SIGN LANGUAGE RECOGNITION

MAJOR PROJECT REPORT

Submitted by

AREEBA SABRI
2020-310-032

in partial fulfillment for the award of the degree of

B.TECH COMPUTER SCIENCE AND ENGINEERING

Under the supervision of

DR. IHTIRAM RAZA KHAN

Department of Computer Science & Engineering
School of Engineering Sciences & Technology

JAMIA HAMDARD

(Deemed to be University)
New Delhi-110062

2024

DECLARATION

I, **Ms. Areeba Sabri** a student of **Bachelors Of Computer Science And Technology (B.Tech CSE) , Enrolment No:2020-310-032,** hereby declare that the Project/Dissertation entitled **“ Sign Language Recognition ”** which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **Bachelors Of Computer Science And Technology (B.Tech CSE)**,is my original work and it has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Areeba Sabri

(Signature and Name of the Applicant)

Date: 22 March 2024

Place: New Delhi

ACKNOWLEDGEMENT

I express my sincere thanks to **Dr Ihtiram Raza Khan** (Dept. of Computer Science and Engineering), my project in charge, who guided me through the project for his valuable suggestions and guidance for completing the project. This project has been a success only because of his guidance.

I deeply express my sincere thanks to our Head of Department **Dr. Farheen Siddiqui** for encouraging me and allowing me to present the project on the topic "**Sign Language Recognition**" at the department premises for the partial fulfilment of the requirements leading to the award of B.Tech degree.

I am also thankful to the whole computer science and engineering department for providing the technical support to carry out the project work, letting us utilize all the necessary facilities of the institute and providing guidance at each & every step during the project work.

INDEX

OBJECTIVE

A python application that uses camera input to train a Random Forest Classifier to predict a specific sign that classifies different gestures on the basis of what is being shown to the camera into separate classes.

INTRODUCTION

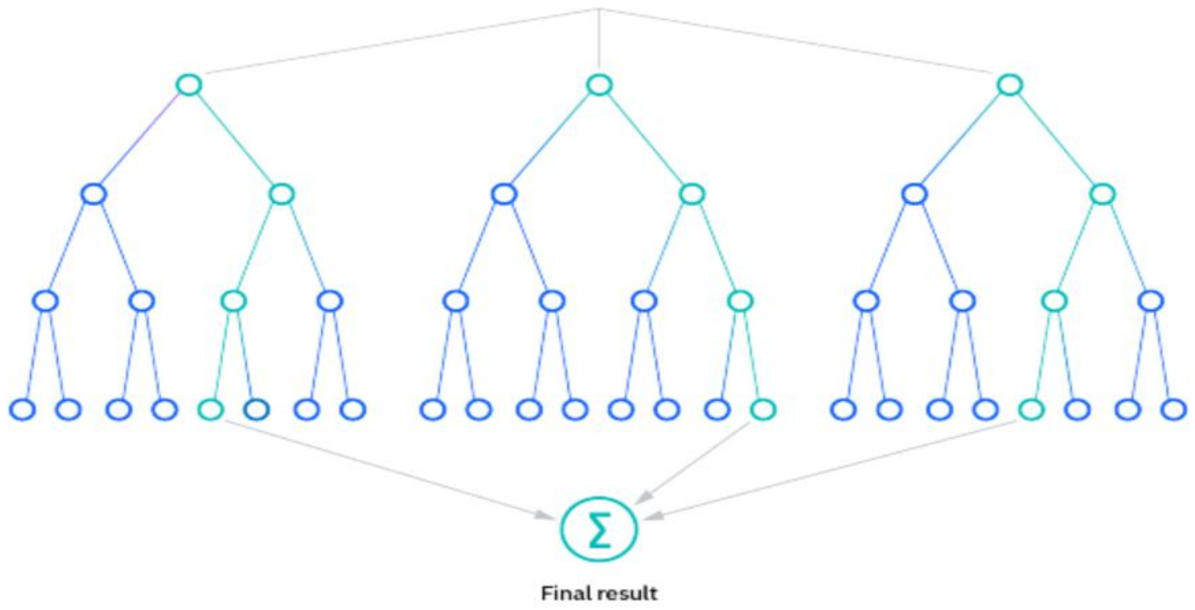
Sign language is as a fundamental mode of communication for thousands of people worldwide who are hard of hearing. Unlike spoken languages, sign languages rely on hand signs, face expressions, and body movements to communicate and express thoughts and emotions. Despite its significance, there exists a pervasive communication barrier for individuals who are not proficient in sign language, hindering effective interaction and understanding between individuals with and without hearing impairments. This barrier can lead to social isolation, limited educational opportunities, and restricted access to employment and healthcare services for people who rely on sign language for communication.

In recent years due to the advancements in technology, particularly in the fields of computer vision, machine learning and artificial intelligence, have opened new avenues for addressing the communication challenges faced by individuals with hearing impairments. Sign language detection systems, powered by sophisticated algorithms and deep learning models, have emerged as promising solutions for real-time interpretation and translation of sign language gestures into spoken or written language. These systems analyze video/image data captured through cameras, recognize hand movements and gestures, and convert them into text, enabling individuals with hearing impairments to convey messages effectively with those who don’t understand sign gestures.

The development of sign language recognition technology holds immense potential for better communication accessibility for people with hearing disorders. By providing real-time interpretation and translation of sign language, these systems empower individuals to participate more actively in social interactions, educational activities, and professional settings. They facilitate better communication between individuals with and without hearing impairments, fostering inclusivity and equal opportunities for all members of society. Moreover, sign language detection technology has the potential to improve access to essential services such as healthcare and emergency assistance for individuals who rely on sign language for communication.

RANDOM FOREST CLASSIFIER FOR SIGN LANGUAGE DETECTION

The Random Forest Classifier is a supervised learning algorithm that is a member of ensemble learning methods. The idea behind it is constructing multiple decision trees in the training phase and outputting the mode of the classes or the mean prediction of individual trees. Each decision tree in the forest is trained on a random subset of the training data and at each node, the algorithm selects the best split among a subset of features that results in a diverse set of trees that helps to form the forest. This diversity helps reduce overfitting thus helps to improves the overall performance of the classifier.



In the accordance of sign language detection, the Random Forest Classifier is to be trained by a dataset consisting of hand gesture images or video frames labelled with corresponding sign language gestures. During the training phase, the algorithm learns to extract important features from the input data such as the position and movement of the hands, as well as the shape and configuration of the fingers and palm. These features serve as discriminative cues that enable the classifier to differentiate between different sign language gestures.

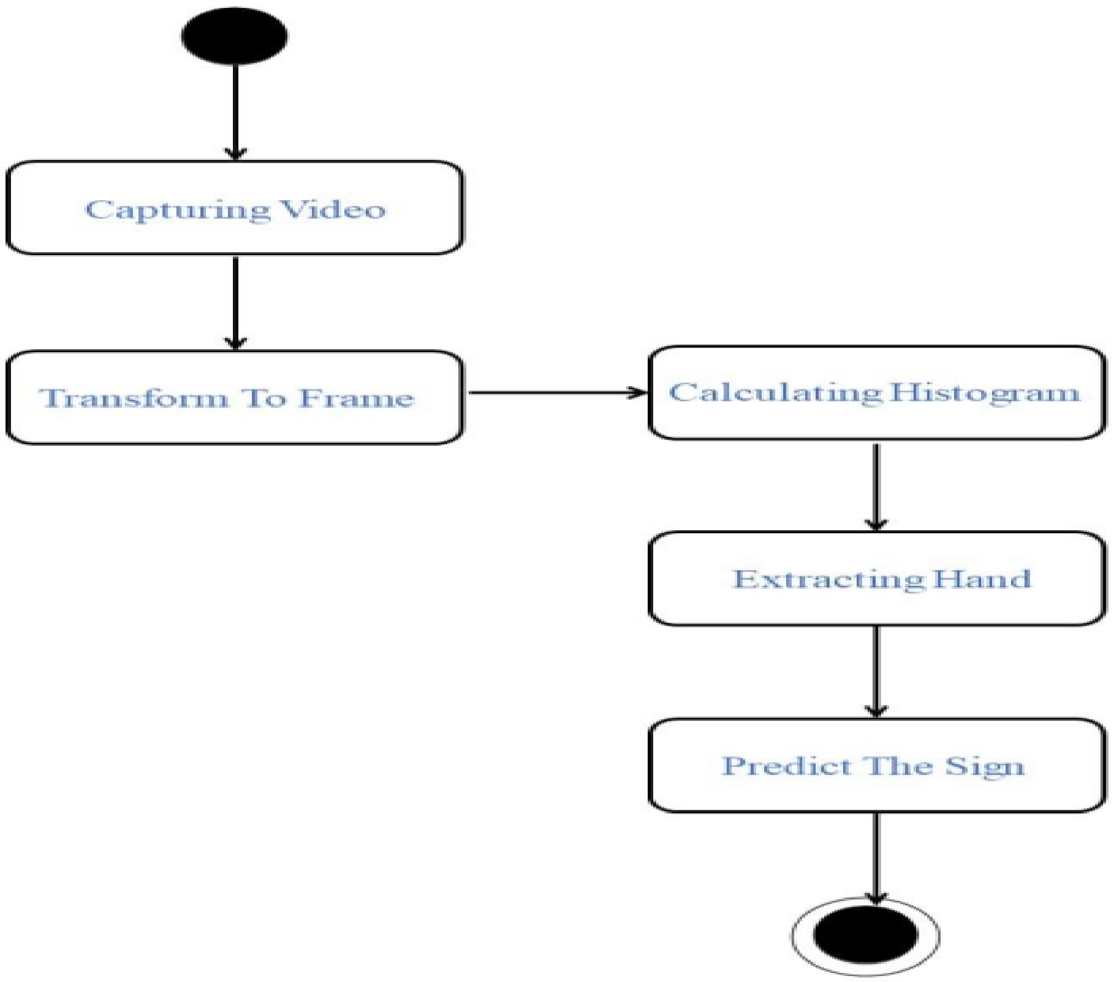
Once trained, the Random Forest Classifier can be deployed to recognize sign language gestures in real-time. Given a new input image, the classifier processes the data and predicts the corresponding sign language gesture based on the learned decision boundaries of the trees that are generated in the forest. This prediction can be translated into text or spoken language, allowing individuals without knowledge of sign language to understand and respond to the communicated message effectively.

One of the key advantages of using the Random Forest Classifier for sign language detection is its ability to handle multi-dimensional data and non-linear similarities between features.

- Robustness: Random forests are known for their robustness against noise and outliers in the data. This makes them well-suited for sign language detection, where hand movements may vary in speed, orientation, and lighting conditions.
- High Accuracy: Random forests are capable of achieving high levels of accuracy in classification tasks, especially when trained on large and diverse datasets. This is advantageous for sign language detection, where precise recognition of hand gestures is essential for effective communication.
- Handling High-Dimensional Data: Sign language detection often involves processing high-dimensional data, such as images or video frames containing multiple features related to hand movements. Random forests can effectively handle such data, making them suitable for complex gesture recognition tasks.
- Scalability: Random forests can be parallelized and distributed across multiple processors or computing nodes, enabling efficient processing of large datasets and real-time inference in applications such as sign language detection systems deployed on mobile devices or embedded systems.

PROBLEM STATEMENT

To develop a real-time sign language detection model to accurately interpret and translate the sign gestures into text, facilitating seamless communication for individuals with hearing impairments



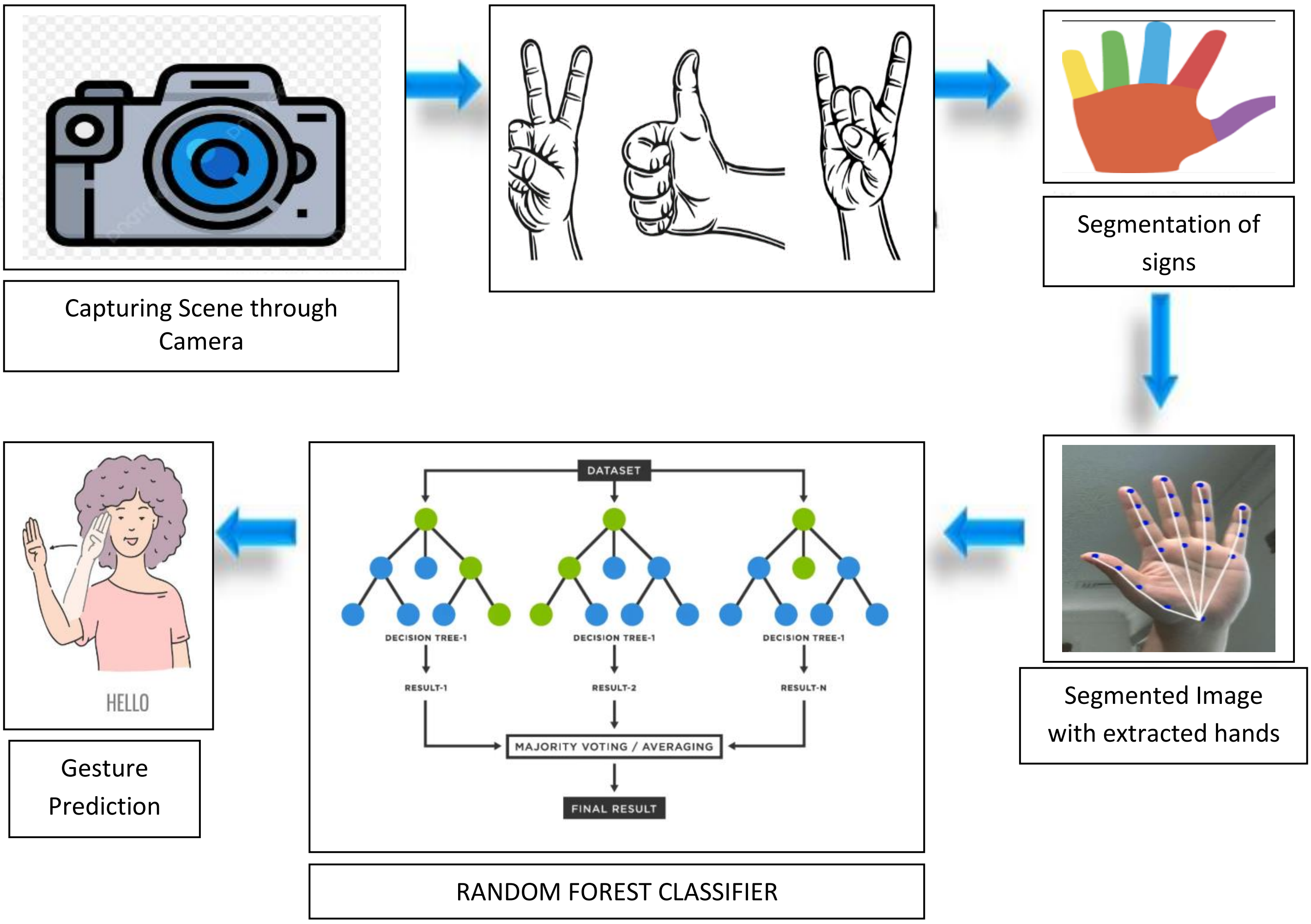
SOFTWARE REQUIREMENTS AND SPECIFICATIONS

VS Code: An open-source code editor by Microsoft, known for its lightweight design, rich features, and extensive extension ecosystem, providing a customizable and efficient development environment.

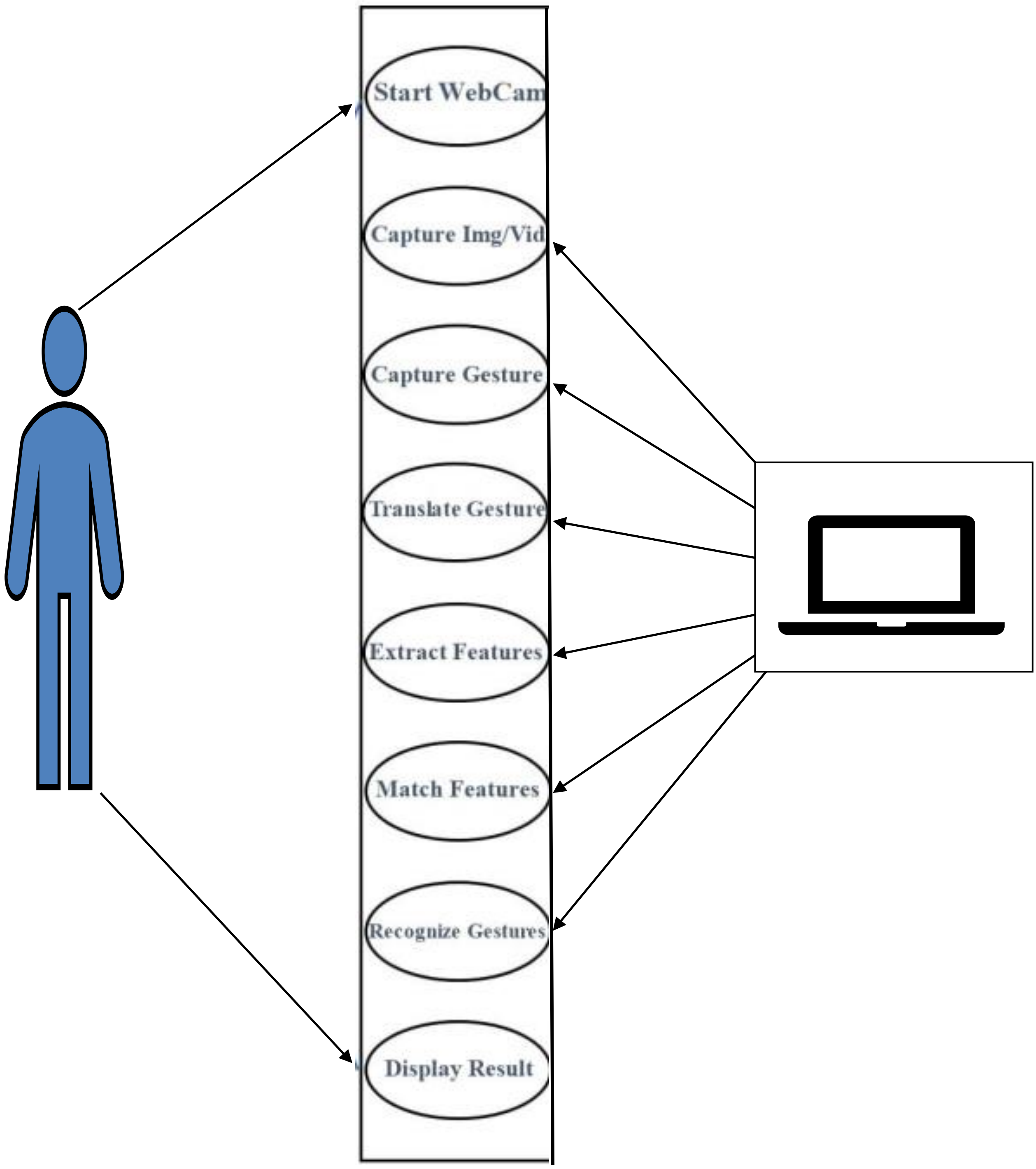
Python: An interpreted, high-level programming language with a clean syntax and strong community support, widely used for various applications.

- OpenCV: An open-source computer vision library offering algorithms for image and video processing tasks like object detection, facial recognition, and image enhancement, enabling developers to build powerful computer vision applications.
- OS: A library in Python providing functions for interacting with the OS, allowing tasks for file manipulation, directory handling and environment variable management facilitating platform-independent programming.
- Scikit-learn (sklearn): offers tools for data preprocessing, model selection, training, evaluation, and deployment, enabling developers to build and deploy machine learning models with ease.
- Pickle: A module in Python used for serializing and deserializing Python objects, allowing objects to be saved to disk in a compact binary format and later restored, commonly used for model persistence and data storage in machine learning applications.
- NumPy: NumPy is a critical Python package for scientific computing, serving as the foundation for many numerical computing activities. It enables multidimensional arrays, mathematical functions, linear algebra operations, and the production of random numbers.
- Mediapipe: A Google-developed library for building machine learning pipelines to process perceptual data like images and video, with pre-built components for tasks like hand tracking, pose estimation, and face detection, enabling developers to create real-time computer vision applications.
- Matplotlib: A comprehensive plotting library for creating static, animated, and interactive visualizations in Python, offering a wide range of plotting functions to generate publication-quality graphs, charts, and plots with customizable features, facilitating data exploration and presentation.

SYSTEM ARCHITECTURE DIAGRAM

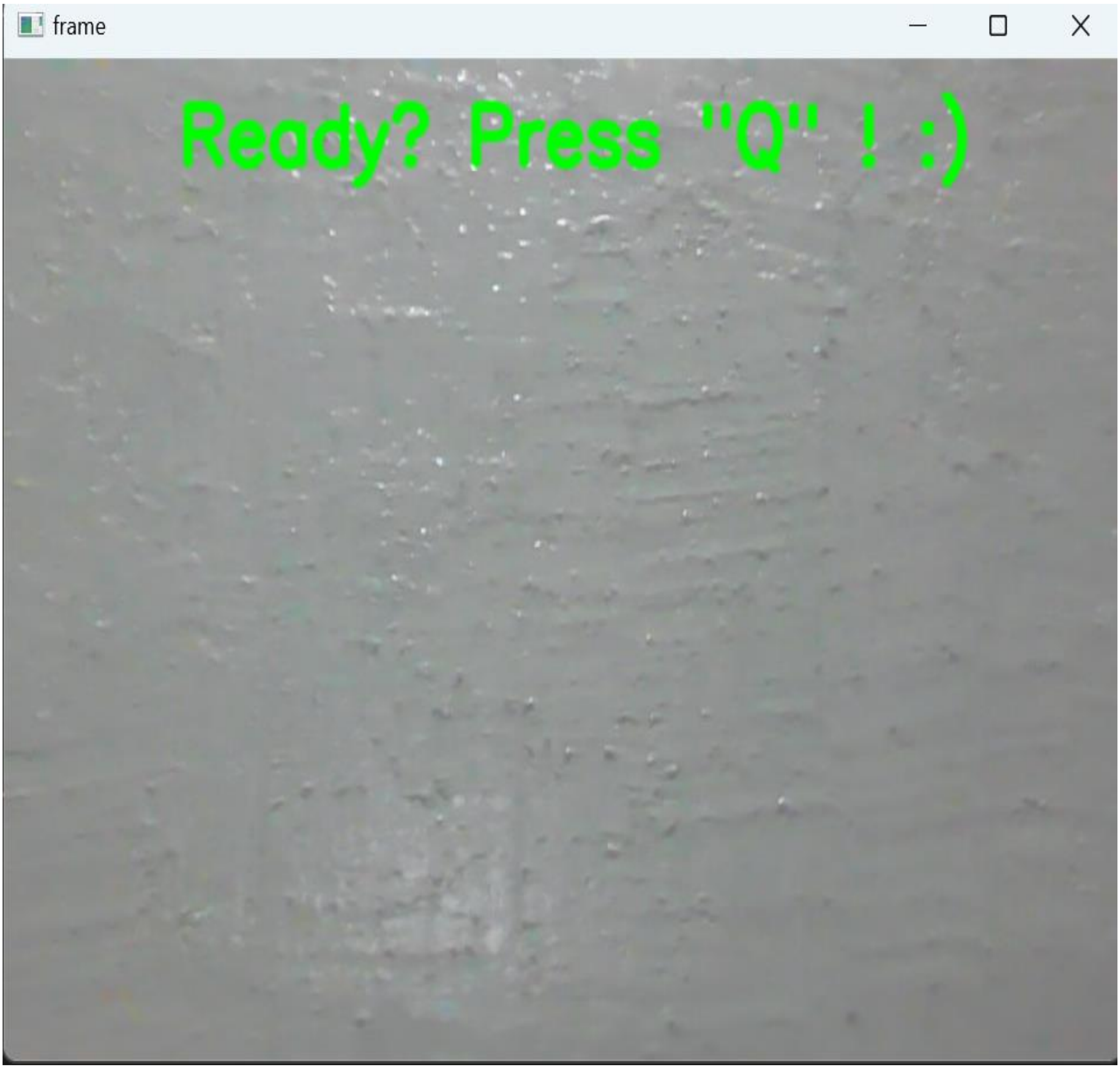


USE CASE DIAGRAM



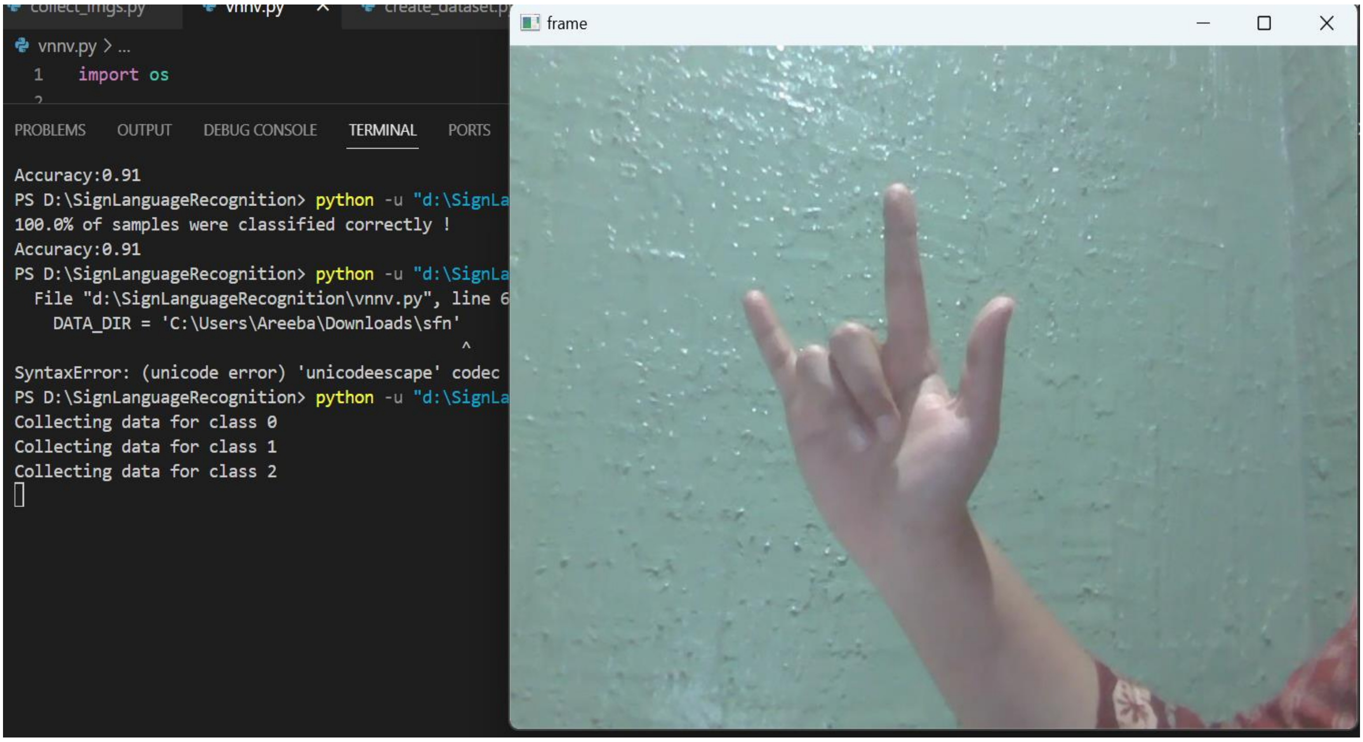
SNAPSHOTS OF DIFFERENT INPUT AND OUTPUT SCREENS

UI of the application



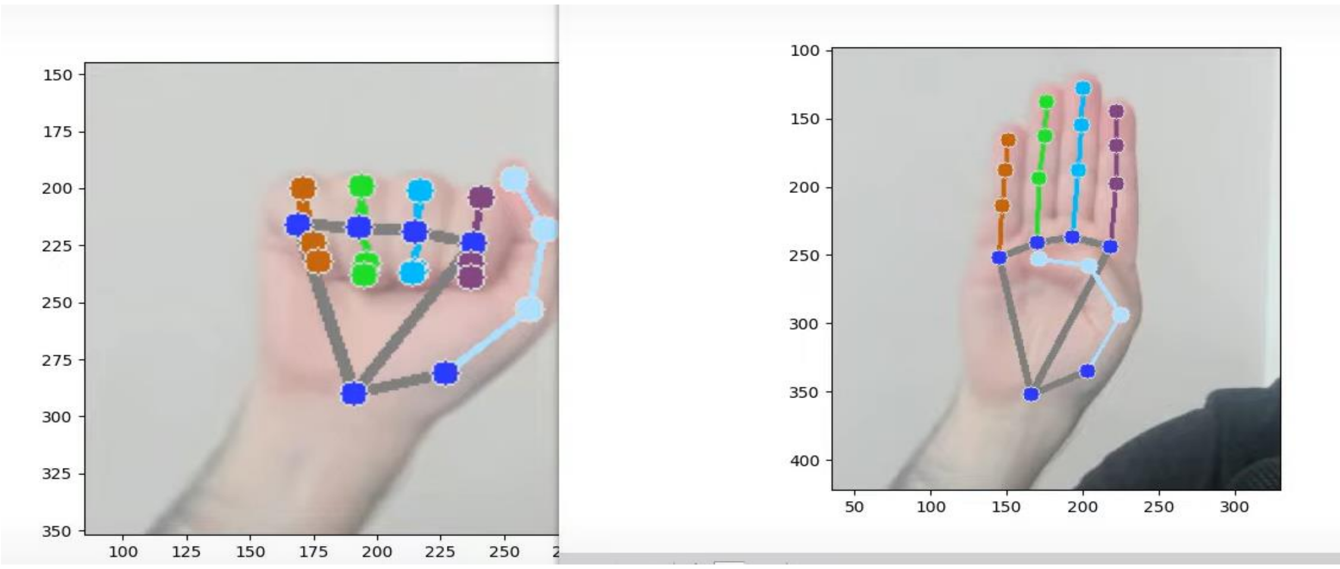
TAKING INPUT AND CREATING DIFFERENT CLASSES

```python
number_of_classes = 5
dataset_size = 100

cap = cv2.VideoCapture(0)
for j in range(number_of_classes):
    if not os.path.exists(os.path.join(DATA_DIR, str(j))):
        os.makedirs(os.path.join(DATA_DIR, str(j)))

    print('Collecting data for class {}'.format(j))

    done = False
    while True:
        ret, frame = cap.read()
        cv2.putText(frame, 'Ready? Press "Q" ! :)', (100, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 255, 0), 3,
                    cv2.LINE_AA)
        cv2.imshow('frame', frame)
        if cv2.waitKey(25) == ord('q'):
            break

    counter = 0
```




```
PS D:\SignLanguageRecognition> python -u "d:\SignLanguageRecognition\vnnv.py"
Collecting data for class 0
Collecting data for class 1
Collecting data for class 2
Collecting data for class 3
Collecting data for class 4
```

CREATING DATASET
(EXCTRACTION OF HAND LANDMARKS AND PREPROCESSING THE DATA)



TRAINING THE DATA SET

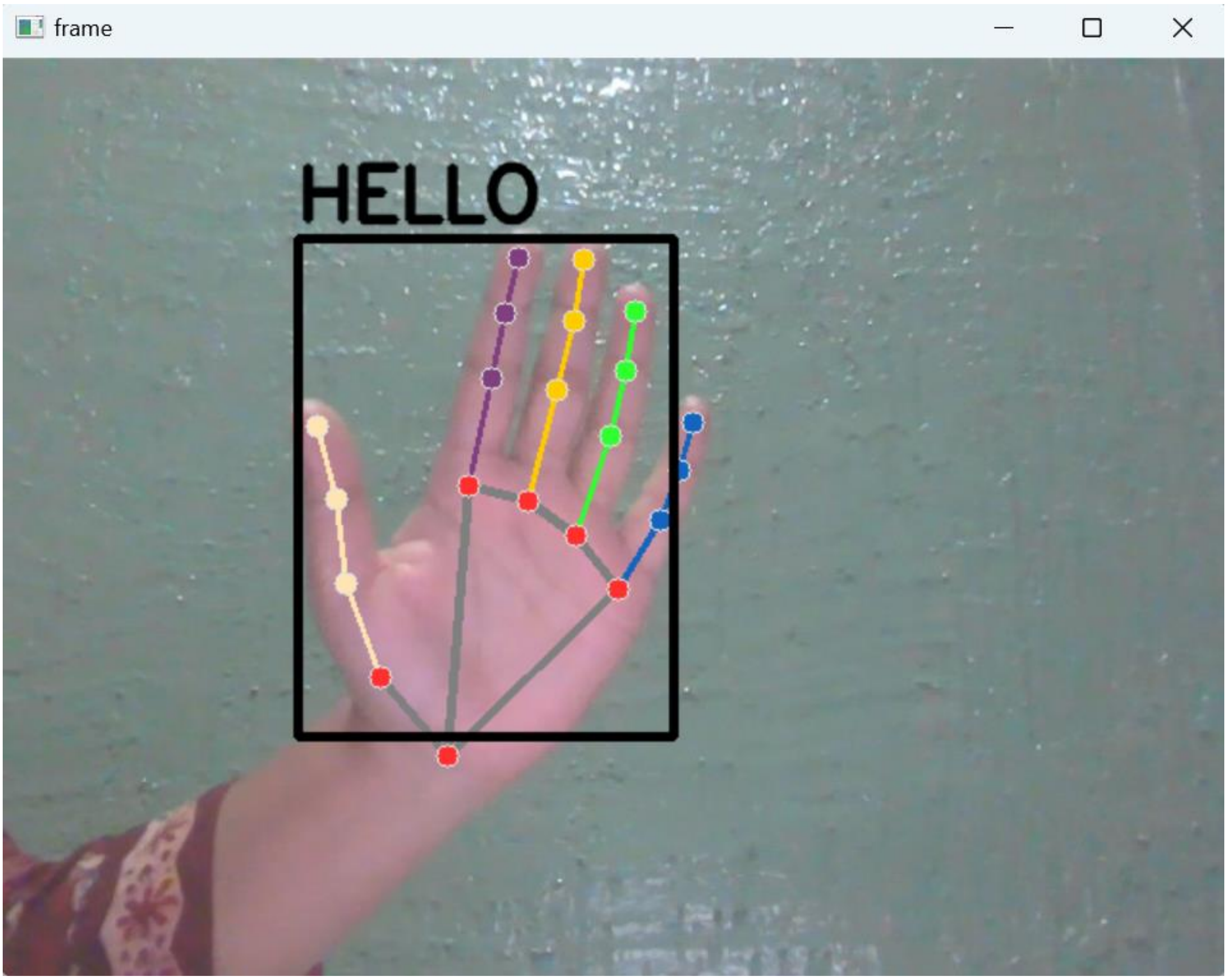
```
data_dict = pickle.load(open('./data.pickle', 'rb'))

data = np.asarray(data_dict['data'])
labels = np.asarray(data_dict['labels'])

x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, shuffle=True, stratify=labels)

model = RandomForestClassifier()

model.fit(x_train, y_train)

y_predict = model.predict(x_test)

score = accuracy_score(y_predict, y_test)

print('{}% of samples were classified correctly !'.format(score * 100))
```

```
PS D:\SignLanguageRecognition> python -u "d:\SignLanguageRecognition\train_classifier.py"
100.0% of samples were classified correctly !
PS D:\SignLanguageRecognition>
```

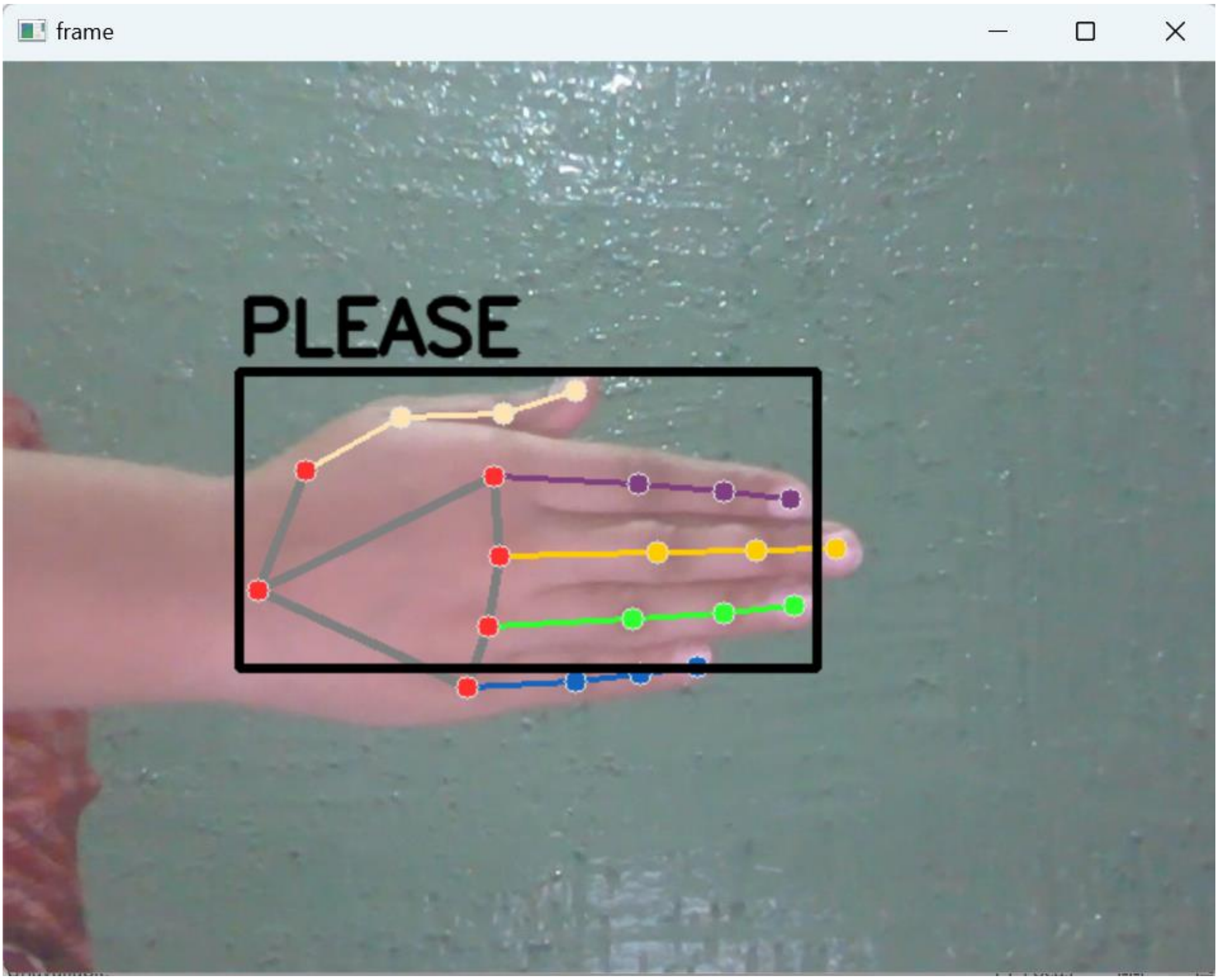
PREDICTING THE OUTPUT

```
    prediction = model.predict([np.asarray(data_aux)])

    predicted_character = labels_dict[int(prediction[0])]

    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 0), 4)
    cv2.putText(frame, predicted_character, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 0, 0), 3,
                cv2.LINE_AA)

cv2.imshow('frame', frame)
cv2.waitKey(1)
```

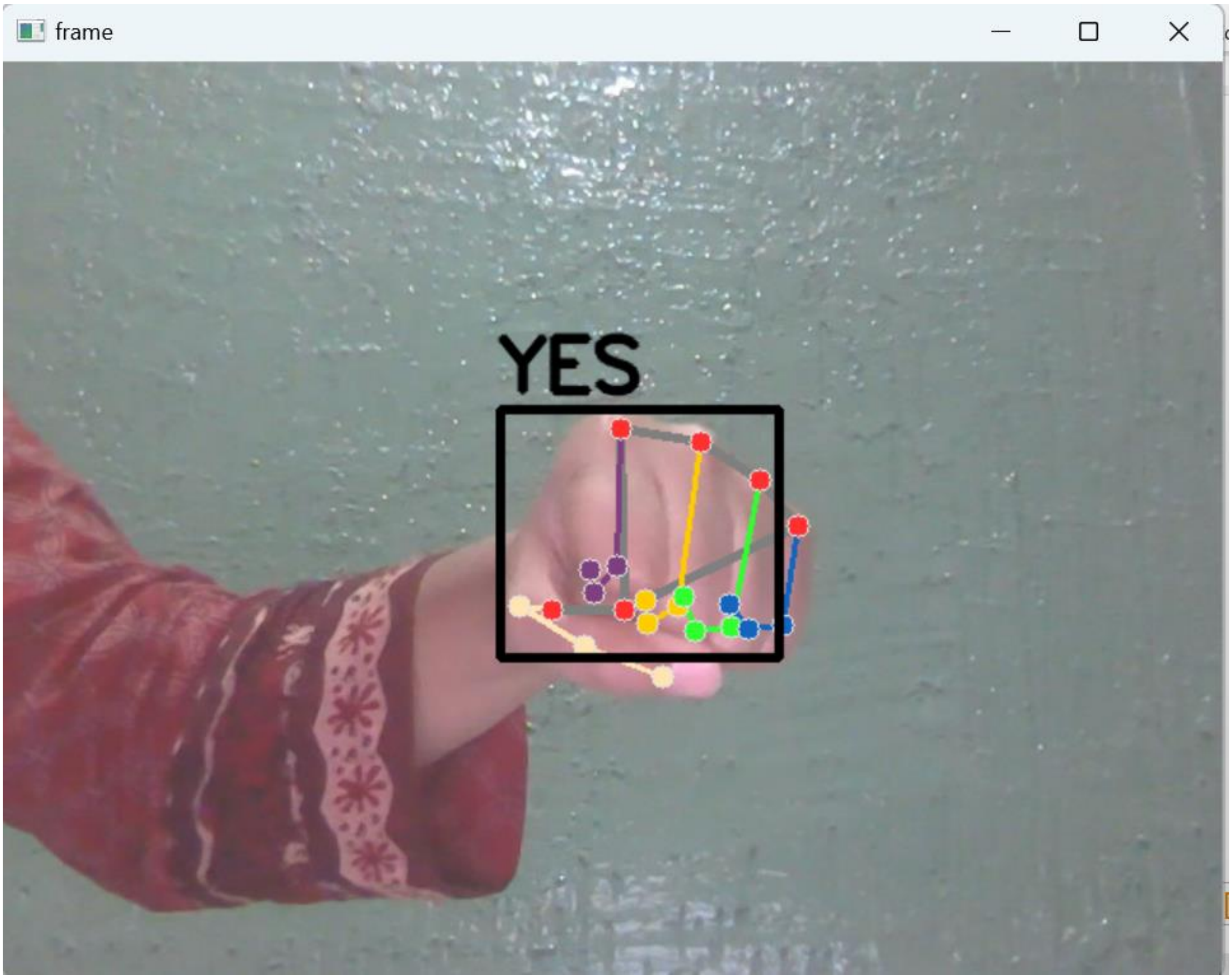
```
labels_dict = {0: 'HELLO', 1: 'PLEASE', 2: 'YES', 3:'NO', 4:'I LOVE YOU'}
```



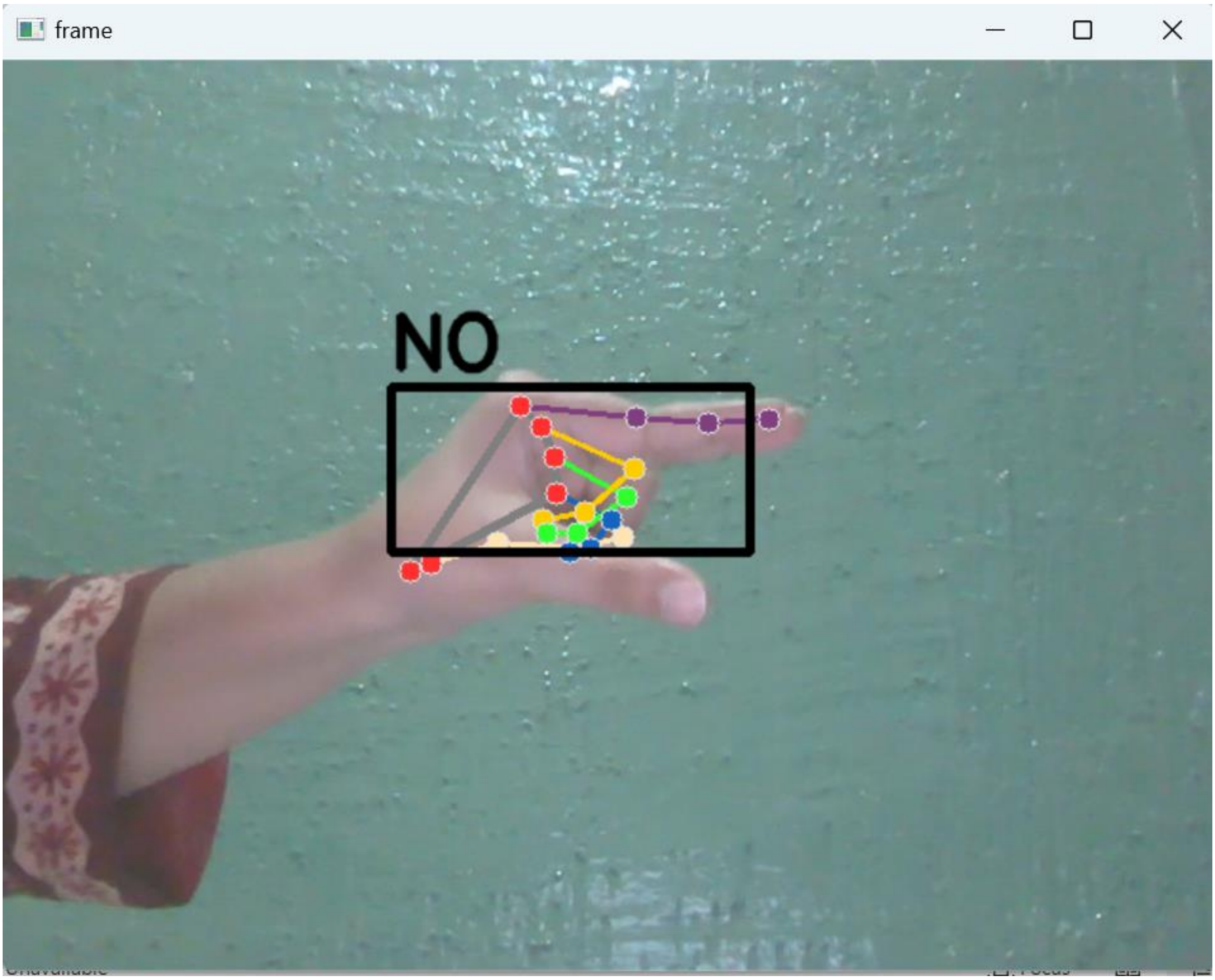
```
labels_dict = {0: 'HELLO', 1: 'PLEASE', 2: 'YES', 3:'NO', 4:'I LOVE YOU'}
```



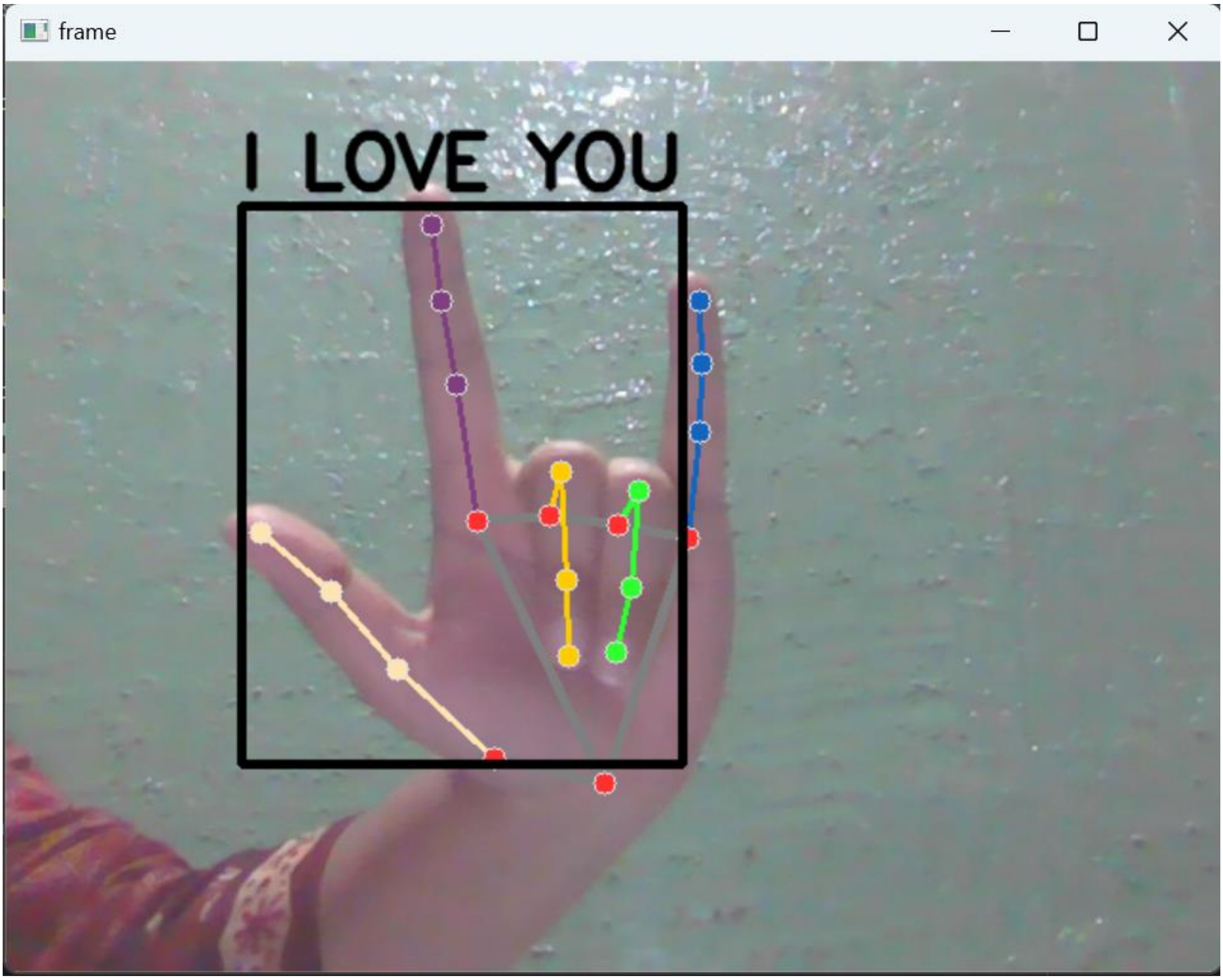
```
labels_dict = {0: 'HELLO', 1: 'PLEASE', 2: 'YES', 3:'NO', 4:'I LOVE YOU'}
```



```
labels_dict = {0: 'HELLO', 1: 'PLEASE', 2: 'YES', 3:'NO', 4:'I LOVE YOU'}
```



```
labels_dict = {0: 'HELLO', 1: 'PLEASE', 2: 'YES', 3:'NO', 4:'I LOVE YOU'}
```



CONCLUSION

In conclusion, sign language detection technology represents a groundbreaking innovation with the potential to transform communication accessibility for individuals with hearing impairments. By leveraging the power of computer vision and machine learning, these systems have the ability to break down communication barriers, empower individuals with hearing impairments, and promote greater equality and inclusion in society. However, further research and development efforts are needed to address the technical challenges and ensure the effectiveness and usability of sign language detection systems in real-world settings. With continued advancements in technology and collaboration between researchers, engineers and hard of hearing community, sign language detection technology can become an invaluable tool for promoting communication accessibility and empowerment for all.

LIMITATIONS

Despite the promising advancements in sign language detection technology, several challenges remain to be addressed. One significant challenge is the diversity of sign languages and dialects used around the world, each with its own unique vocabulary, grammar, and syntax. Developing sign language detection systems that can accurately recognize and interpret multiple sign languages poses a considerable technical challenge. Additionally, ensuring the accuracy and reliability of sign language detection algorithms in various environmental conditions, such as different lighting conditions and camera angles, is essential for real-world deployment.

BIBLIOGRAPHY

- https://ijcrt.org/papers/IJCRT2205740.pdf
- https://medium.com/@denizgunay/random-forest-af5bde5d7e1e
- https://www.youtube.com/results?search_query=sign+language+recognition+using+machine+learning
- https://nilservices.com/different-types-sign-language/

PLAGIARISM REPORT

areeba_sabri_report.pdf

ORIGINALITY REPORT

9%	3%	6%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Soft Computing for Security Applications", Springer Science and Business Media LLC, 2023 Publication	3%
2	Submitted to Nottingham Trent University Student Paper	2%
3	Submitted to Liverpool John Moores University Student Paper	1%
4	Pragati Sathia, Swarnalatha P, Shravan Prakash. "Sign Language Interpreter via Gesture Detection", 2023 Third International Conference on Smart Technologies, Communication and Robotics (STCR), 2023 Publication	1%
5	Submitted to Colorado Technical University Student Paper	1%
6	Submitted to University of Greenwich Student Paper	1%
7	open-innovation-projects.org Internet Source	1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		