AI Fitness Trainer: Squats Analysis

Presented To: Emanuel Panizzi

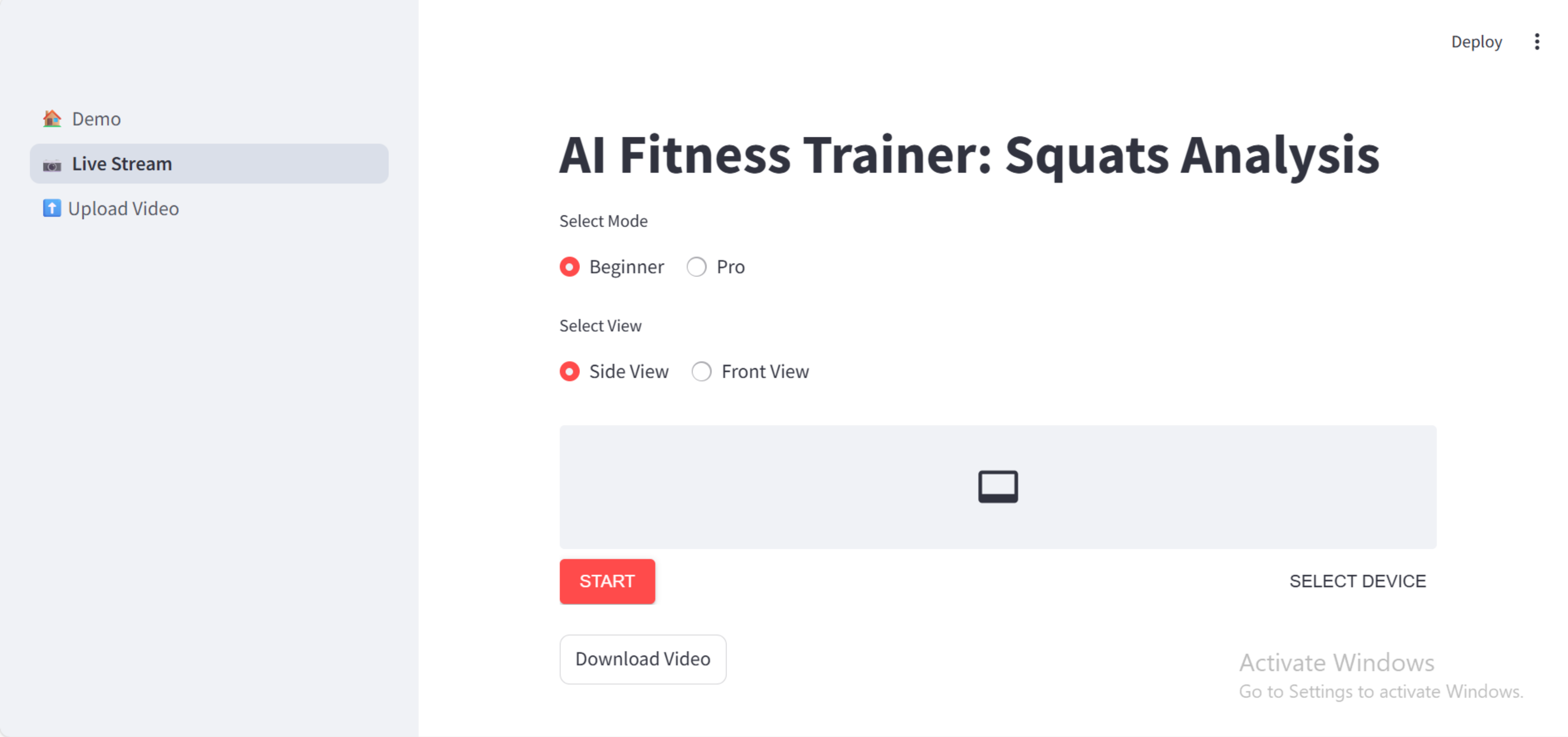
Presented By: Azmeera Qureshi

AI Fitness Trainer: Squats Analysis

Introduction:

We aim to build an **AI fitness trainer** that can help to perform squats seamlessly, irrespective of whether you are a beginner or a pro. To achieve this task we have opted to use Mediapipe's Pose pipeline for estimating the human keypoints.

The application will also have provisions to perform squats in **Beginner** and **Pro** modes, along with appropriate feedback.



AI Fitness Trainer: Squats Analysis

Select Mode

● Beginner ○ Pro

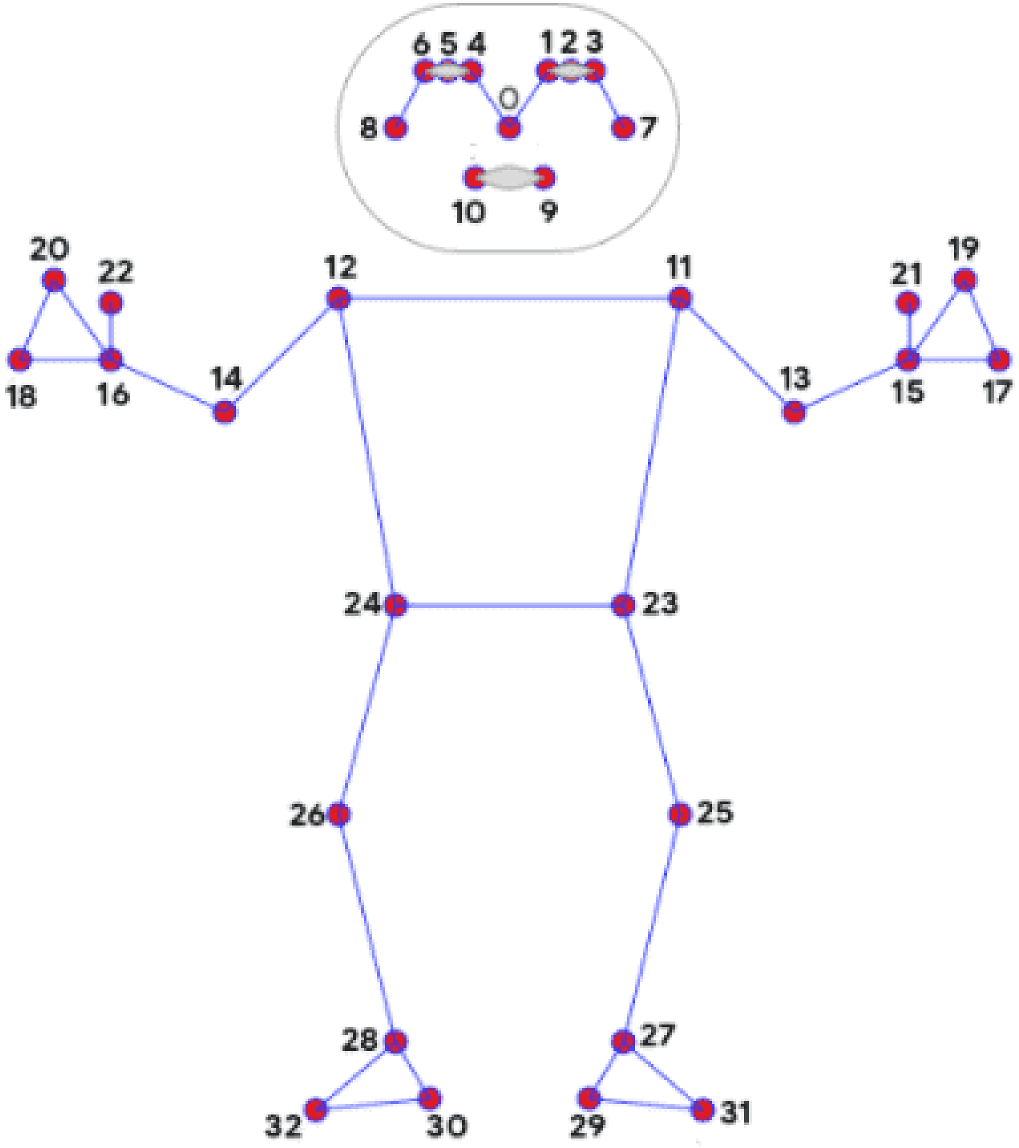
Select View

● Side View ○ Front View

AI Fitness Trainer: Squats Analysis

MediaPipe Pose is an ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation masks on the whole body from RGB video frames utilizing the **BlazePose**, which is a superset of **COCO**, **BlazeFace,** and **BlazePalm** topologies.

The pipeline for MediaPipe pose consists of a two-step detection-tracking pipeline similar to **MediaPipe Hands** and **MediaPipe Face Mesh** solutions. Using a detector, the pipeline first locates the person/pose region-of-interest (ROI) within the frame. The tracker subsequently predicts the pose landmarks and segmentation mask within the ROI using the ROI-cropped frame as input.



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Fig: BlazePose Landmarks

Intuition of Frontal and Side View for Posture Analysis

While designing an application to analyze various fitness exercises, one might be curious to perform various calculations keeping in mind the view of the object (person) from the camera.

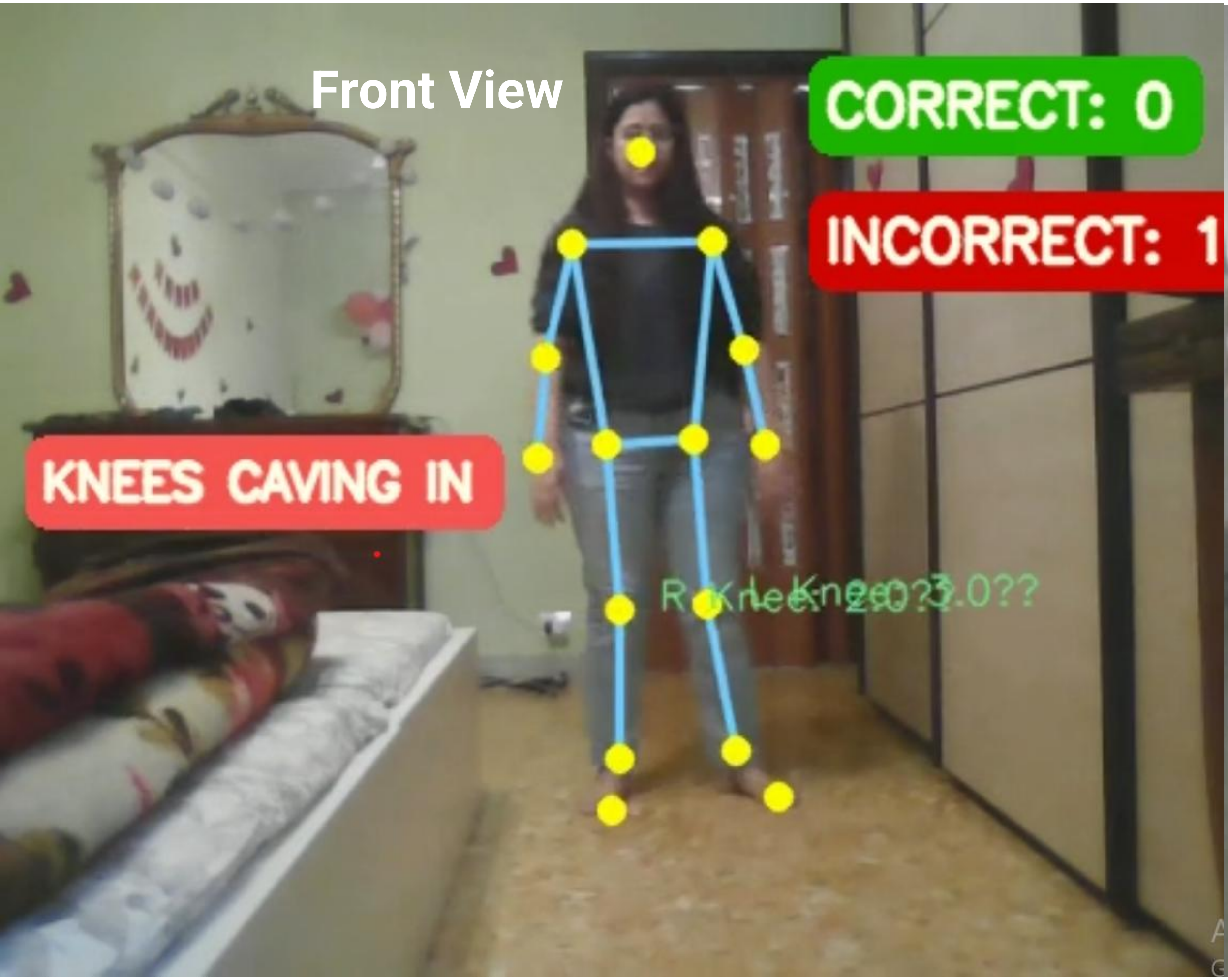
We have added the frontal view option as well as side view. A person can try one view at one time.

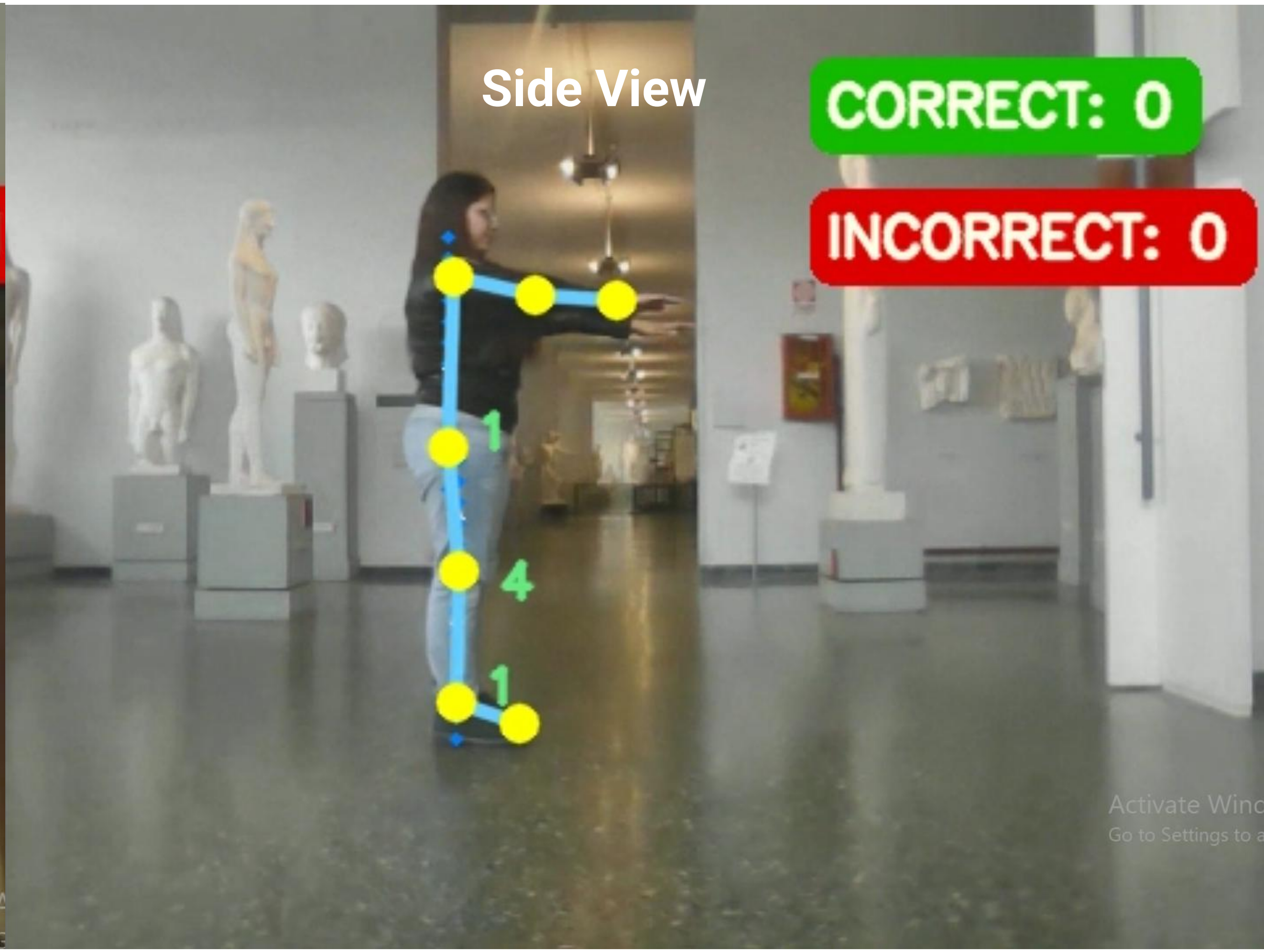
Using the **frontal view**, we have access to both left and right sides and hence can leverage the use of slopes and angles of the various landmark points, such as the angle between the knee-hip and knee-knee lines, etc.

Intuition of Frontal and Side View for Posture Analysis

We can use the **side view** to find better estimates of various inclinations concerning the verticals or horizontals. Such information can be beneficial for analyzing exercises such as deadlifts, pushups, squats, dips, etc.

Since we are analyzing squats and all significant computations concerning the appropriate inclinations with the verticals, we have opted for a side view.


Front View
CORRECT: 0
INCORRECT: 1
KNEES CAVING IN


Side View
CORRECT: 0
INCORRECT: 0
1
4
1

AI Fitness Trainer: Squats Analysis (side view)

AI Fitness Trainer: Squats Analysis (side view)

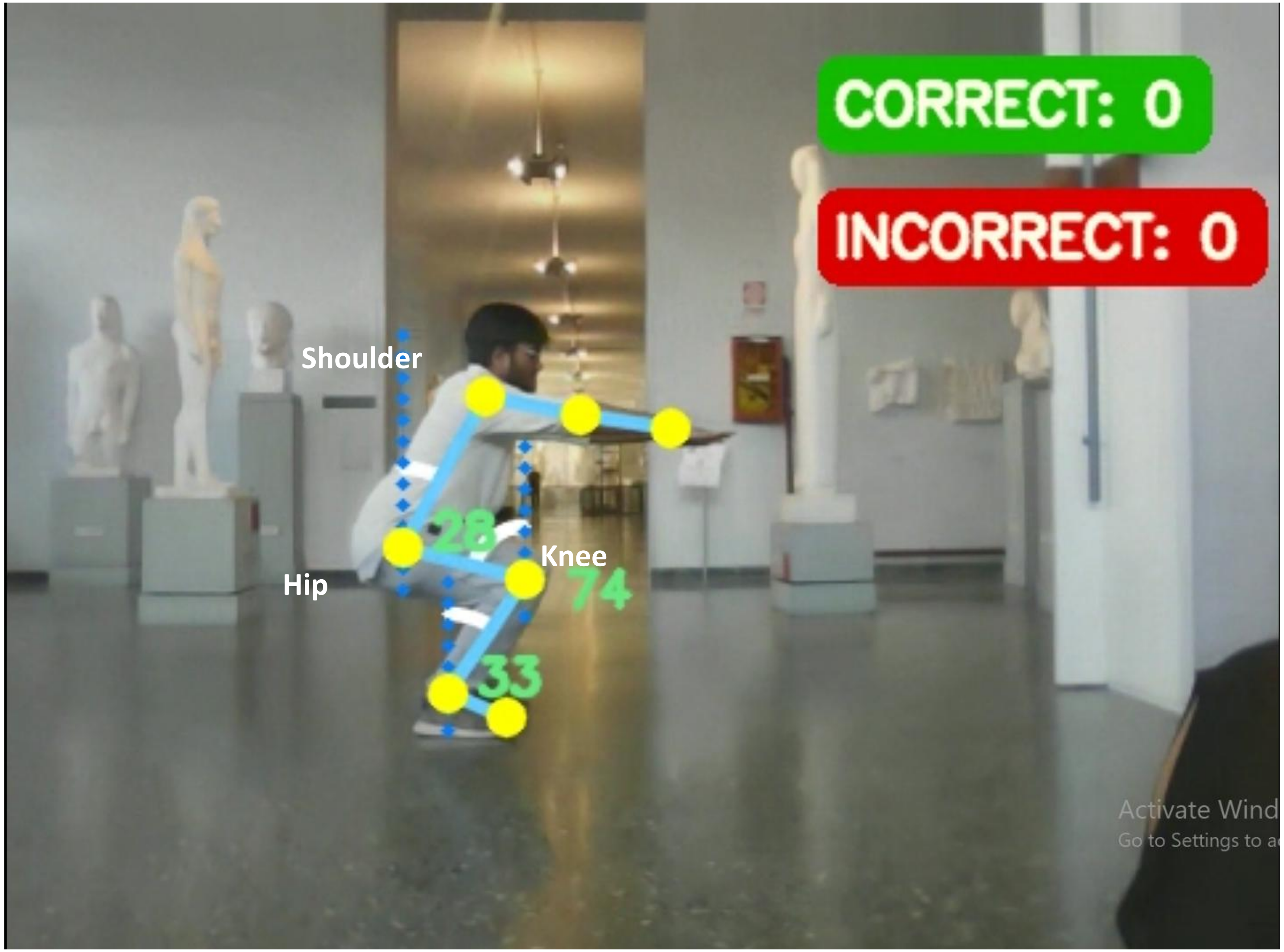
We will consider the angles of the **hip-knee**, **knee-ankle**, and **shoulder-hip** lines with the verticals to calculate the states (explained in the subsequent sections) and perform the appropriate feedback messages. This is depicted in the image below.

Additionally, we shall calculate the **offset angle** (the angle subtended by the nose and the shoulders) with a proper warning to maintain a good side view.

Besides, we will also consider the timings for computing inactivity subject to which the counters for proper and improper squats would be reset.

The application will also provide two modes: **Beginner and Pro**; one can choose either of them and start performing squats seamlessly, irrespective of whether he is a beginner or an expert.

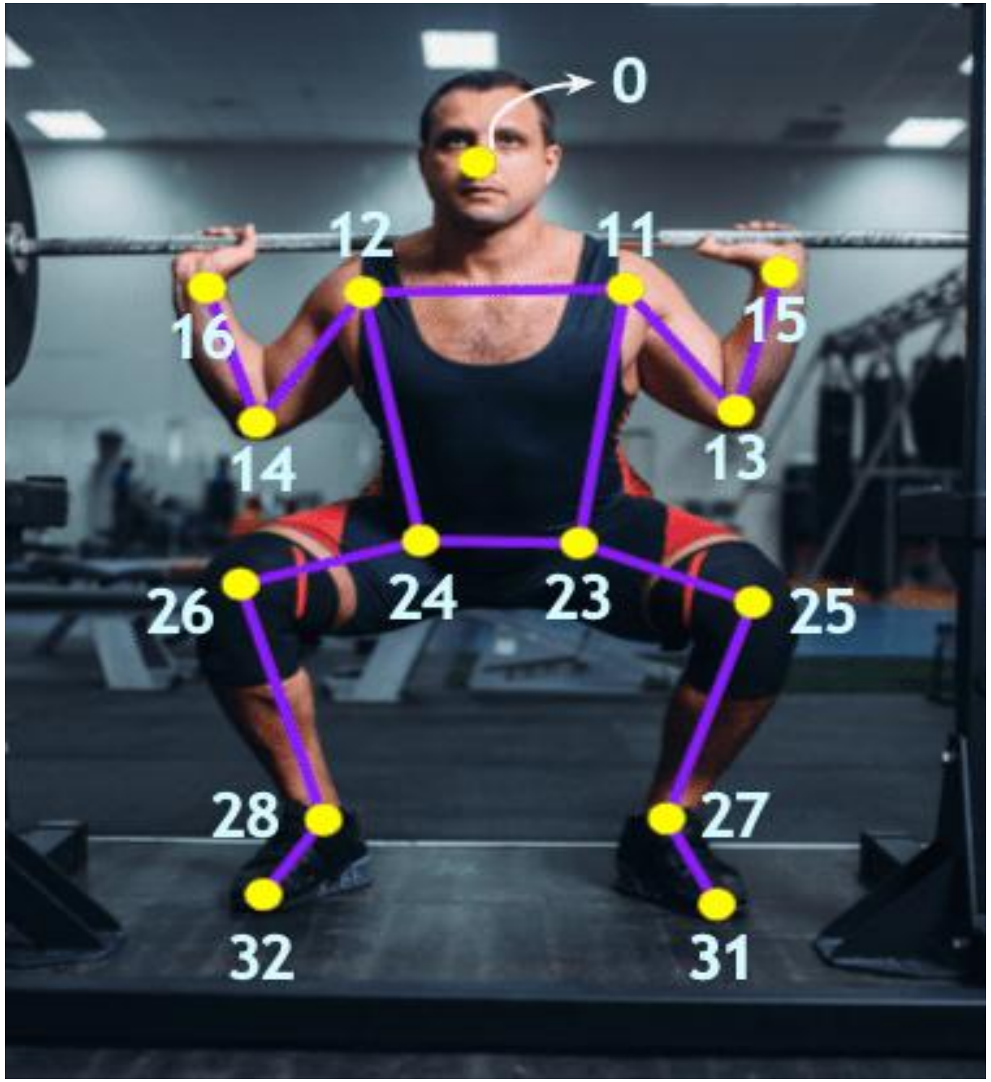
AI Fitness Trainer: Squats Analysis (side view)



Core Functionalities Added

A. Skeleton Detection & Visualization

- **What it does**: Draws a complete body skeleton using 33 pose landmarks
- **Logic**:
  - Uses MediaPipe's pose detection (11-32 landmarks + nose)
  - Connects key joints (shoulders, hips, knees, ankles) with colored lines
  - Only draws visible landmarks (visibility score > 0.1)



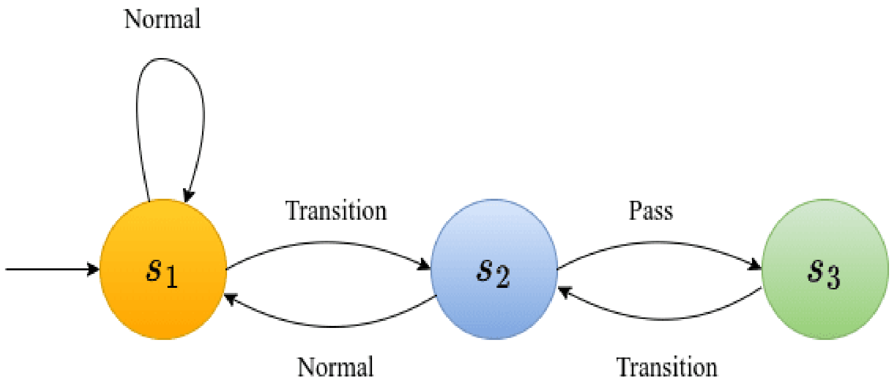
B. Camera Alignment Check

What it does: Ensures proper Side-view positioning

Logic:

- Calculates angle between shoulders and nose (offset_angle)
- Checks if all key landmarks are visible
- Threshold: OFFSET_THRESH = 35° (configurable)

C. Squat State Machine

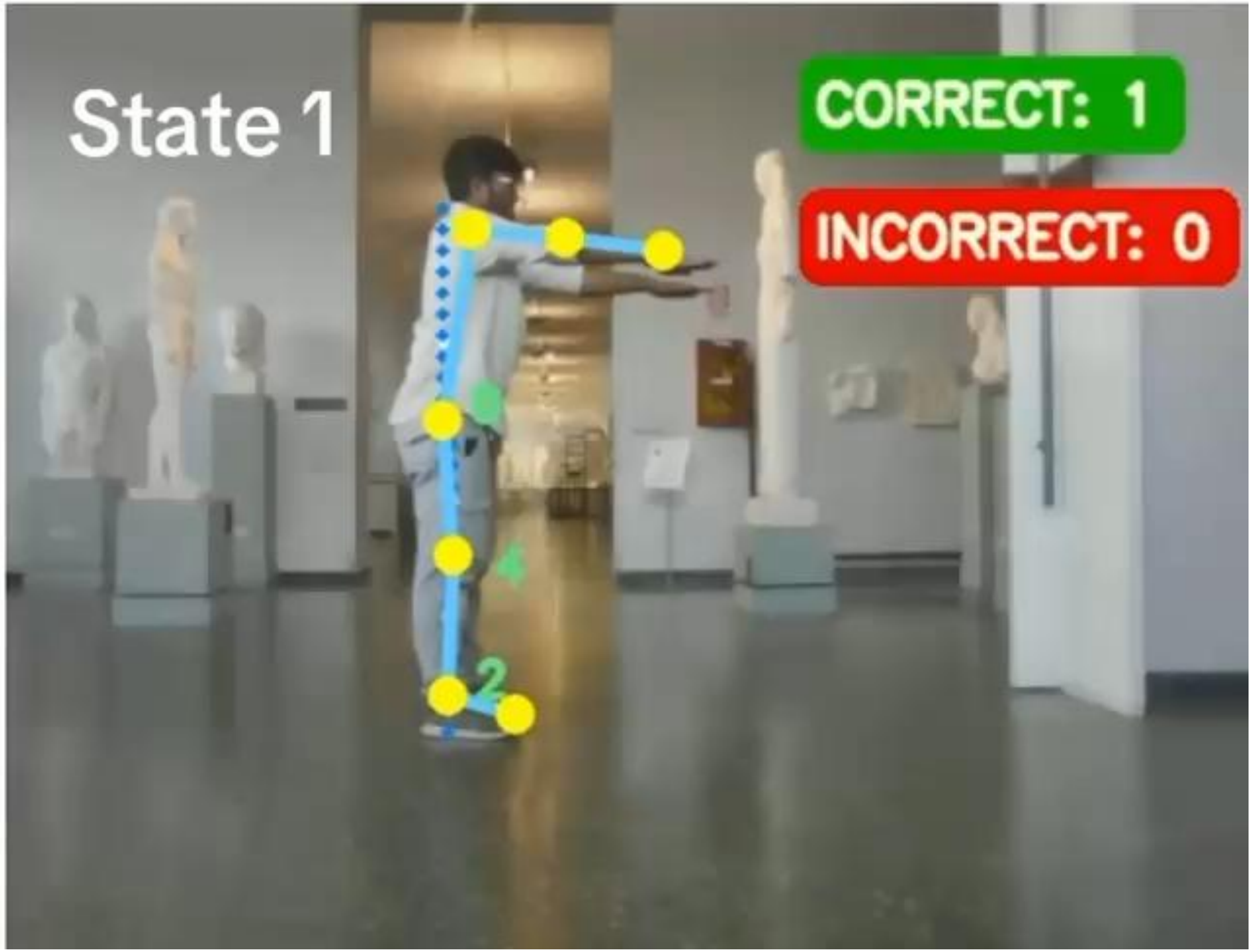


States:

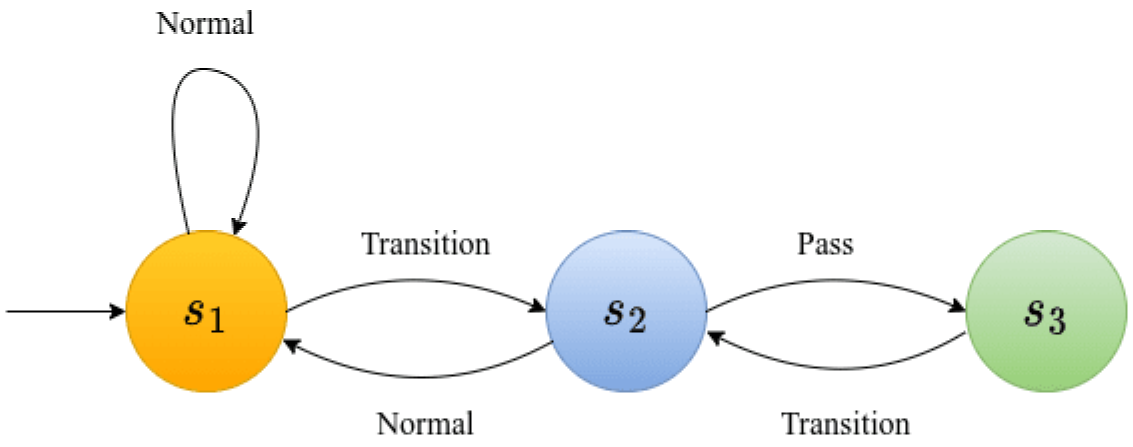
- **s1 (Standing):** If the angle between the knee and the vertical falls within **32°**, then it is in the **Normal** phase, and its state is **s1**. It is essentially the state where the counters for proper and improper squats are updated.
- **s2 (Mid-Squat):** If the angle between the knee and the vertical falls between **35°** and **65°**, it is in the **Transition** phase and subsequently goes to state **s2**.
- **s3 (Deep Squat** If the angle between the knee and the vertical lies within a specific range (say, between **75°** and **95°**), it is in the **Pass** phase and subsequently goes to state **s3**

- **Valid Rep**: s1 → s2 → s3 → s2 → s1
- **Early Termination**: s1 → s2 → s1 (counts as error)

AI Fitness Trainer: Squats Analysis (side view)



We can finally provide the state transition diagram as well.



- All calculations related to feedback are computed for states **s2** and **s3.**
- During our implementation, we maintain a list: **state_sequence**. It contains the series of states as the person goes from states **s1** through **s3** and back to **s1**. The maximum number of states in state_sequence is **3** (**[s2, s3, s2]**). This list determines whether a correct or an incorrect squat is performed.
- Once we encounter state **s1**, we re-initialize state_sequence to an empty list for subsequent squat counts.

E. Rep Counting System and Inactivity Handling

Correct Rep:

- Completes full state sequence [s2, s3, s2]
- No form errors detected
- Returns to s1 (standing)

Incorrect Rep:

- Any form error during movement
- Incomplete state sequence

Inactivity Handling

Resets counters after 15s (INACTIVE_THRESH) of:

- No poses detected **OR**
- No state changes (static pose)
- Offset angle> offset threshold (facing directly towards camera and when camera is not aligned properly)

F. Visual Feedback Pipeline

- **Frame Processing**:
  - Detect pose → Draw skeleton → Check alignment
- **Form Analysis**:
  - Calculate distances/angles → Detect errors → Update states
- **Feedback Generation**:
  - Show counters → Display errors → Handle inactivity

G. Real-Time Performance:

- Processes ~30 FPS on modest hardware
- Minimal lag between actual movement and feedback

Application Workflow and Thresholds

1. We first declare the following thresholds along with the two counters:
   - **STATE_THRESH**: A set of thresholds that determine the state that each frame belongs to.
   - **FEEDBACK_THRESH**: A set of thresholds to determine the feedback information that needs to be displayed.
   - **OFFSET_THRESH**: Threshold to determine if the person is facing directly toward the camera.
   - **INACTIVE_THRESH**: Threshold to determine inactivity, failing which the counters: CORRECT and INCORRECT are reset.
   - Counters: **CORRECT** and **INCORRECT** to count the number of proper and improper squats, respectively.

2. We read each frame from the webcam/video, pre-process it and pass it through **MediaPipe's Pose solution**

3. We then retrieve the desired landmarks for the Shoulders, Nose, Knee, Hip, and Ankle, provided the detection landmarks are available; else, we move on to compute **INACTIVE TIME** (in secs) when there are **no detections**.

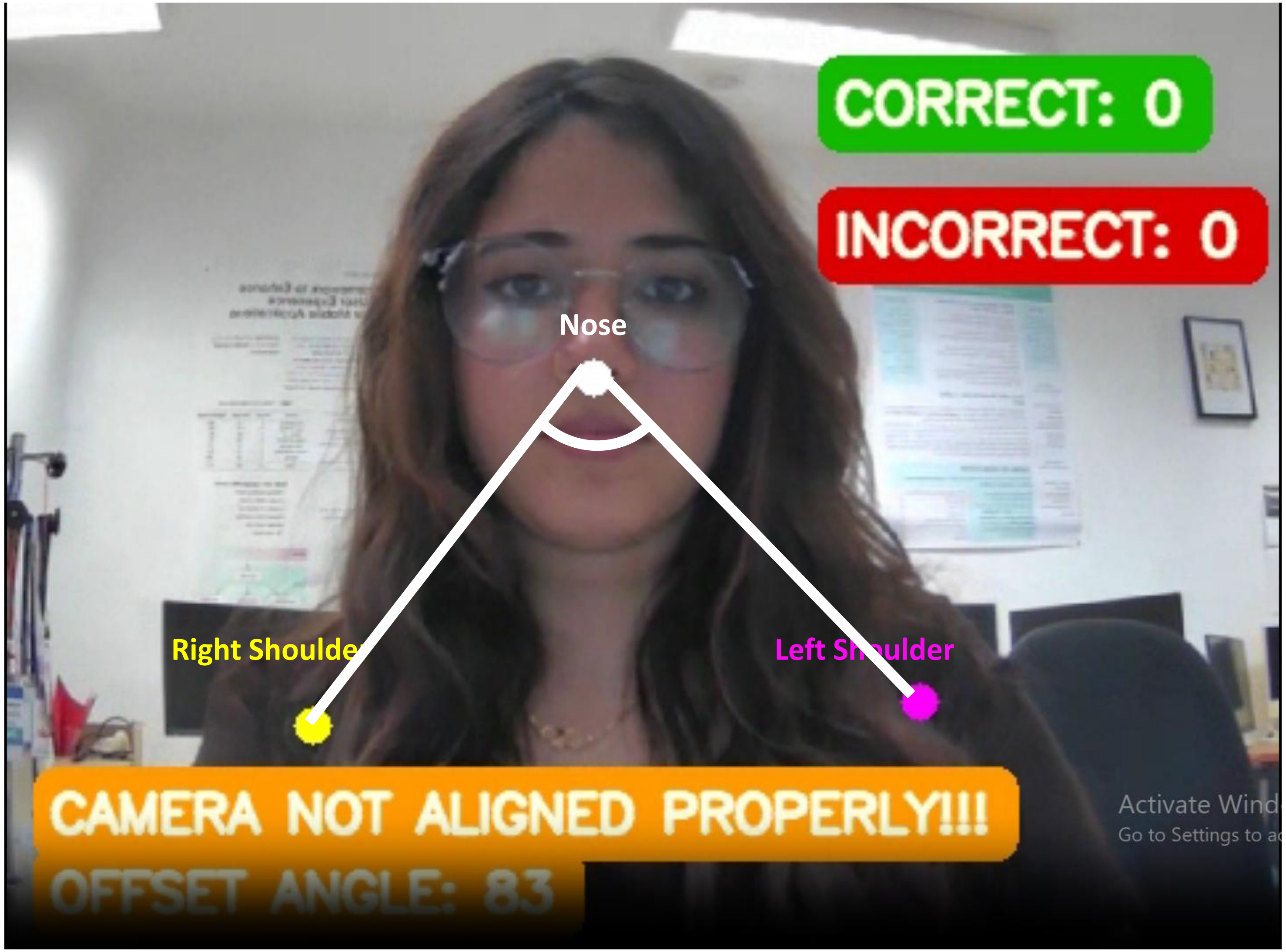
   - If this INACTIVE TIME passes the INACTIVE_THRESH, we reset the counters: CORRECT and INCORRECT.

4. The **offset angle** (discussed in the later section) is calculated for the Nose and Shoulder coordinates.

   - If the offset angle overshoots the OFFSET_THRESH, we display the appropriate warning and compute the INACTIVE TIME as discussed in **Step 3**

- 5. When the **offset angle** is within the OFFSET_THRESH, we go on to calculate the following:
    - The angles **shoulder-hip**, **hip-knee**, and **knee-ankle** lines with the verticals.
    - The **current_state** of the frame is calculated based on STATE_THRESH.
    - A list: **state_sequence** is maintained (discussed in the previous section).
- 6. When the current state is encountered as s1, we update the counters: CORRECT and INCORRECT based on the contents of state_sequence**.** Otherwise, we compute and display the feedback messages based on FEEDBACK_THRESH and compute the INACTIVE TIME.
- 7. We assign **prev_state** with **current_state** and proceed to fetch the subsequent frames.

For instance, to calculate the **offset angle**, we shall find the angle between the nose and the shoulders, with the coordinates of the nose being the reference point.

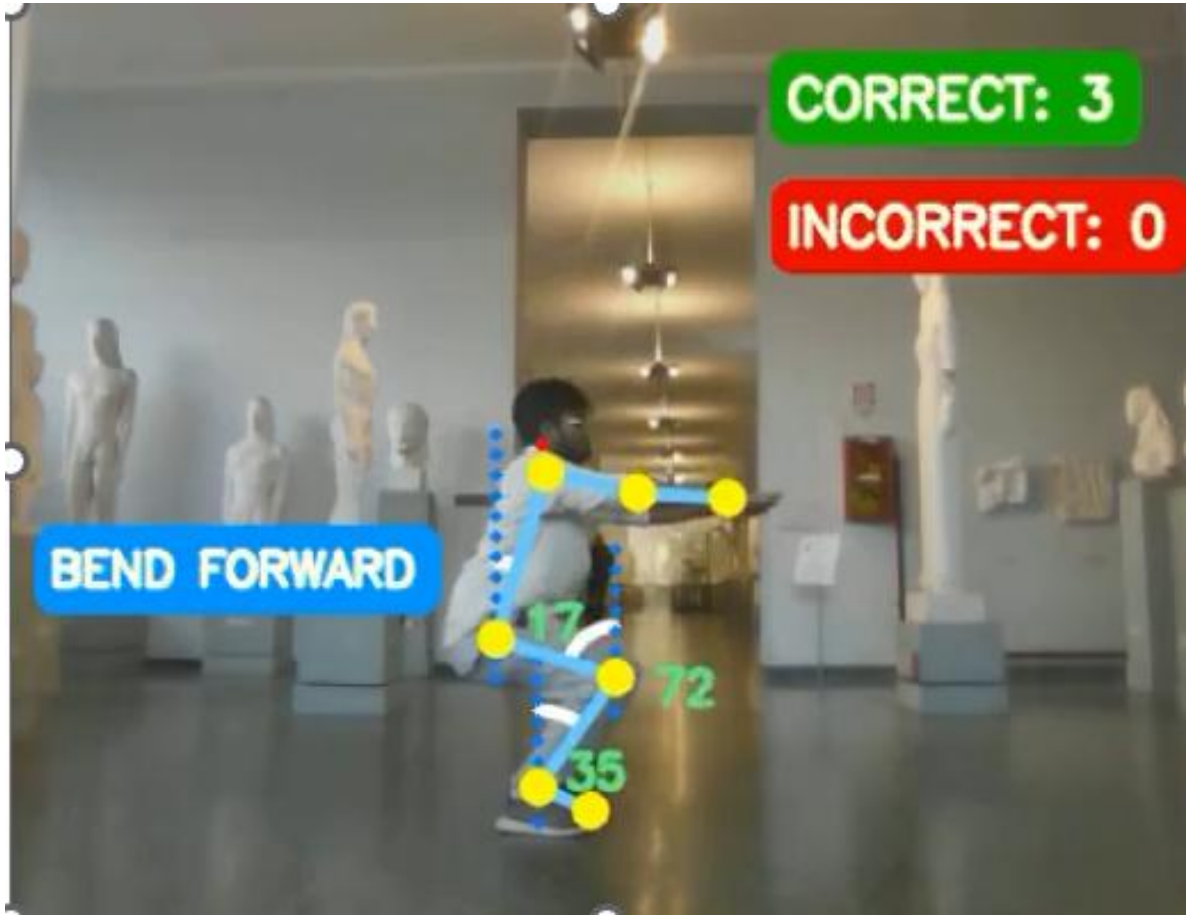


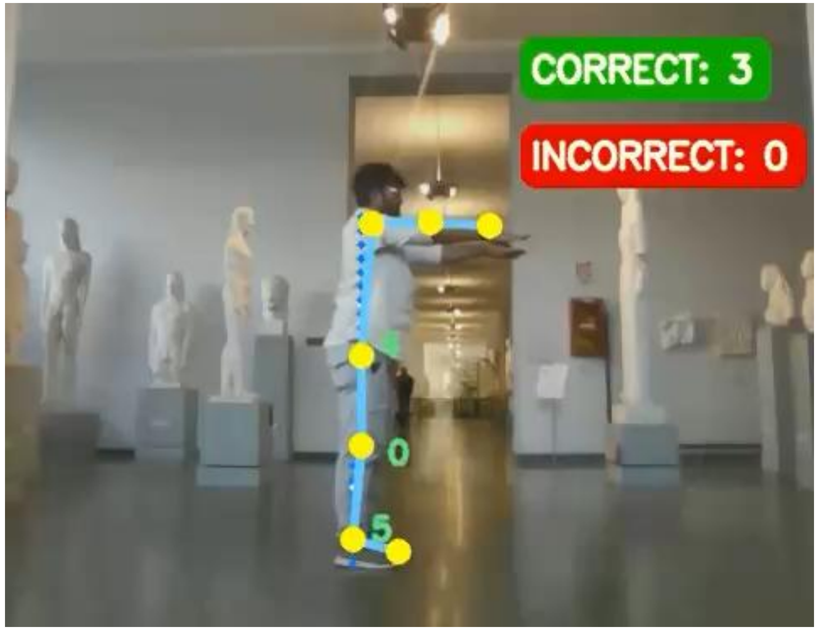
Feedback Actions

Our application shall provide five feedback messages while one performs a squat, namely:

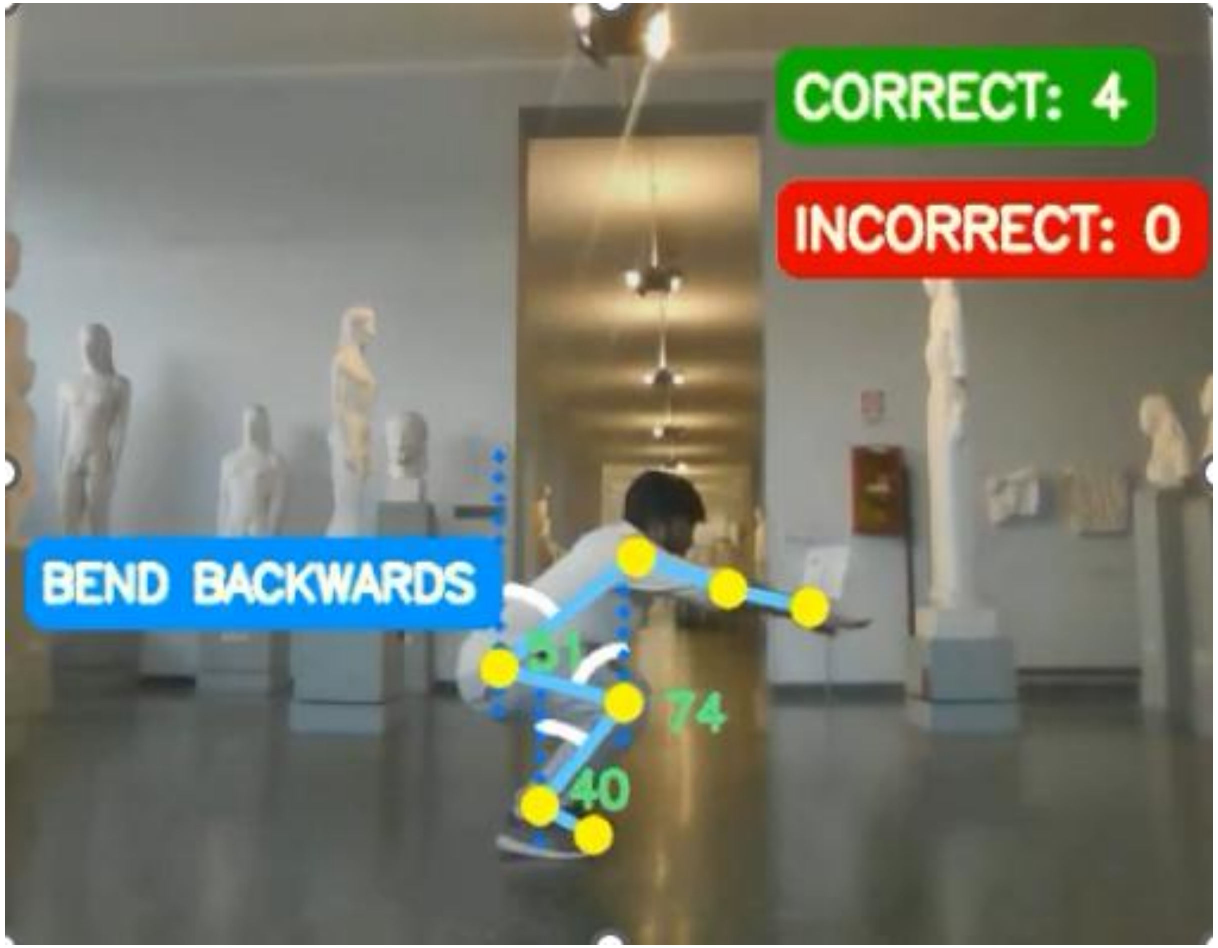
- Bend Forward
- Bend Backwards
- Lower one’s hips
- Knee falling over toes
- Deep squats

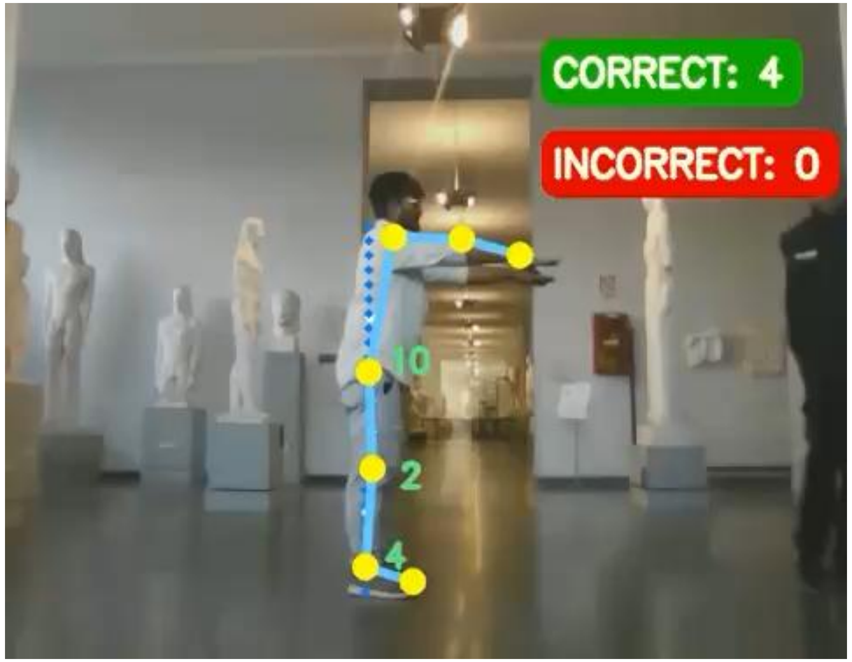
- **Feedback 1** "Bend Forward" is displayed when the hip-vertical angle (i.e., the angle between the shoulder-hip line with the vertical) falls **below a threshold**, for instance, 20°, as shown in the following figure.



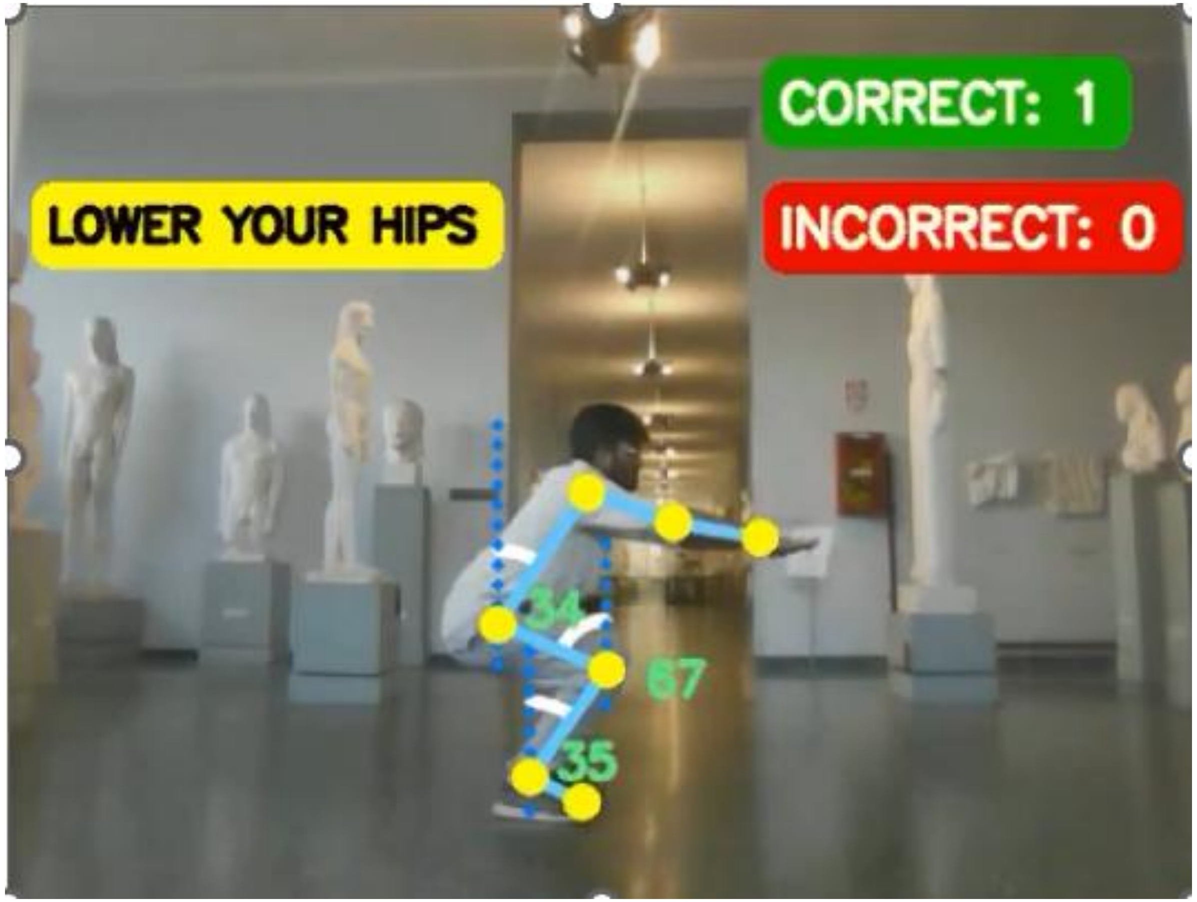


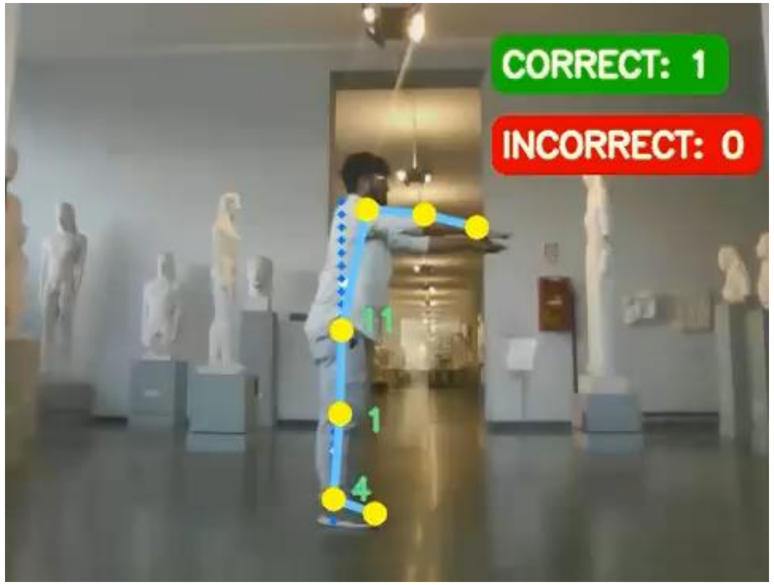
- **Feedback 2** "Bend Backwards" is displayed when the hip-vertical angle falls **above a threshold**, for instance, 45°, as shown below.



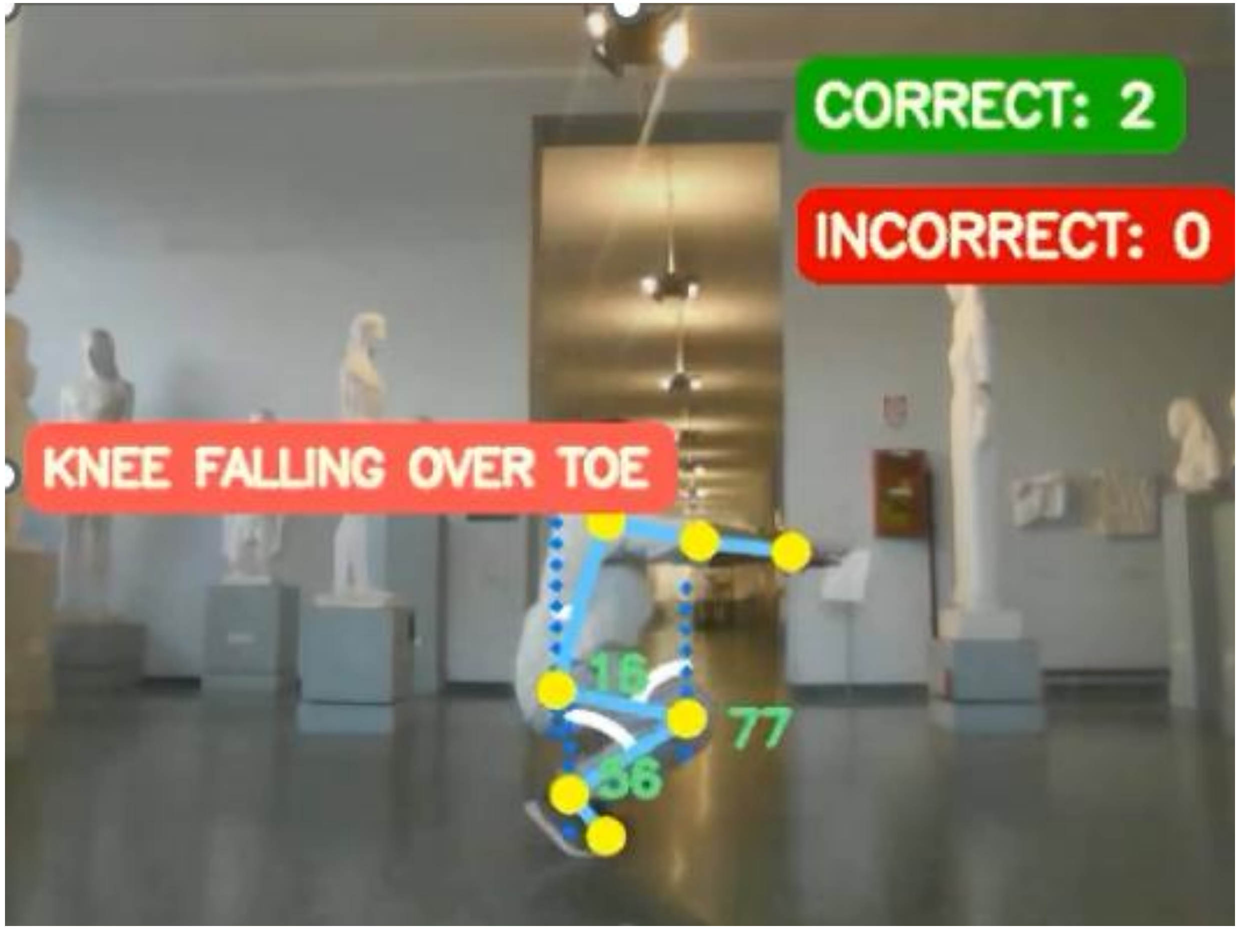


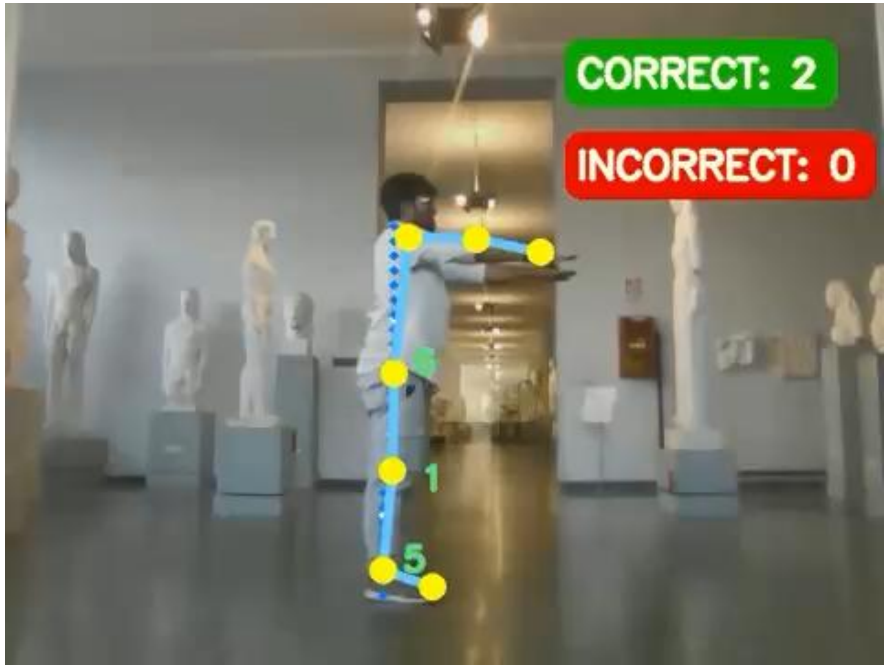
- **Feedback 3** "Lower one’s hips" is responsible when the angle between the hip-knee line with the vertical is **within thresholds**, say between 50° and 80°, as shown below.



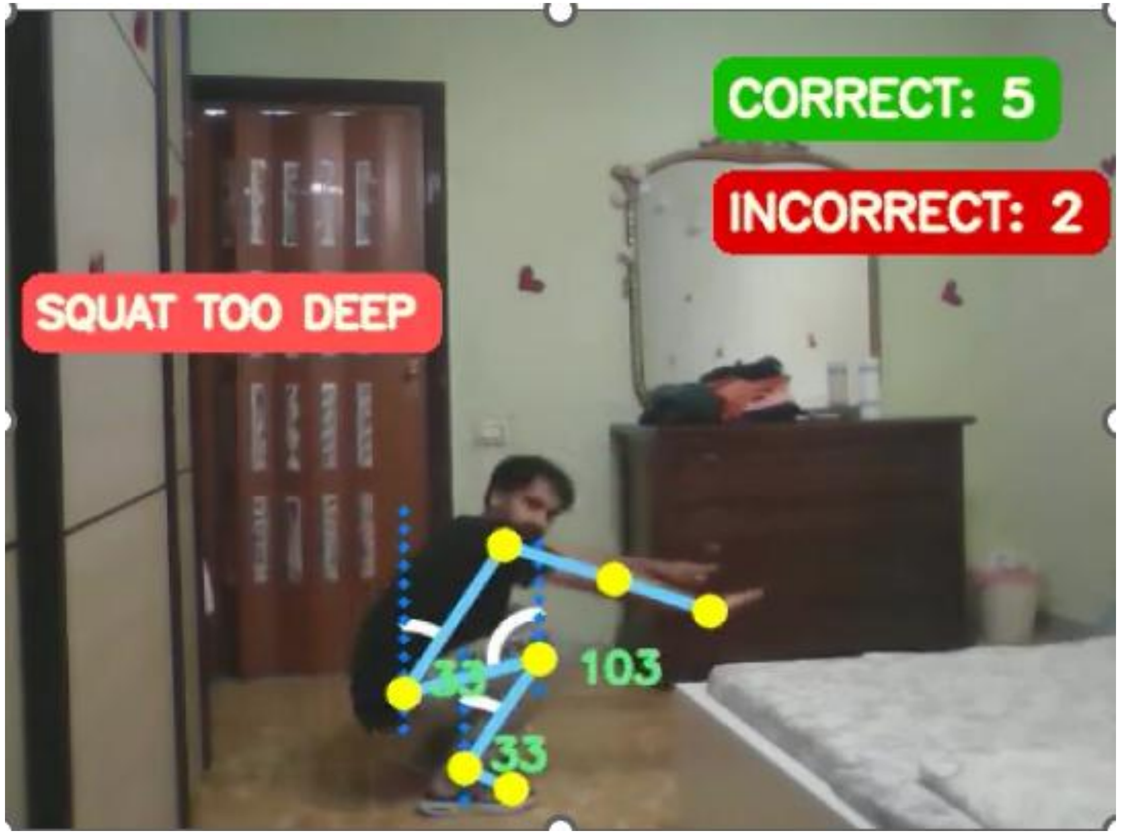


- **Feedback 4** “Knee falling over toes” is displayed when the angle between the knee-ankle line with the vertical lies above a threshold, for instance, 30°, as shown.





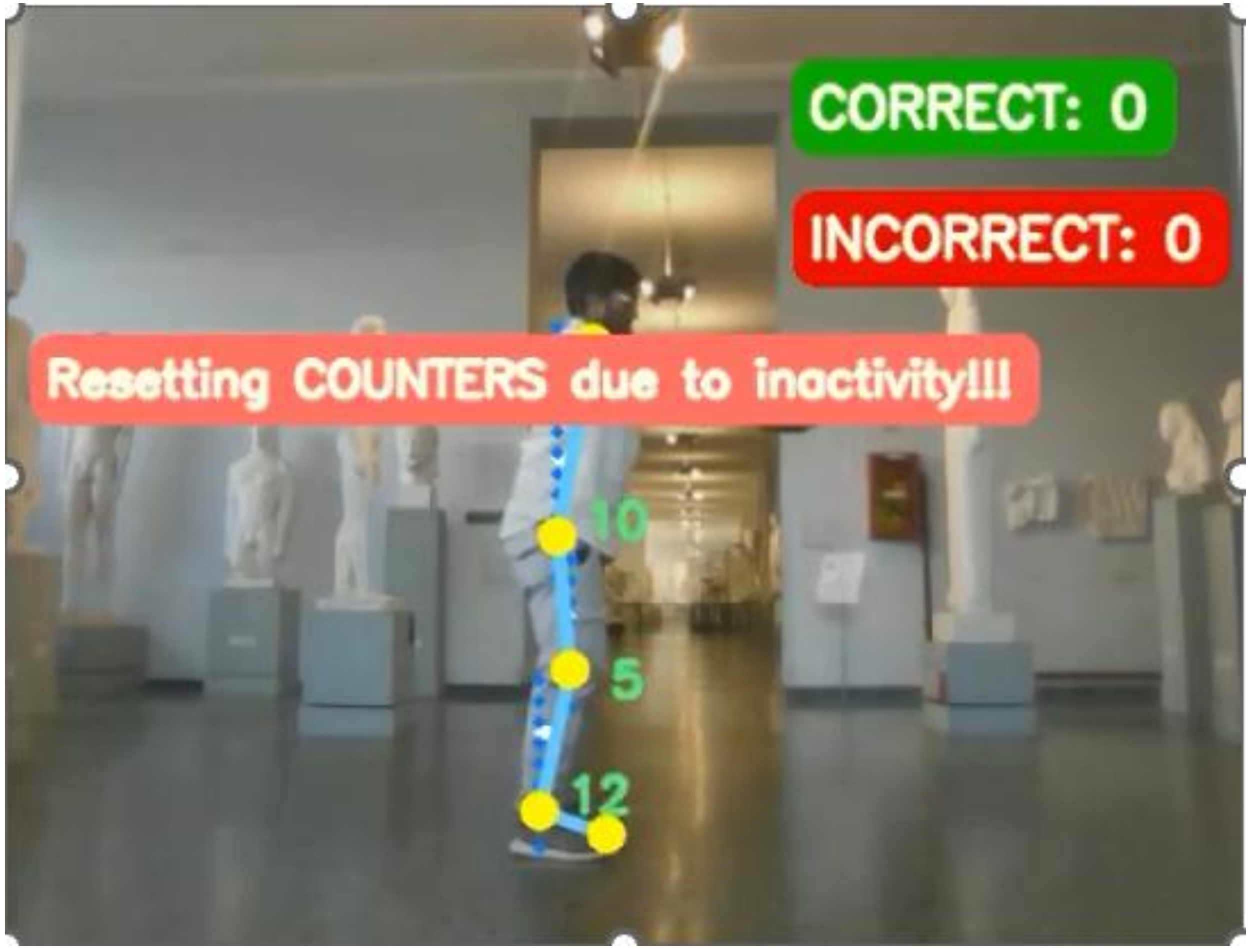
- **Feedback 5** "Deep squats" is displayed when the angle between the hip-knee line with the vertical; transitions through state **s3** and goes **beyond a threshold**, for instance, 95°.

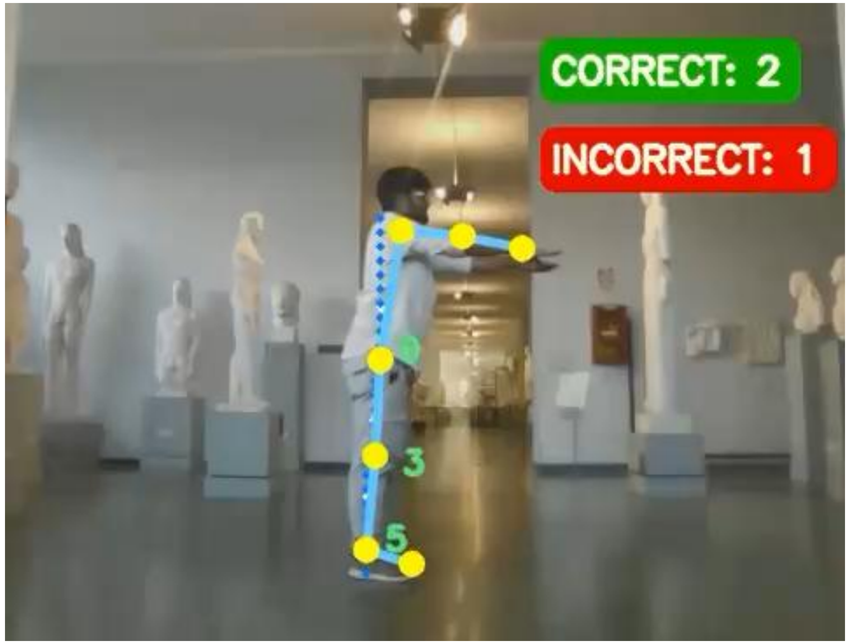


Computing Inactive Times

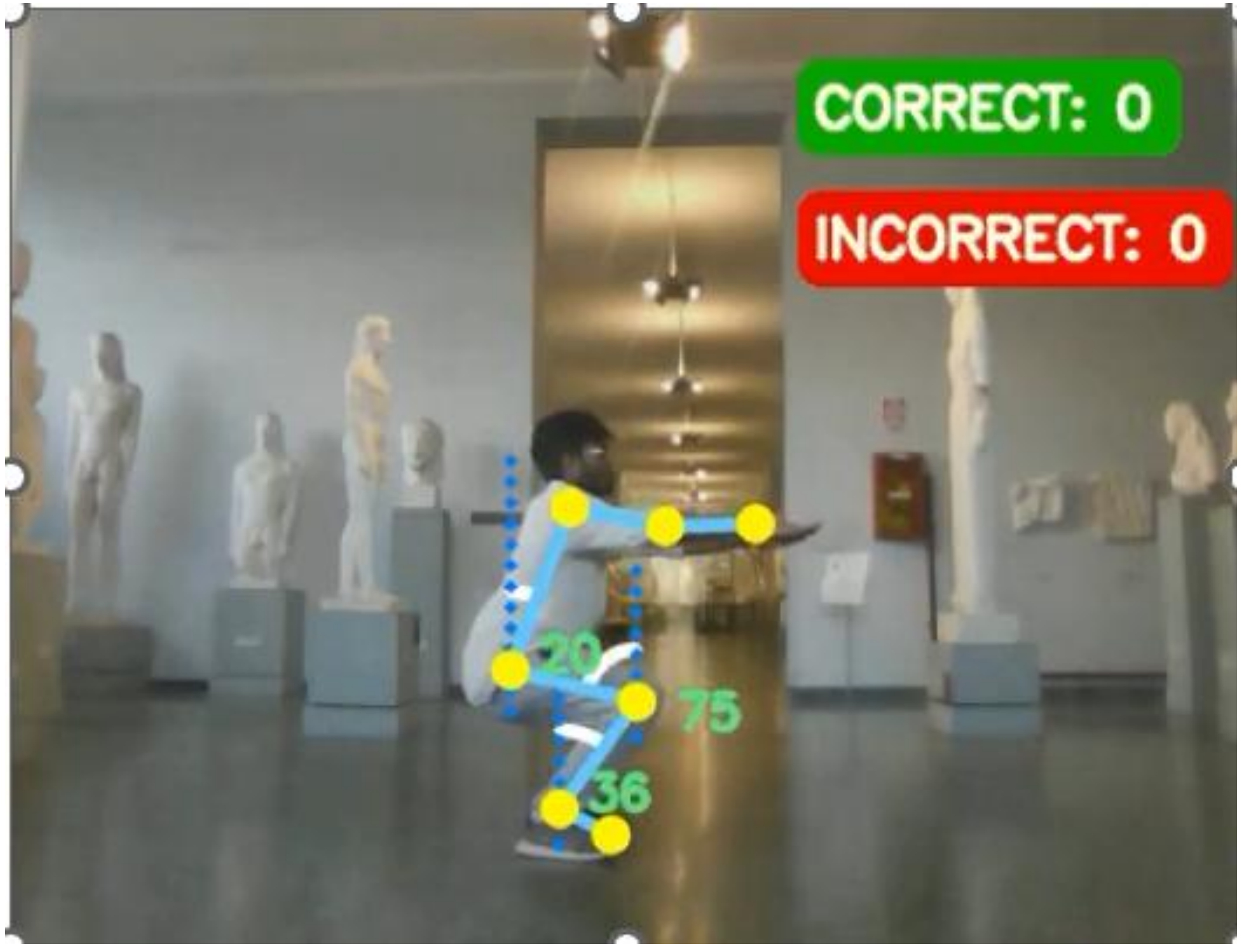
- As mentioned earlier, another aspect of our application is to reset all counters (for correct and incorrect squats) owing to inactivity. Inactivity is computed when an object (person) maintains a certain state beyond some **T** time. The threshold **T** is measured in seconds. We have set the value of **T** to **15** seconds.
- There are three situations when our application encounters inactivity:
- We are facing directly toward the camera (i.e., **offset angle** > **OFFSET_THRESH**) beyond **T** seconds.
- The state of the person remains unchanged beyond **T** seconds.
- There are no detections beyond **T** seconds.

Computing Inactive Times



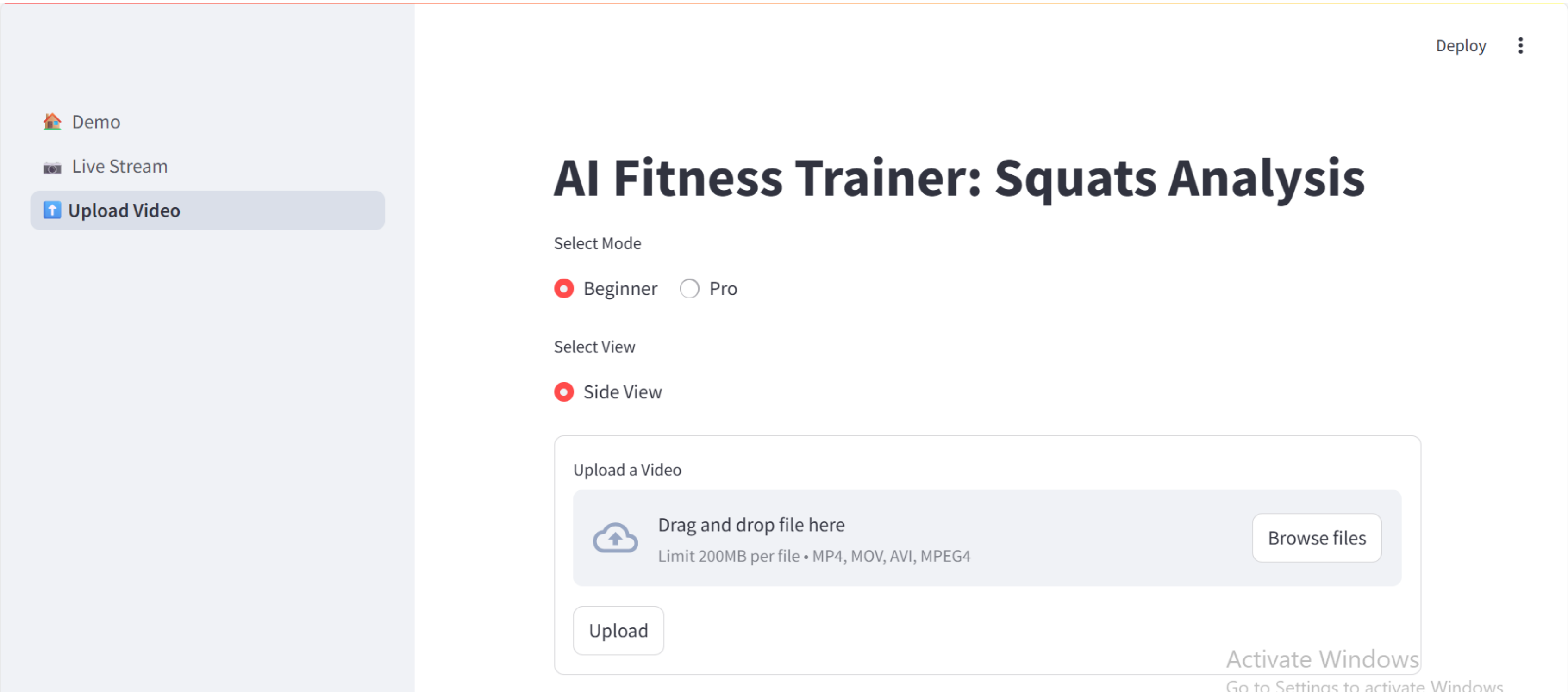


Correct Squats



Upload Section

- The **Upload Section** allows users to **upload a pre-recorded workout video** (such as squats) for **pose analysis and feedback** using AI-powered posture detection.



AI Fitness Trainer: Squats Analysis

Select Mode

● Beginner ○ Pro

Select View

● Side View

Upload a Video

Drag and drop file here

Limit 200MB per file • MP4, MOV, AVI, MPEG4

Key Points – Upload Section (AI Fitness Trainer)

📁 **Upload Feature:**
Allows users to upload pre-recorded workout videos (.mp4, .mov, .avi) for posture analysis.

🧠 **AI-Powered Pose Detection:**
Each video frame is analyzed using MediaPipe to detect human body keypoints.

🎯 **Mode & View Selection:**
Users can choose "Beginner" or "Pro" mode and "Side View" (Front View can be added later).

📊 **Posture Evaluation:**
Squat form is checked against predefined angle thresholds to detect errors or provide feedback.

🖼️ **Frame-by-Frame Display:**
Each processed frame with visual feedback is shown live inside the app.

💾 **AI-Processed Output:**
The processed video is saved with overlays indicating correctness of the exercise.

⬇️ **Downloadable Feedback:**
Users can download the final video for personal review or coaching use.

AI Fitness Trainer: Squats Analysis (Application Reviews)

Name and surname	Email	Occupation/ Role	How easy was it to use this application (1 - very difficult –5 very easy)	How accurate did you find the posture detection and feedback (1-5)	Did you find any technical issue while using this tool? (yes/no/somewhat)	Was the real time feedback helpful in improving your squat form? (yes/no somewhat)	How would you rate the user interface and visual clarity? (Linear scale 1-5)	What did you like most about the interface ?	What additional feature would you like to see in the future?
Azmeera Qureshi	Azmeeraqureshi@gmail.com	Masters student	5	5	no	yes	5	Live streaming	More exercises should be added
Lorenzo	Antonelli@di.uniroma1.it	Masters student	5	5	no	yes	5	Real time feedback	Frontal view and also a sound feedback
Marco	realacci@di.uniroma.it	Masters student	5	5	no	yes	5	Real time feedback	Sound feedback
Ismail Hossain Sohan	Sohanh621@gmail.com	Masters Student	5	5	no	yes	5	Accuracy	Frontal view
Shrikant verma	Datla@di.uniroma1.it	Phd scholar	5	5	yes	yes	4	Live stream and the feedback system	Even bigger feedback or textual
palma	Palma.2043253@studenti.uniroma1	Masters student	5	4	no	yes	5 for live stream and for 4 for upload	Simplicity of the interface like everything	More bigger feedbacks in upload video section