Team Members

1.	Piyush Raundal	202301060043
2.	Ravikumar Langade	202301060046
3.	Nachiket Mahajan	202301060047
4.	Sahil Raichurkar	202301060051

Batch – D2

Project Title:

Real-Time Hand Gesture Recognition for Multi-DOF Robotic Arm Control.

Aim: To design and implement a real-time system that recognizes human hand gestures using MediaPipe and machine learning, and uses the recognized gestures to control a multi-DOF robotic arm.

Objective:

- To detect and track hand gestures in real time.
- To train a machine learning model that classifies hand gestures based on extracted features.
- To interface the ML model output with a multiDOF robotic arm for motion control.

Tools & Software Required:

- **Python –** Google Colab
- **Libraries:** mediapipe , numpy, cv2, serial, flask, ngrok
- **Dataset:** hand_landmarks.csv

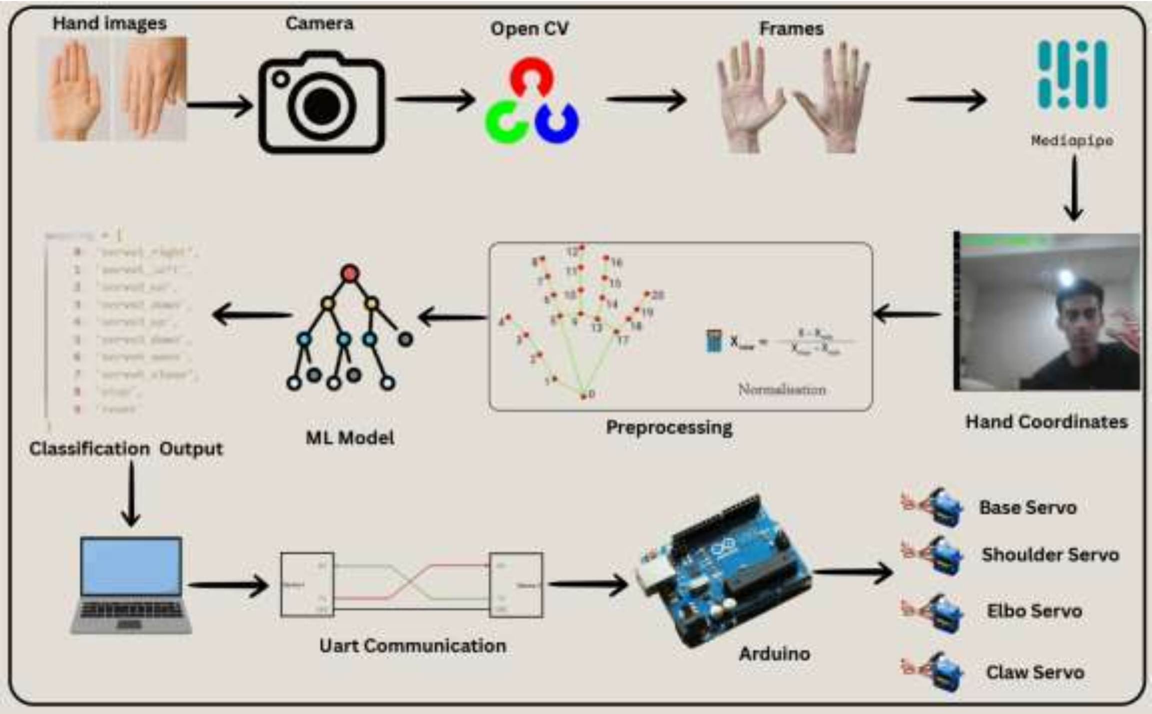
Theory :

The system is based on real-time hand-gesture recognition using MediaPipe, a computer vision framework that detects 21 key landmark points on the human hand. These landmarks capture the shape and orientation of the hand for every gesture. Since different users place their hands at different distances and angles, the raw coordinates must be normalized. Min-Max normalization is applied to scale all landmark values between 0 and 1 so that the model receives consistent and comparable input data.

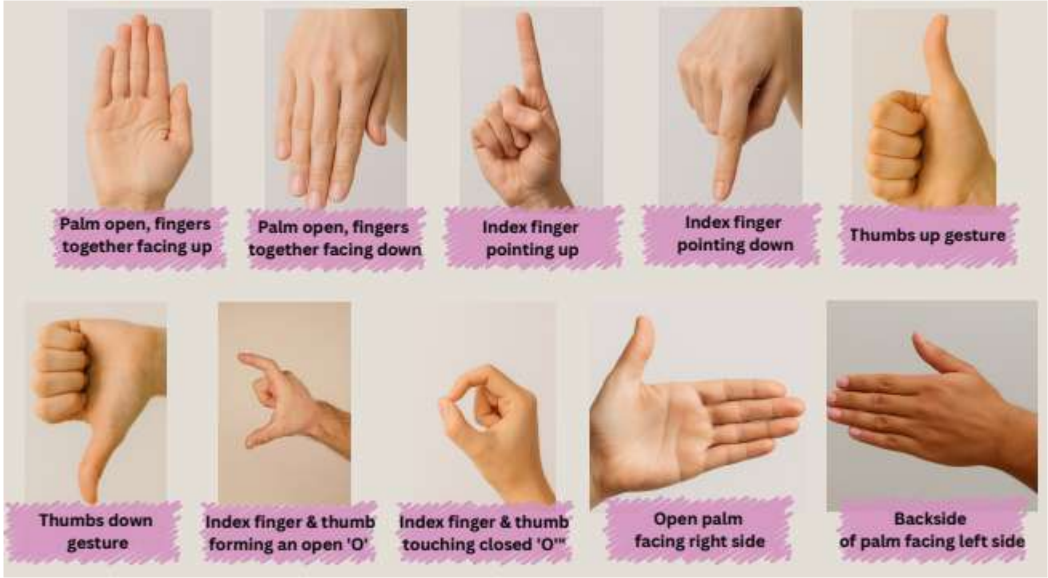
Machine learning is used to classify the hand gestures from the extracted landmark features. Models such as XGBoost, Random Forest, Decision Tree, Naïve Bayes, and SVM trained on labeled gesture data. Among these, Decision Tree is preferred due to its high accuracy, fast training speed, and ability to handle complex patterns in the data. PCA is also used to reduce feature dimensionality, improve model performance, and remove redundant information. After training, each model is evaluated using accuracy scores and classification reports.

The final recognized gesture is mapped to specific actions of a multi-DOF robotic arm. When the system is running in real time, MediaPipe detects hand landmarks, the trained machine learning model predicts the gesture, and the mapped command is sent to the robotic arm via serial communication. This allows the user to control the robotic arm using simple hand movements, making the system efficient, intuitive, and suitable for applications like assistive robotics, touchless interfaces, and human–machine interaction.

Block Diagram



Hand Gestures


Palm open, fingers together facing up
Palm open, fingers together facing down
Index finger pointing up
Index finger pointing down
Thumbs up gesture
Thumbs down gesture
Index finger & thumb forming an open 'O'
Index finger & thumb touching closed 'O'"
Open palm facing right side
Backside of palm facing left side

Data Collection

The dataset for this project was created using a real-time hand-landmark extraction pipeline built with OpenCV and MediaPipe Hands, which detects 21 key hand landmarks. The goal of data collection was to capture multiple gesture samples across different orientations, ensuring high-quality data for training a gesture-recognition model.

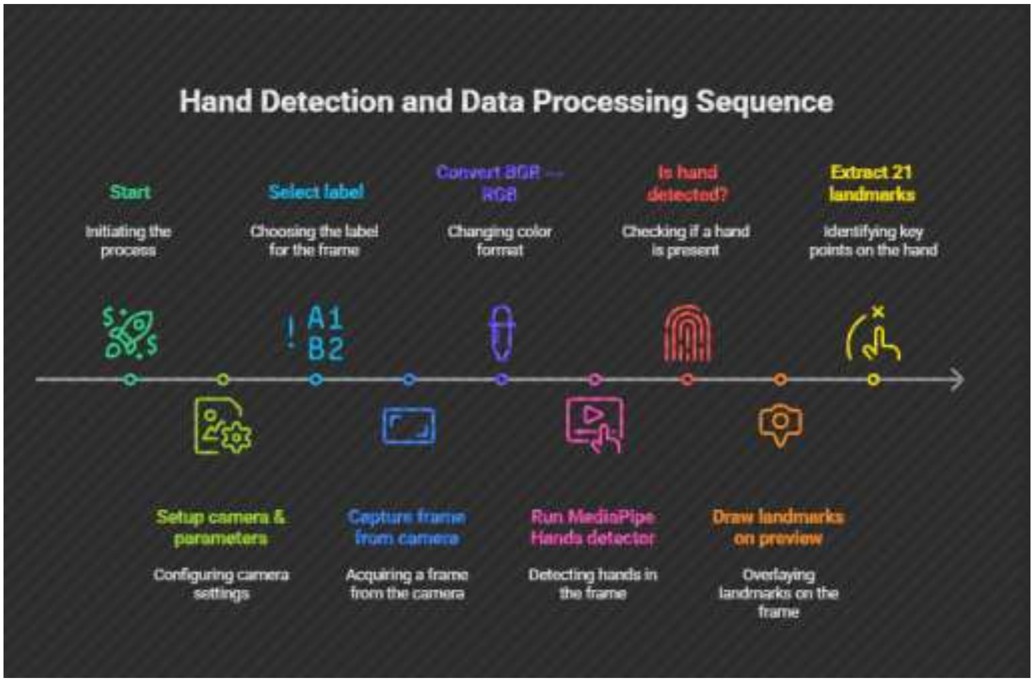
Purpose of Data Collection:

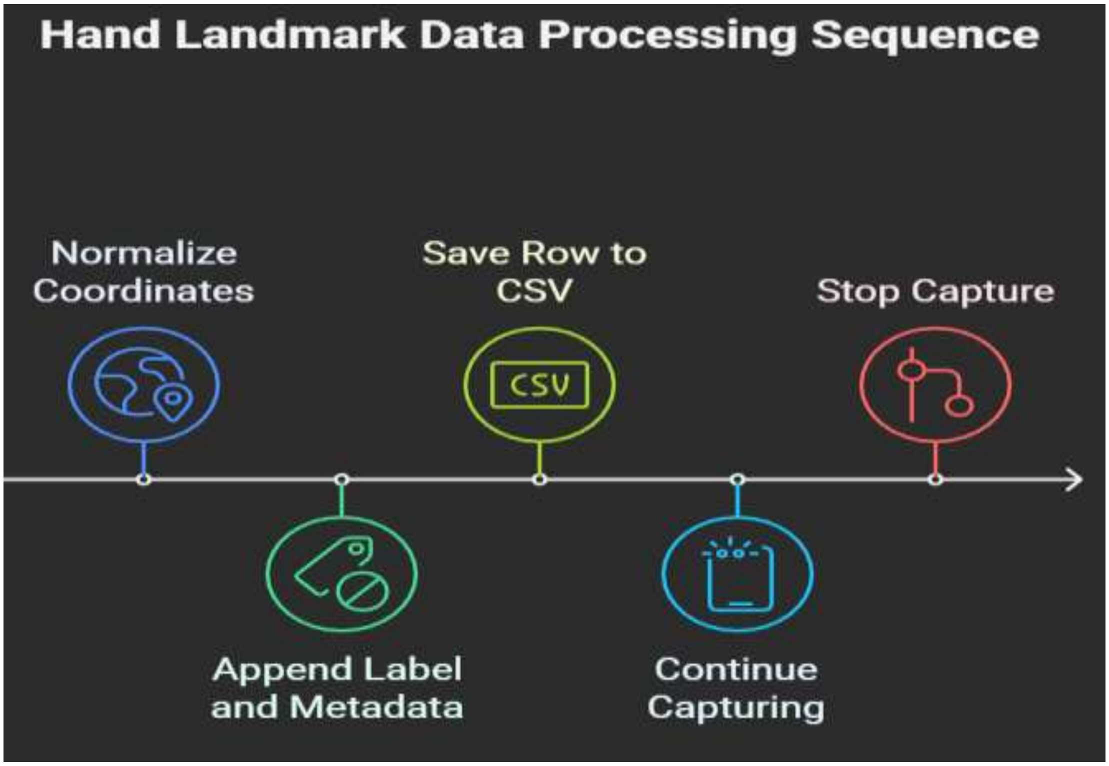
The purpose of data collection was to create a labeled dataset of hand gesture landmark coordinates. Each gesture was assigned a numerical label (0–9), and the corresponding 21 MediaPipe landmarks (x,y) were captured and saved into a CSV file for machine learning model training.

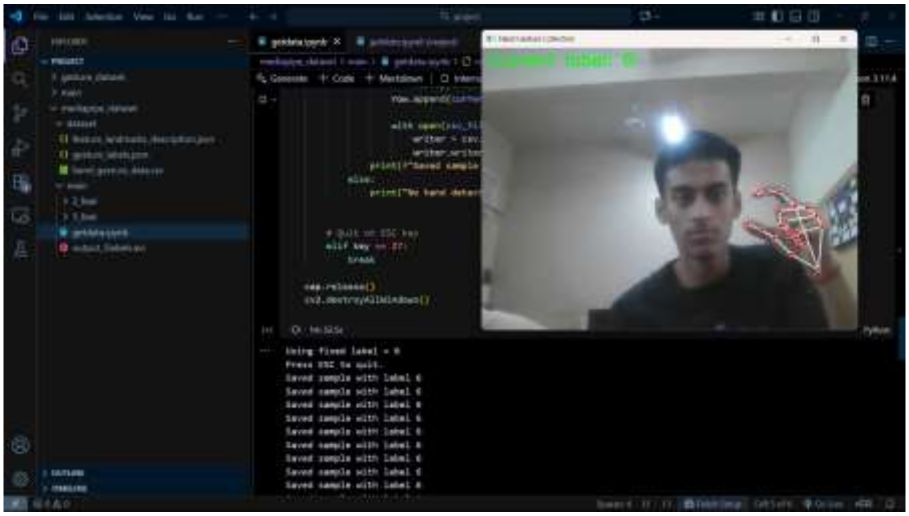
Data Collection Environment:

- The webcam (640×480 resolution) was used as the primary capture device.
- The video stream was flipped horizontally for natural user interaction.
- MediaPipe Hands was used with:
    - min_detection_confidence = 0.7
    - min_tracking_confidence = 0.7

Data Capture Process

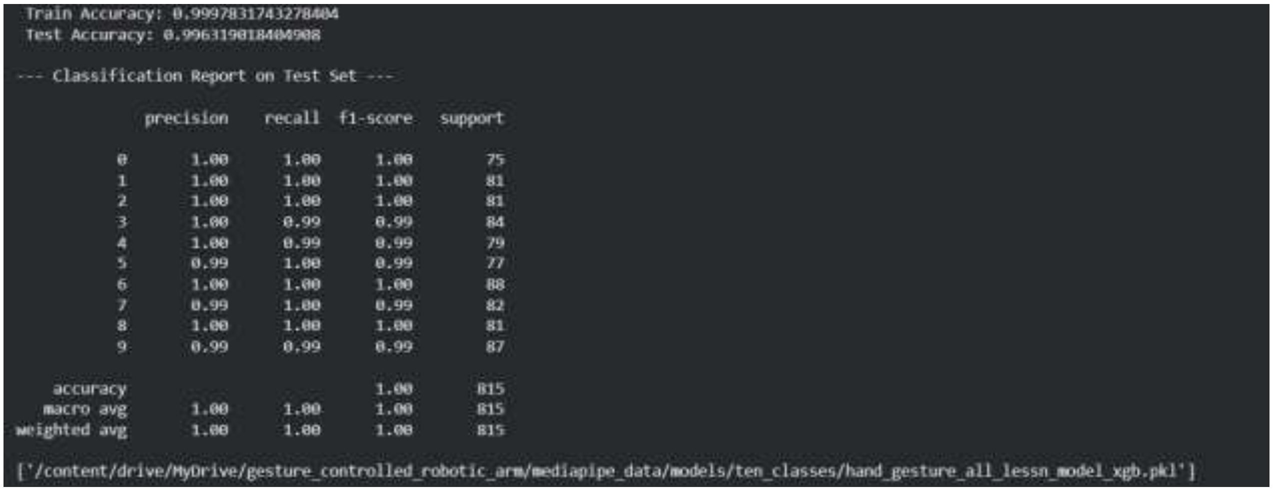

Hand Detection and Data Processing Sequence
Start
Initiating the process
Setup camera & parameters
Configuring camera settings
Select label
Choosing the label for the frame
A1
B2
Capture frame from camera
Acquiring a frame from the camera
Convert BGR → RGB
Changing color format
Run MediaPipe Hands detector
Detecting hands in the frame
Is hand detected?
Checking if a hand is present
Draw landmarks on preview
Overlaying landmarks on the frame
Extract 21 landmarks
Identifying key points on the hand



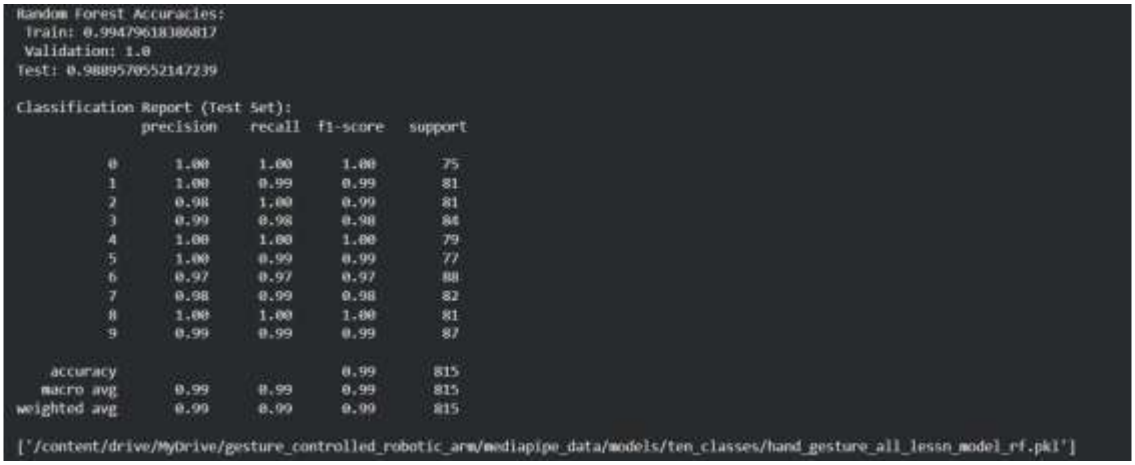


```
                        row.append(
                        with open(csv_fi
                            writer = csv.
                            writer.writer
                    print(f"Saved sample
                else:
                    print("No hand detec

            # Quit on ESC key
            elif key == 27:
                break

    cap.release()
    cv2.destroyAllWindows()
```

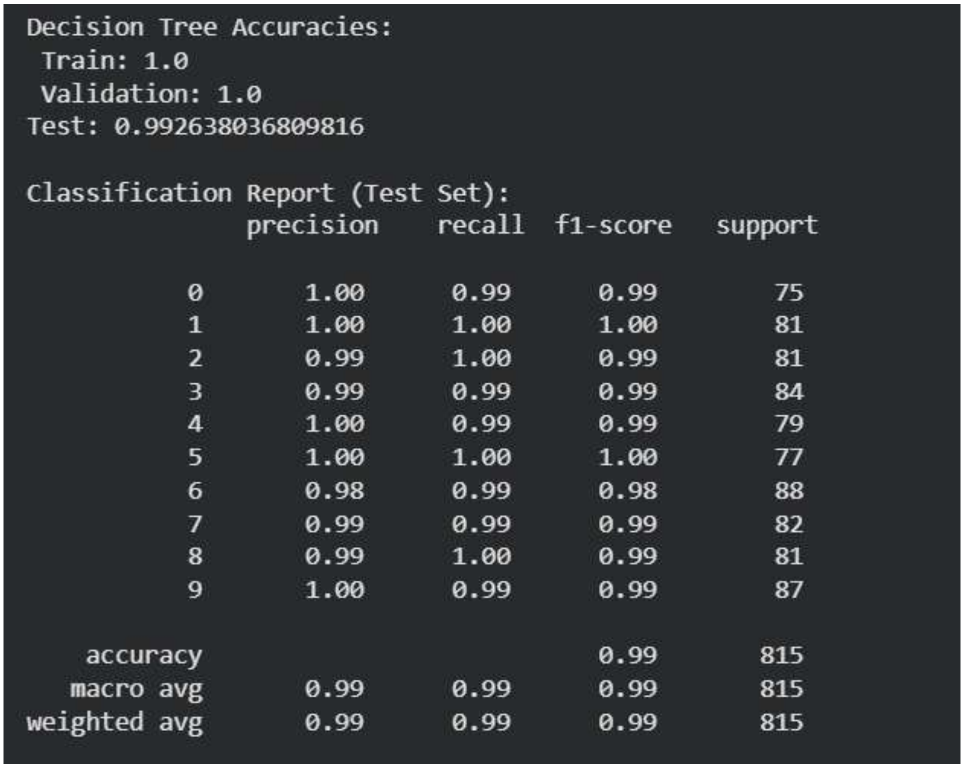
```
Using fixed label = 6
Press ESC to quit.
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
Saved sample with label 6
```



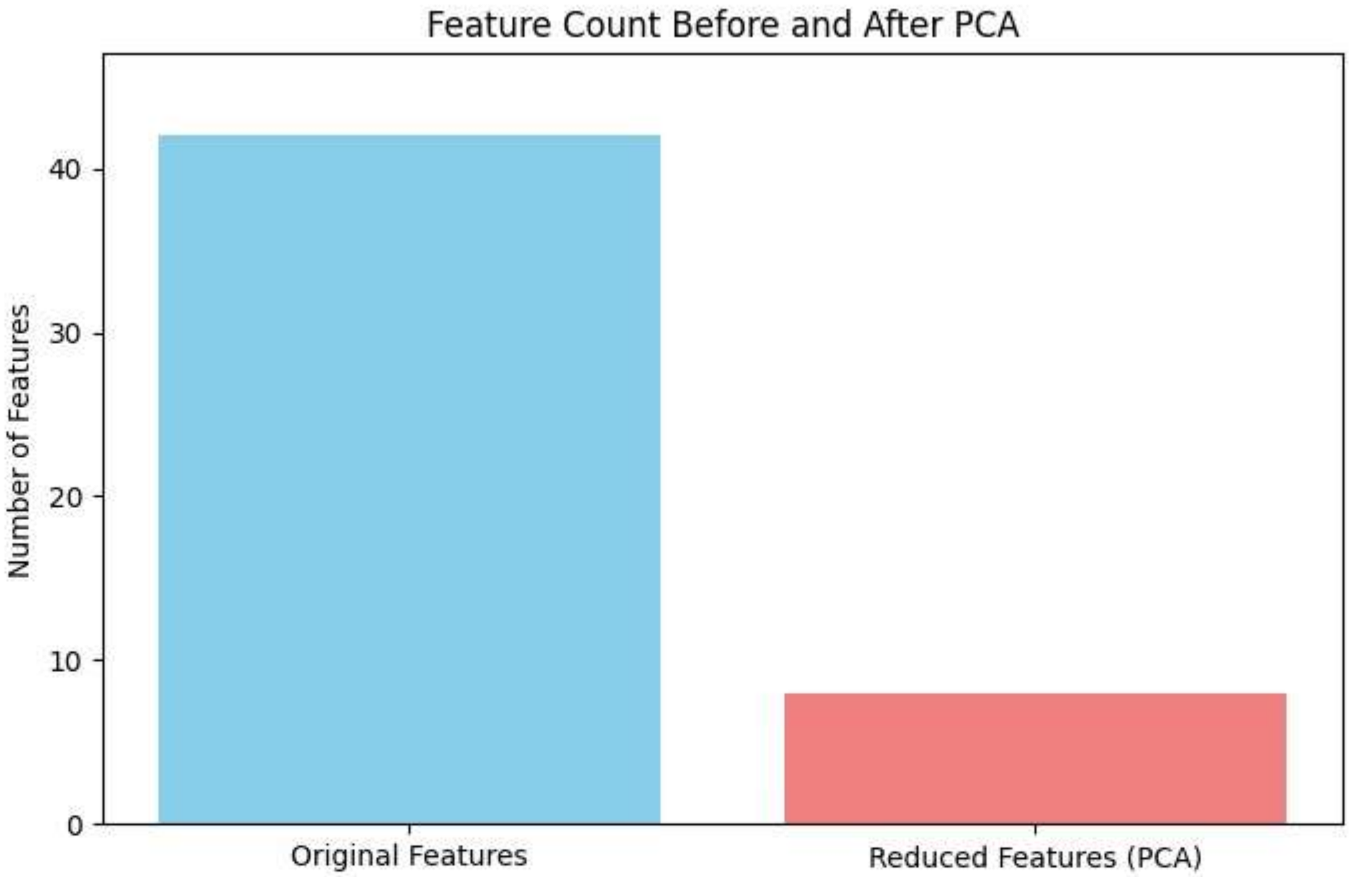
```
 Train Accuracy: 0.9997831743278404
 Test Accuracy: 0.996319018404908

--- Classification Report on Test Set ---

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        75
           1       1.00      1.00      1.00        81
           2       1.00      1.00      1.00        81
           3       1.00      0.99      0.99        84
           4       1.00      0.99      0.99        79
           5       0.99      1.00      0.99        77
           6       1.00      1.00      1.00        88
           7       0.99      1.00      0.99        82
           8       1.00      1.00      1.00        81
           9       0.99      0.99      0.99        87

    accuracy                           1.00       815
   macro avg       1.00      1.00      1.00       815
weighted avg       1.00      1.00      1.00       815

['/content/drive/MyDrive/gesture_controlled_robotic_arm/mediapipe_data/models/ten_classes/hand_gesture_all_lessn_model_xgb.pkl']
```

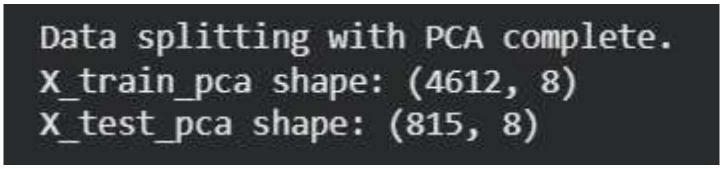


```
Random Forest Accuracies:
 Train: 0.994796183806817
 Validation: 1.0
Test: 0.9889570552147239

Classification Report (Test Set):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        75
           1       1.00      0.99      0.99        81
           2       0.98      1.00      0.99        81
           3       0.99      0.98      0.98        84
           4       1.00      1.00      1.00        79
           5       1.00      0.99      0.99        77
           6       0.97      0.97      0.97        88
           7       0.98      0.99      0.98        82
           8       1.00      1.00      1.00        81
           9       0.99      0.99      0.99        87

    accuracy                           0.99       815
   macro avg       0.99      0.99      0.99       815
weighted avg       0.99      0.99      0.99       815

['/content/drive/MyDrive/gesture_controlled_robotic_arm/mediapipe_data/models/ten_classes/hand_gesture_all_lessn_model_rf.pkl']
```

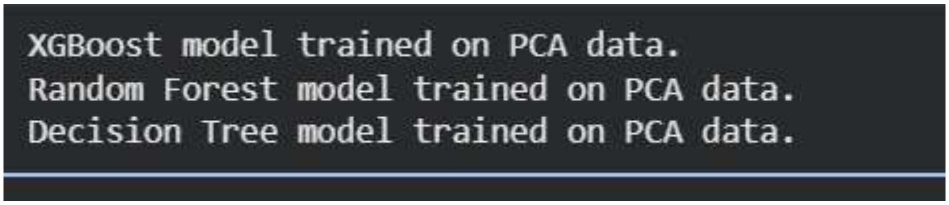


```
Decision Tree Accuracies:
 Train: 1.0
 Validation: 1.0
Test: 0.992638036809816

Classification Report (Test Set):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        75
           1       1.00      1.00      1.00        81
           2       0.99      1.00      0.99        81
           3       0.99      0.99      0.99        84
           4       1.00      0.99      0.99        79
           5       1.00      1.00      1.00        77
           6       0.98      0.99      0.98        88
           7       0.99      0.99      0.99        82
           8       0.99      1.00      0.99        81
           9       1.00      0.99      0.99        87

    accuracy                           0.99       815
   macro avg       0.99      0.99      0.99       815
weighted avg       0.99      0.99      0.99       815
```

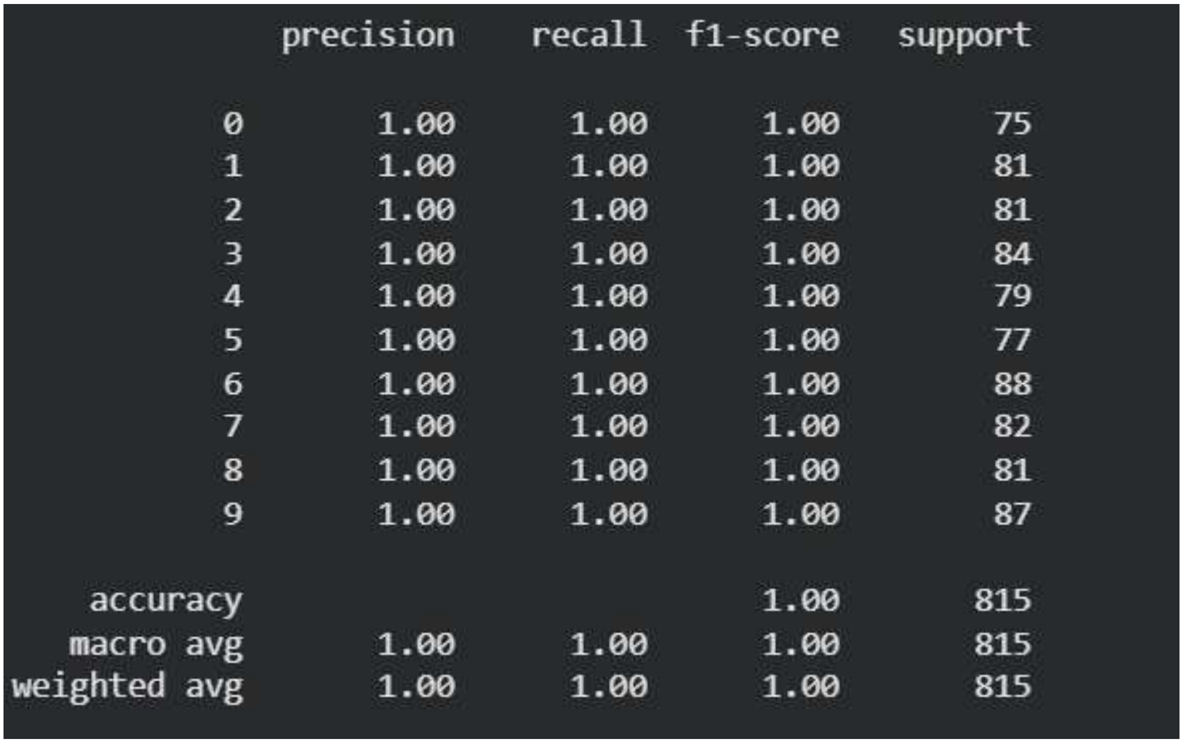




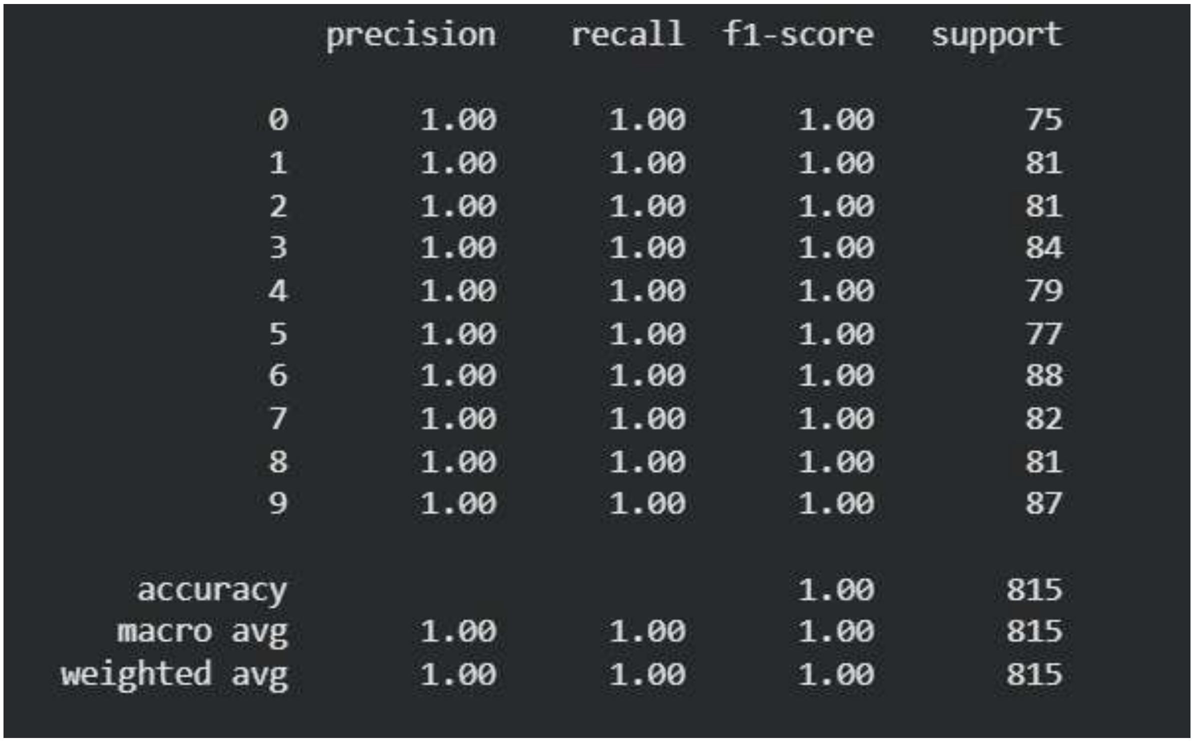
```
Data splitting with PCA complete.
X_train_pca shape: (4612, 8)
X_test_pca shape: (815, 8)
```



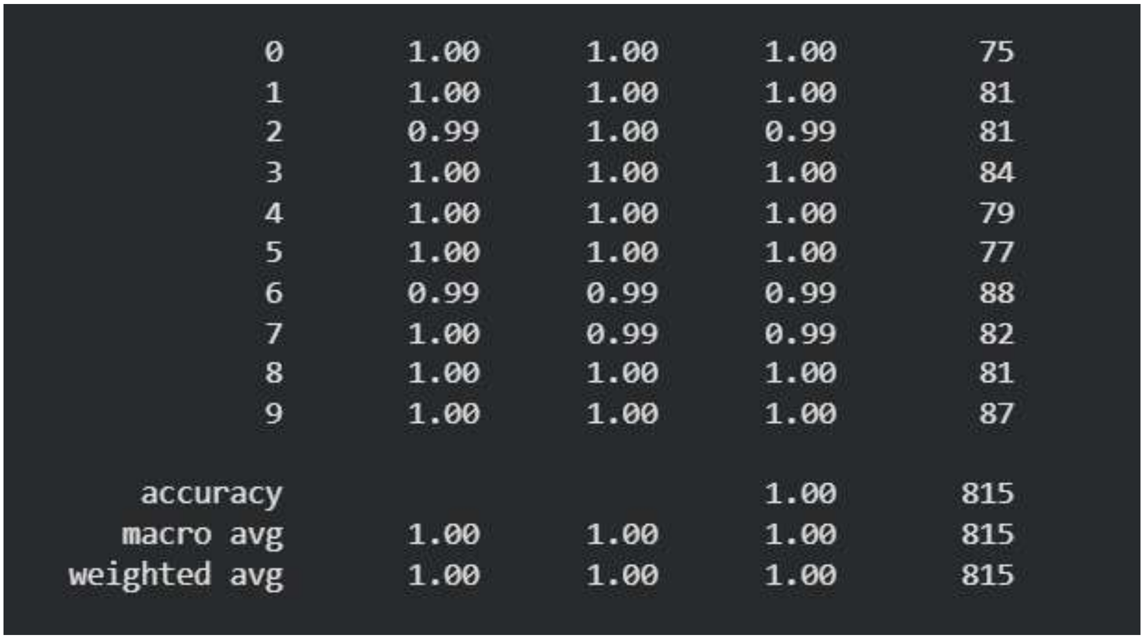
```
XGBoost model trained on PCA data.
Random Forest model trained on PCA data.
Decision Tree model trained on PCA data.
```



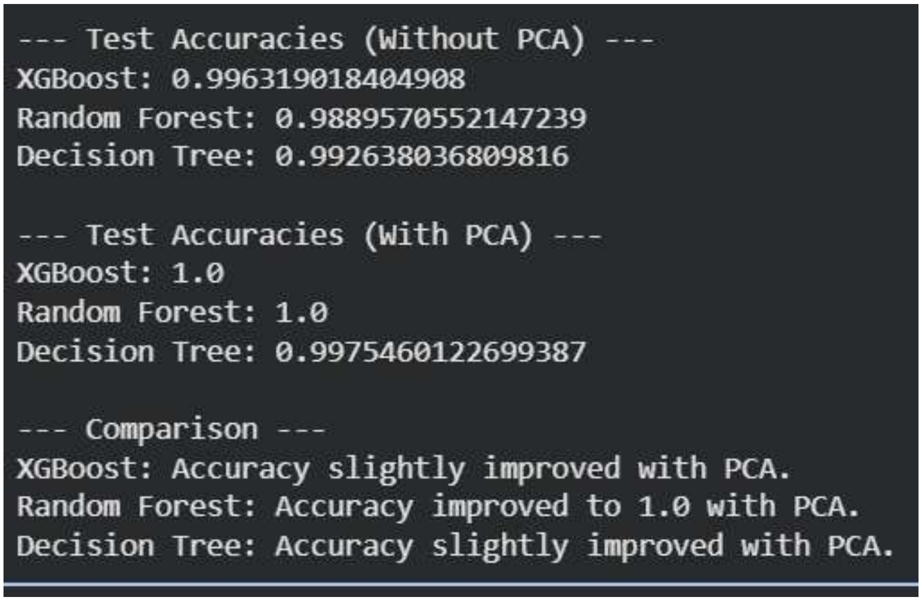
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        75
           1       1.00      1.00      1.00        81
           2       1.00      1.00      1.00        81
           3       1.00      1.00      1.00        84
           4       1.00      1.00      1.00        79
           5       1.00      1.00      1.00        77
           6       1.00      1.00      1.00        88
           7       1.00      1.00      1.00        82
           8       1.00      1.00      1.00        81
           9       1.00      1.00      1.00        87

    accuracy                           1.00       815
   macro avg       1.00      1.00      1.00       815
weighted avg       1.00      1.00      1.00       815
```



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        75
           1       1.00      1.00      1.00        81
           2       1.00      1.00      1.00        81
           3       1.00      1.00      1.00        84
           4       1.00      1.00      1.00        79
           5       1.00      1.00      1.00        77
           6       1.00      1.00      1.00        88
           7       1.00      1.00      1.00        82
           8       1.00      1.00      1.00        81
           9       1.00      1.00      1.00        87

    accuracy                           1.00       815
   macro avg       1.00      1.00      1.00       815
weighted avg       1.00      1.00      1.00       815
```



```
           0       1.00      1.00      1.00        75
           1       1.00      1.00      1.00        81
           2       0.99      1.00      0.99        81
           3       1.00      1.00      1.00        84
           4       1.00      1.00      1.00        79
           5       1.00      1.00      1.00        77
           6       0.99      0.99      0.99        88
           7       1.00      0.99      0.99        82
           8       1.00      1.00      1.00        81
           9       1.00      1.00      1.00        87

    accuracy                           1.00       815
   macro avg       1.00      1.00      1.00       815
weighted avg       1.00      1.00      1.00       815
```

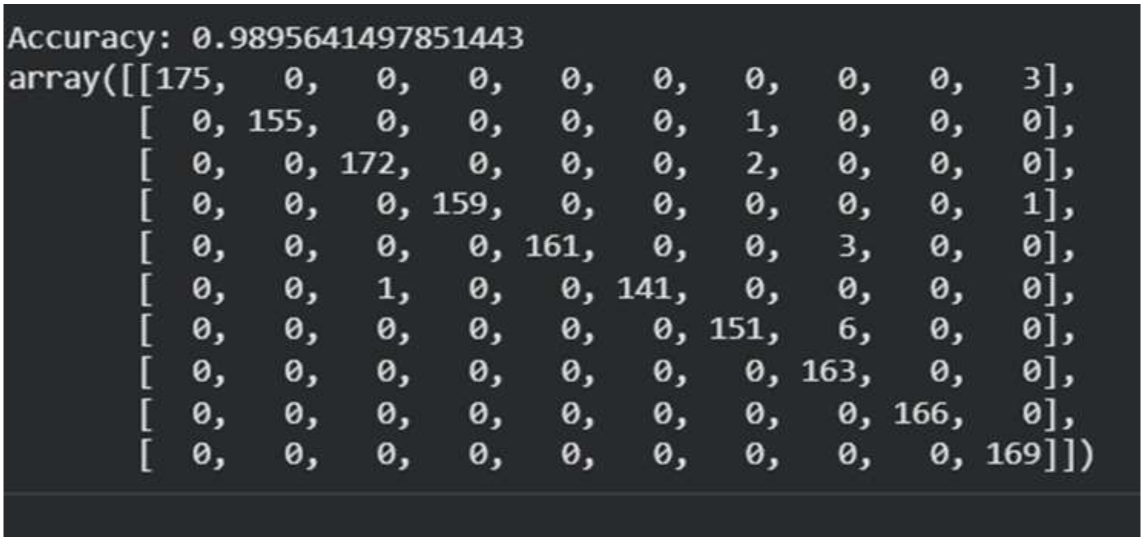


```
--- Test Accuracies (Without PCA) ---
XGBoost: 0.996319018404908
Random Forest: 0.9889570552147239
Decision Tree: 0.992638036809816

--- Test Accuracies (With PCA) ---
XGBoost: 1.0
Random Forest: 1.0
Decision Tree: 0.9975460122699387

--- Comparison ---
XGBoost: Accuracy slightly improved with PCA.
Random Forest: Accuracy improved to 1.0 with PCA.
Decision Tree: Accuracy slightly improved with PCA.
```

XGBoost will be used for this project

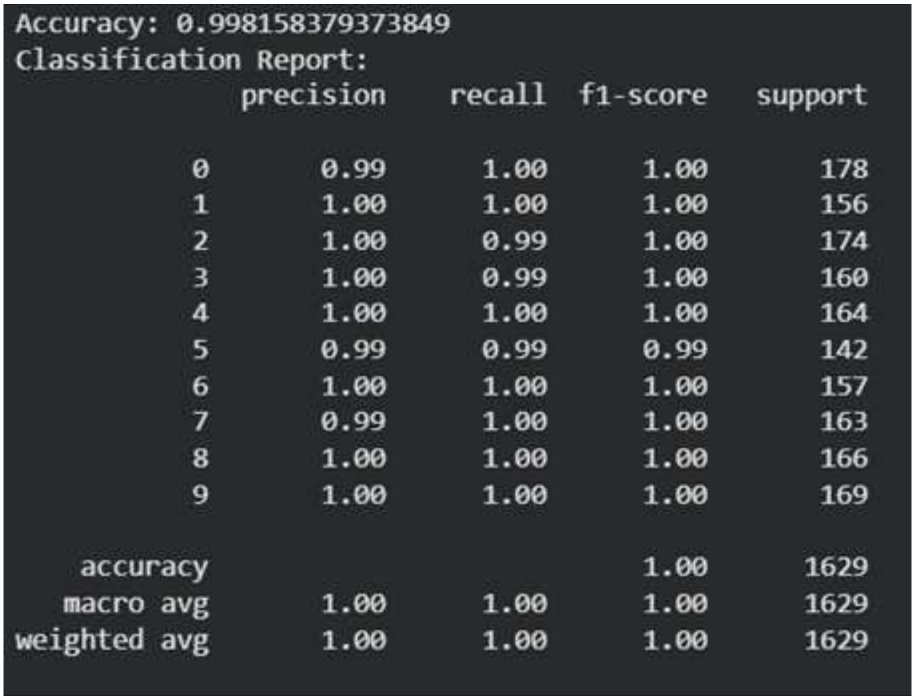
The accuracy of the other models:

Naïve Bayes



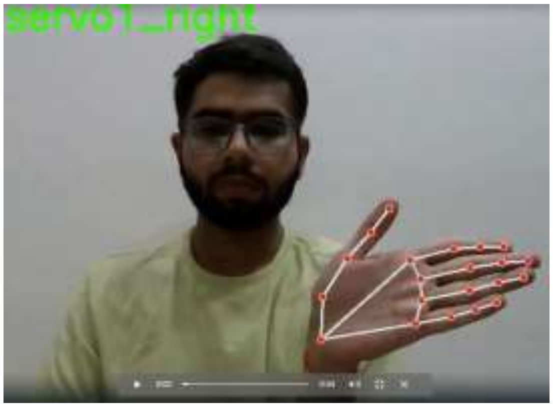
```
Accuracy: 0.9895641497851443
array([[175,   0,   0,   0,   0,   0,   0,   0,   0,   3],
       [  0, 155,   0,   0,   0,   0,   1,   0,   0,   0],
       [  0,   0, 172,   0,   0,   0,   2,   0,   0,   0],
       [  0,   0,   0, 159,   0,   0,   0,   0,   0,   1],
       [  0,   0,   0,   0, 161,   0,   0,   3,   0,   0],
       [  0,   0,   1,   0,   0, 141,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0, 151,   6,   0,   0],
       [  0,   0,   0,   0,   0,   0,   0, 163,   0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0, 166,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0, 169]])
```

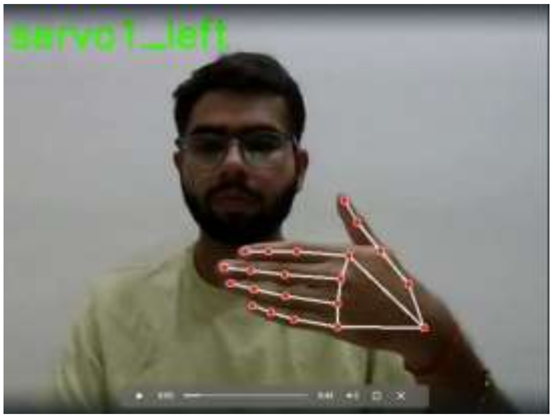
SVM Classifier

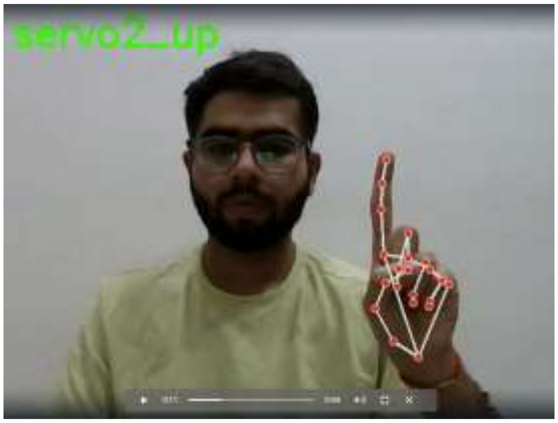


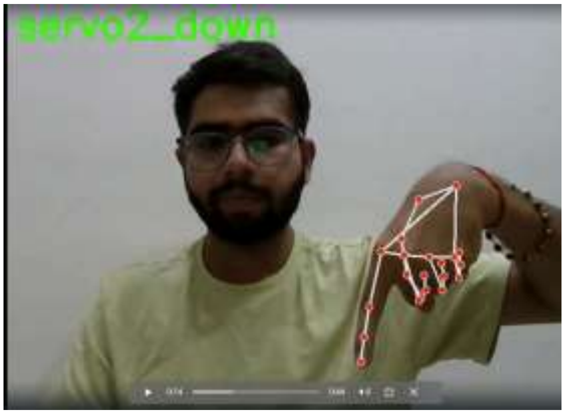
```
Accuracy: 0.9981583793738849
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       178
           1       1.00      1.00      1.00       156
           2       1.00      0.99      1.00       174
           3       1.00      0.99      1.00       160
           4       1.00      1.00      1.00       164
           5       0.99      0.99      0.99       142
           6       1.00      1.00      1.00       157
           7       0.99      1.00      1.00       163
           8       1.00      1.00      1.00       166
           9       1.00      1.00      1.00       169

    accuracy                           1.00      1629
   macro avg       1.00      1.00      1.00      1629
weighted avg       1.00      1.00      1.00      1629
```

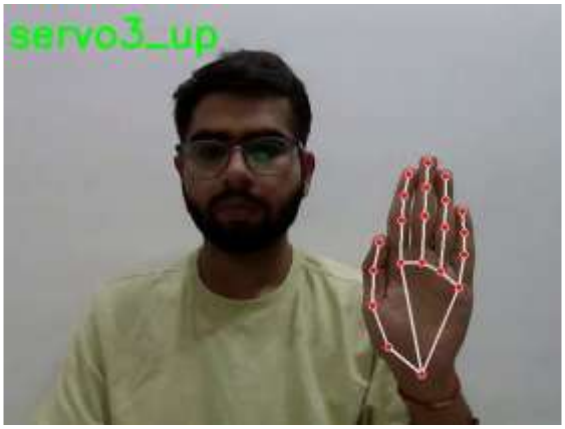
Testing Model on Live Video

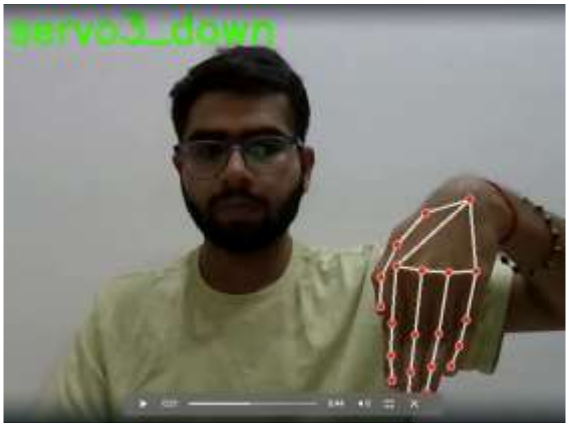


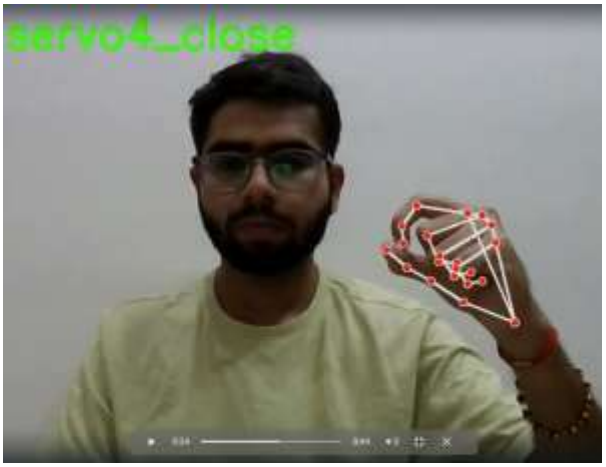


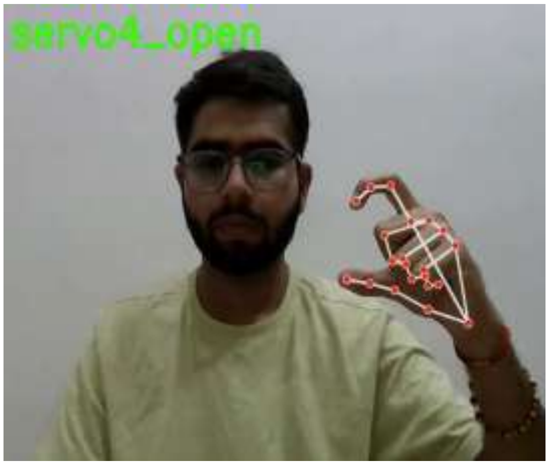

servo2_up

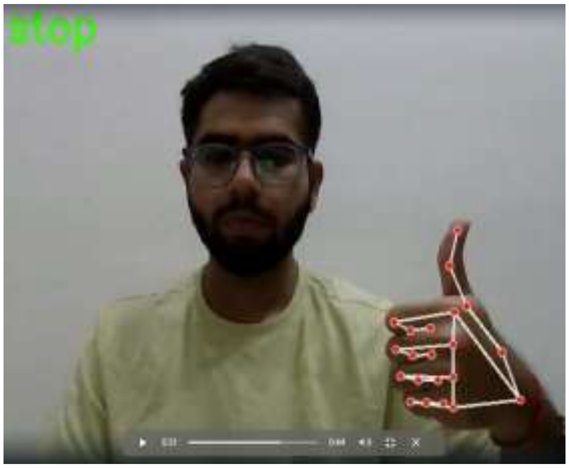

servo2_down

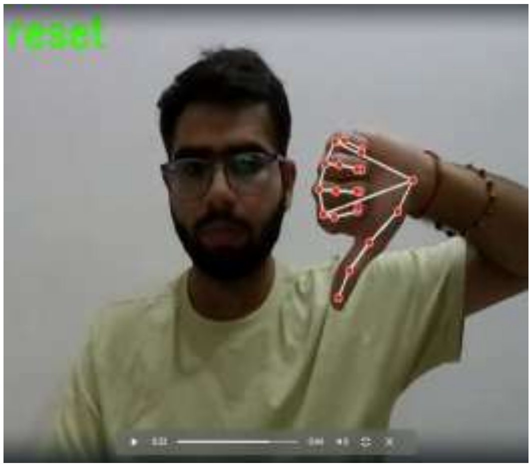

servo3_up


servo3_down


servo4_close


servo4_open


stop

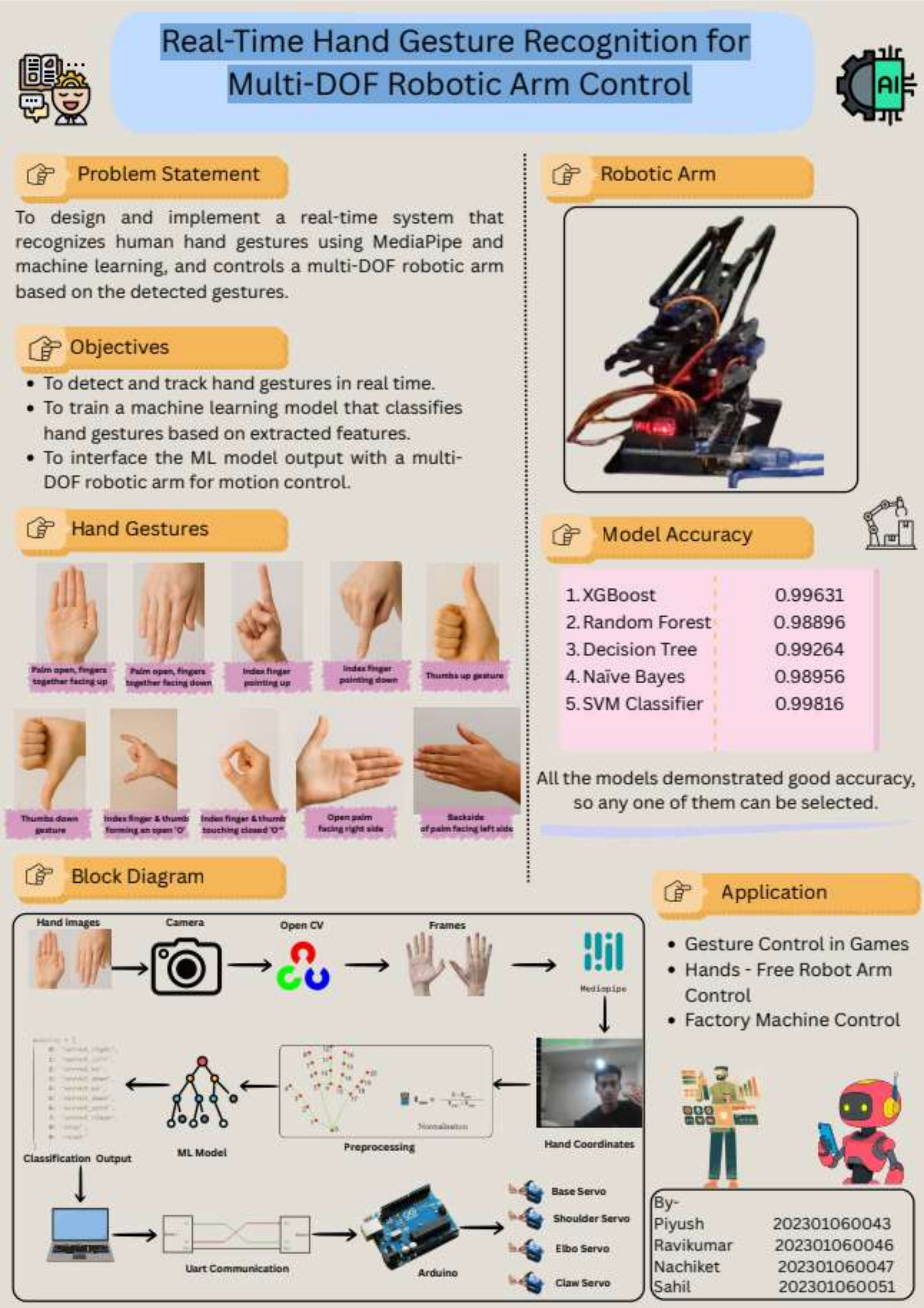

reset

servo1_right

servo4_open

servo2_down

POSTER



Real-Time Hand Gesture Recognition for Multi-DOF Robotic Arm Control

Problem Statement

To design and implement a real-time system that recognizes human hand gestures using MediaPipe and machine learning, and controls a multi-DOF robotic arm based on the detected gestures.

Objectives

- To detect and track hand gestures in real time.
- To train a machine learning model that classifies hand gestures based on extracted features.
- To interface the ML model output with a multi-DOF robotic arm for motion control.

Hand Gestures

Palm open, fingers together facing up
Palm open, fingers together facing down
Index finger pointing up
Index finger pointing down
Thumbs up gesture
Thumbs down gesture
Index finger & thumb forming an open 'O'
Index finger & thumb touching closed 'O'"
Open palm facing right side
Backside of palm facing left side

Block Diagram

Robotic Arm

Model Accuracy

1. XGBoost	0.99631
2. Random Forest	0.98896
3. Decision Tree	0.99264
4. Naïve Bayes	0.98956
5. SVM Classifier	0.99816

All the models demonstrated good accuracy, so any one of them can be selected.

Application

- Gesture Control in Games
- Hands - Free Robot Arm Control
- Factory Machine Control

By-

Piyush	202301060043
Ravikumar	202301060046
Nachiket	202301060047
Sahil	202301060051

Conclusion

This experiment demonstrated that XGBoost is highly effective for classifying hand gestures using normalized MediaPipe landmarks. PCA further improved the performance by reducing feature dimensions while maintaining perfect accuracy. The strength and consistency of XGBoost make it ideal for real-time applications where fast and accurate gesture recognition is required. Based on the results, XGBoost is chosen as the final model for controlling the multi-DOF robotic arm. Therefore, the system is efficient, reliable, and well-suited for practical gesture-based robotic control.