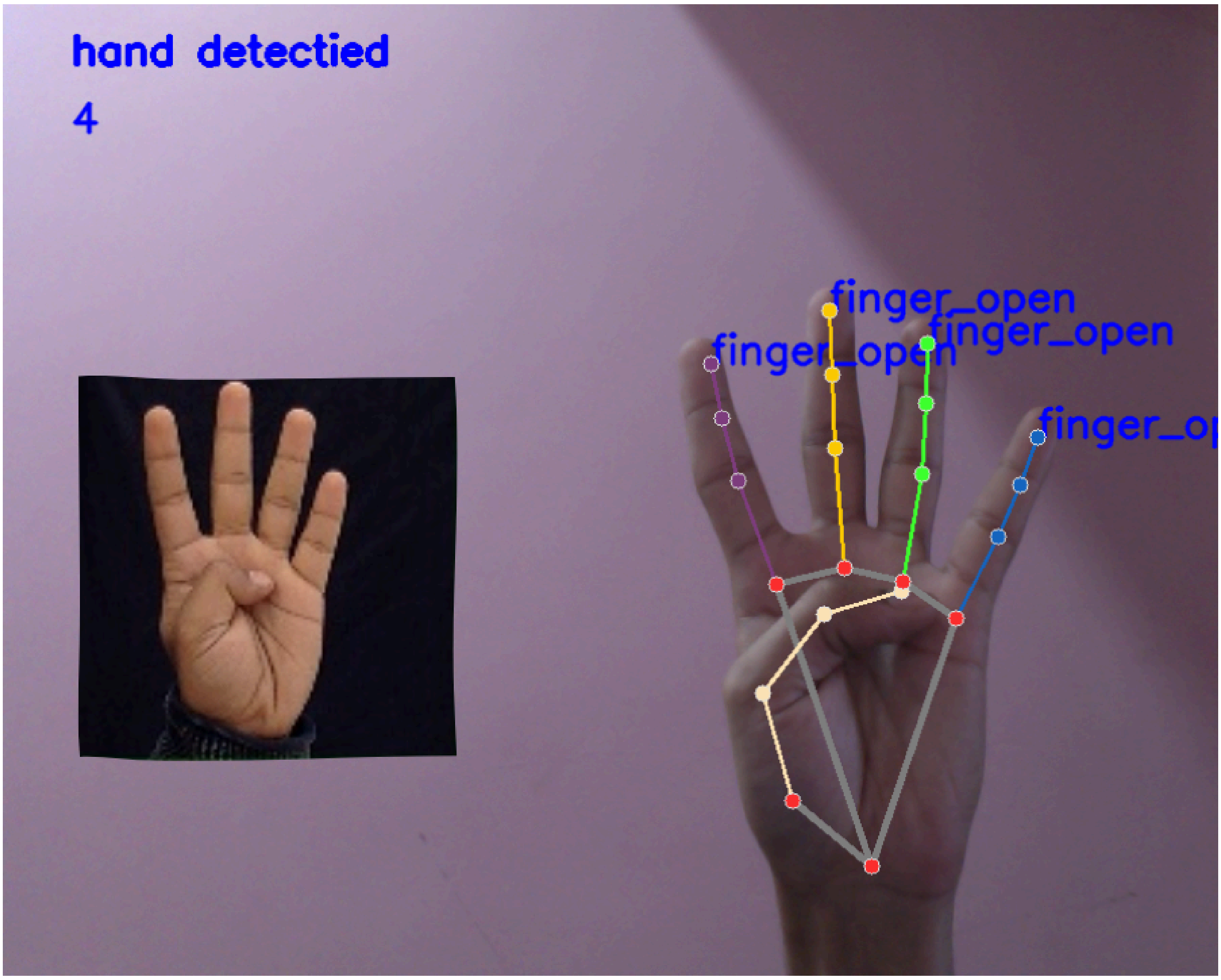
```
#to find the position of a hand_landmark
print(position[8])

print(position[6])

#to dectect left and right hand
results.multi_handedness

results.multi_hand_landmarks[0].landmark[0]

a = []
a+=(2,3)
a+=(3,0)
a

if (position[8][0] < position[8-3][0]):
   print("true")

lm

type(lm)

results.multi_handedness[0].classification[0].label
```

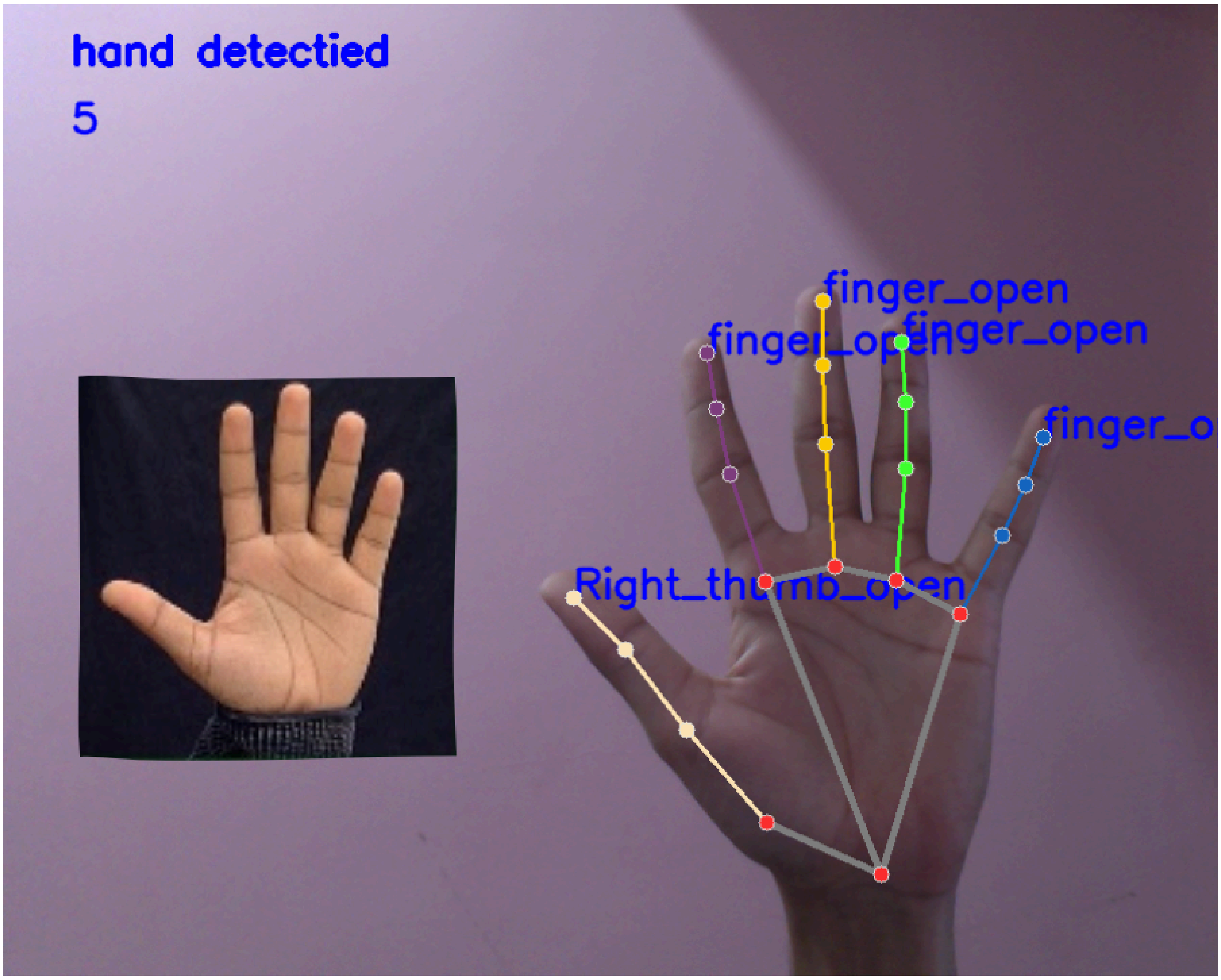
33Outputs

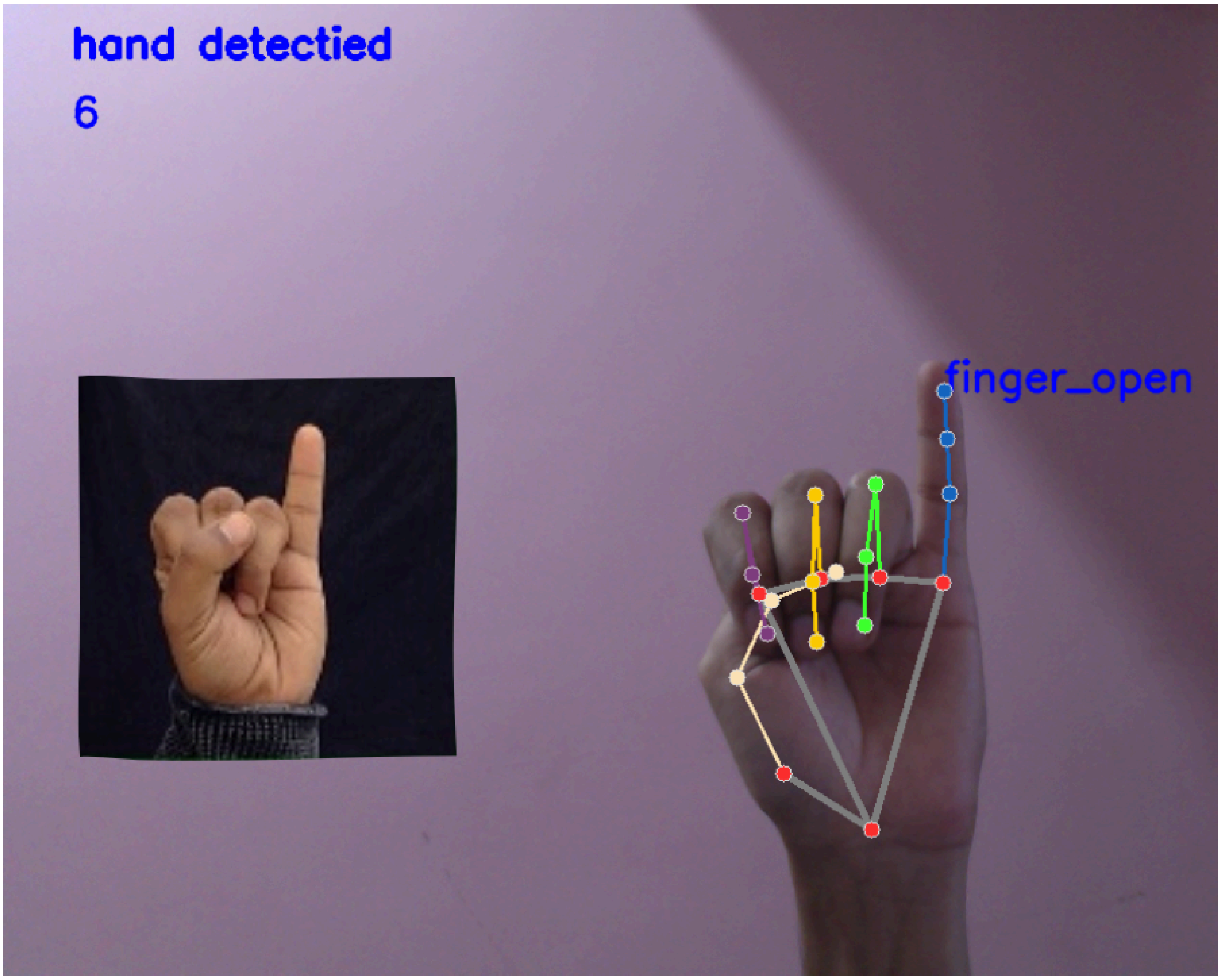


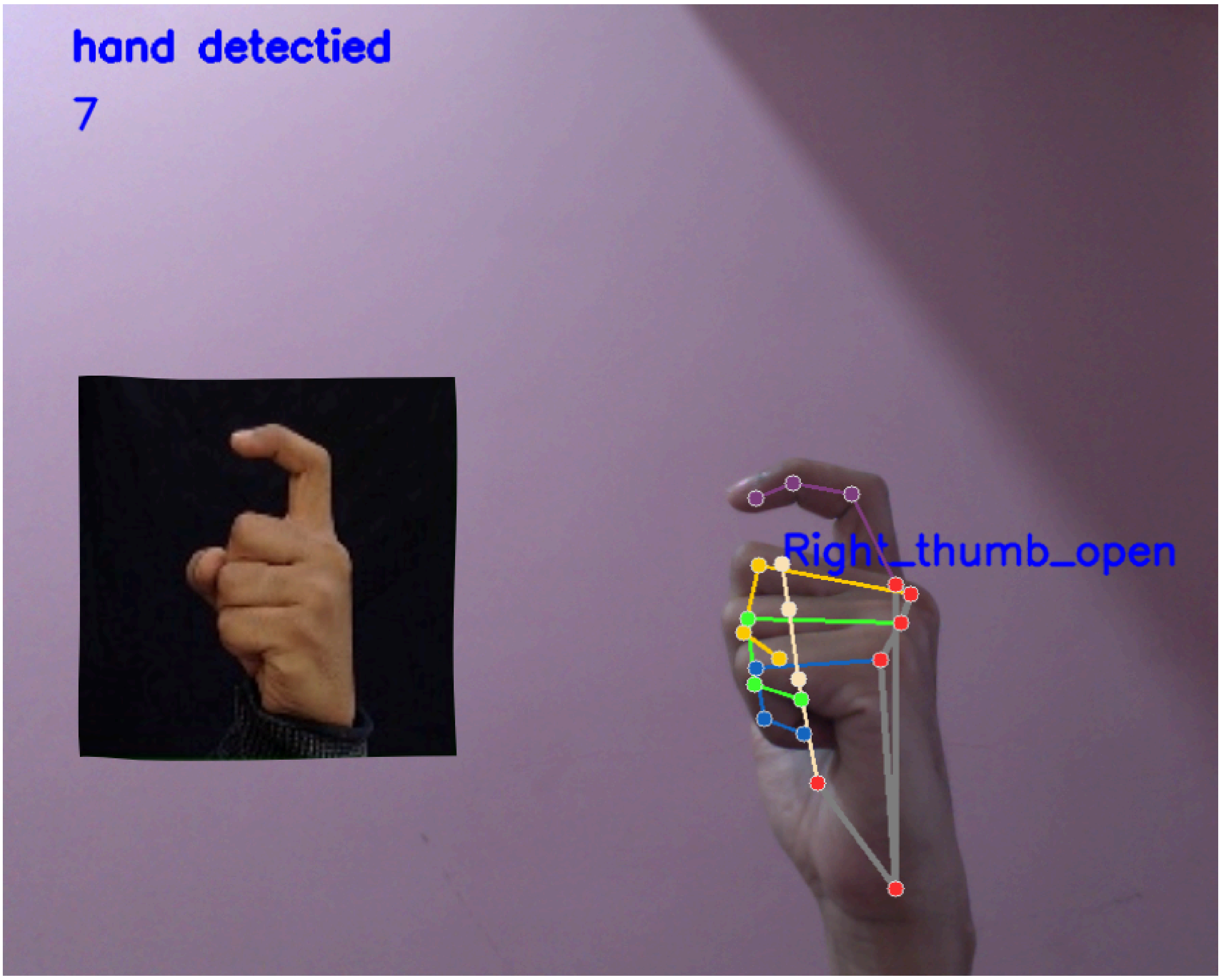

hand detectied
2
finger_open
finger_open

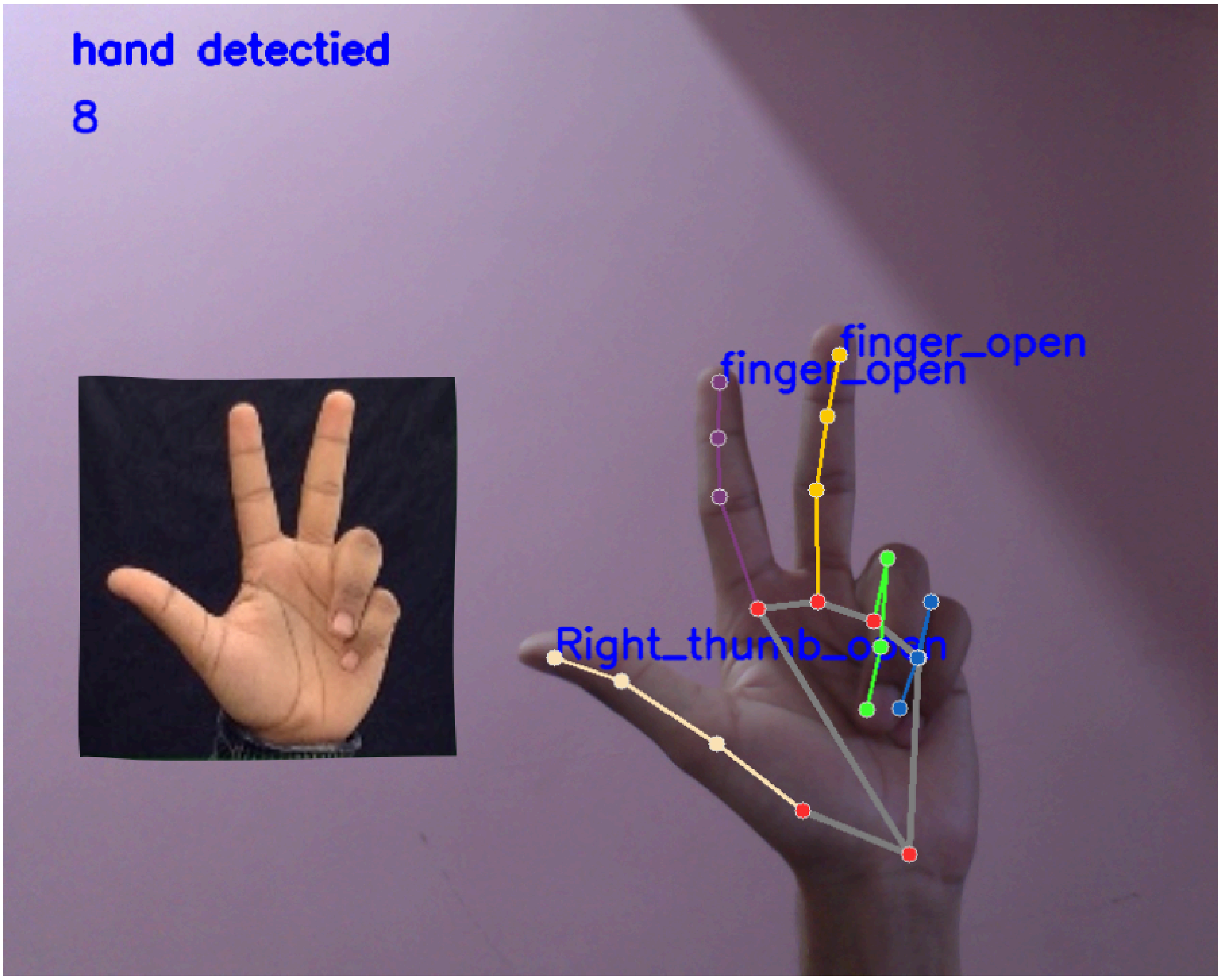

hand detectied
3
finger_open
finger_open
finger_open

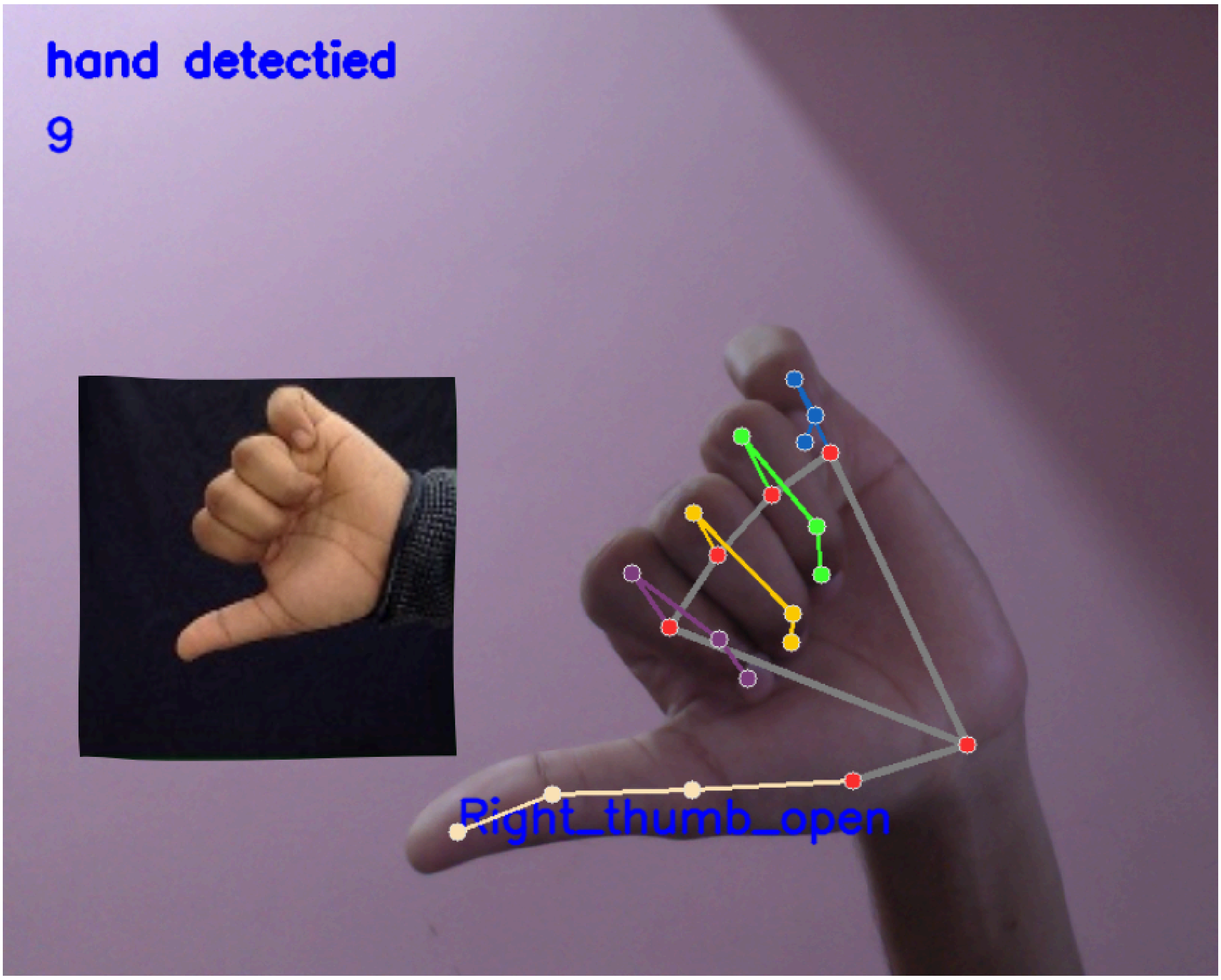

hand detectied
4
finger_open
finger_open
finger_open
finger_o


hand detectied
5
finger_open
finger_open
finger_open
finger_o
Right_thumb_open


hand detected
6
finger_open


hand detectied
7
Right_thumb_open


hand detectied
8
finger_open
finger_open
Right_thumb_open


hand detectied
9
Right_thumb_open

Conclusions and future Enhancements

1. In conclusion, the development of an Indian Sign Language translator has the potential to significantly improve accessibility and communication for the deaf and hard of hearing population in India. With recent advancements in machine learning and computer vision technology, it is now possible to develop accurate and reliable sign language recognition systems that can translate sign language to text or speech.

2. However, there is still much work to be done to improve the accuracy and usability of these systems. Future enhancements to an Indian Sign Language translator could include expanding the vocabulary and grammar it can recognize, improving its real-time performance, and integrating it with other assistive technologies such as text-to-speech or speech-to-text systems.

3. Additionally, there is a need for more research and development in the area of Indian Sign Language recognition and translation, as there are variations in sign language across different regions and communities in India. A more comprehensive understanding of these variations could help improve the accuracy and relevance of a sign language translator.

4. In conclusion, the development of an Indian Sign Language translator is a promising and much-needed application of machine learning and computer vision technology. With further research and development, it has the potential to significantly improve the quality of life and accessibility for the deaf and hard of hearing population in India.

5. Furthermore, another potential future enhancement for an Indian Sign Language translator is to incorporate natural language processing techniques to improve its translation accuracy. This would involve analyzing the context and meaning of the sign language gestures to produce more accurate and meaningful translations.

6. Moreover, integrating a feedback system into the translator could help improve its accuracy over time. For example, if a user spots a translation error, they could provide feedback which would be incorporated into the system's training data to improve future translations.

7. Another area for improvement is the availability of data for training and testing sign language recognition models. Collecting a large and diverse dataset of sign language gestures can help improve the accuracy of the models, and this could be achieved through crowdsourcing or collaborations with sign language schools and communities.

8. In terms of economic feasibility, the availability of low-cost hardware such as mobile phones and tablets could make a sign language translator more accessible and affordable to a wider audience. Additionally, partnerships with NGOs or government agencies could help increase the reach and adoption of the technology among the deaf and hard of hearing population in India.

9. In conclusion, there is a significant potential for the development of an Indian Sign Language translator with further research and development. The application of machine learning and computer vision technology in this domain could have a significant positive impact on the accessibility and quality of life for the deaf and hard of hearing population in India.

References

- R. Mishra, A. Ekbal, and G. Saha, “A review on automatic sign language recognition systems,” International Journal of Speech Technology, vol. 22, no. 3, pp. 439–452, 2019.

- S. Swathi and K. Balamurugan, “Automatic Translation of Indian Sign Language to Text Using Neural Networks,” in 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 373–377.

- D. Meena and V. Bhatnagar, “Indian Sign Language Recognition using Deep Learning: A Survey,” in 2019 9th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 2019, pp. 272–278.

- S. Singh and S. Kumar, “Indian Sign Language Recognition: A Review,” in 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1–6.

- S. Verma, A. Kumar, and R. Kumar, “Real-time Indian Sign Language Recognition using Deep Learning Techniques,” in 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 116–120.

- N. Sharma, V. Jain, and D. Srivastava, “Sign Language Recognition using Computer Vision: A Review,” in 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 530–535.
- S. Mishra and R. Tripathi, “A Review on Indian Sign Language Recognition: Issues and Challenges,” in 2020 6th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 200–205.

- S. Pal and S. S. Tuli, “Deep Learning Techniques for Indian Sign Language Recognition: A Review,” in 2019 IEEE International Conference on Signal Processing, Communication and Networking (ICSCN), 2019, pp. 1–5.

- A. Tiwari and D. K. Singh, “Sign Language Recognition using Image Processing and Machine Learning: A Review,” in 2019 International Conference on Computer Communication and Informatics (ICCCI), 2019, pp. 1–5.

- P. Kumar, P. Kumar, and A. Kumar, “Indian Sign Language Recognition using Deep Learning: A Review,” in 2021 2nd International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1057–1061.

- K. Jain and S. Kumar, “Indian Sign Language Recognition using Computer Vision: A Review,” in 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 391–395.

- M. Singh and R. Bhatia, “A Survey on Indian Sign Language Recognition,” in 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), 2021, pp. 12–17.