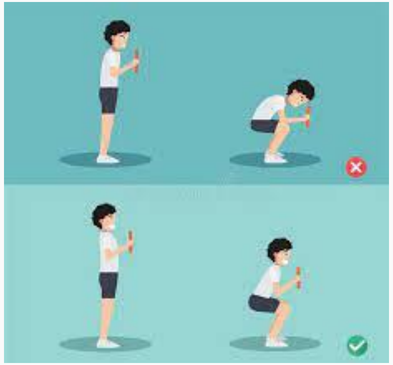
Exercise Pose Recognition

By Connor Shyan & Lucas Jang

Motivation & Goal

Motivation



Exercising on your own can sometimes lead to improper postures and ineffective workouts with nobody to check your form.

Goal

- Need something to check if the user's exercise postures are correct and proper
- Seeing feedback on UI to help users check on their form and recognize if they are doing the right exercise

Solution: An application that can track the user's body through laptop webcam and show the user if they are doing the exercise properly or not, along with if the user is going up or down for the exercise.

General Design

- Two different exercise poses:
  - Pushups
  - Squats
- Two different statuses:
  - Up
  - Down
- Approach #1:
  - Detect and calculate the angles between the user’s joints
  - Use the joint angles to determine the current status of the user’s pose
- Approach #2:
  - Train a ML model from images for the two exercise poses for each status
  - Use the trained model to predict the current status of the user’s pose

Libraries Used

For both:

- OpenCV (To import images, capture webcam video, process video frames into RGB images)
- MediaPipe Pose (To detect the person/pose in the frame, predict pose landmarks and segmentation mask)

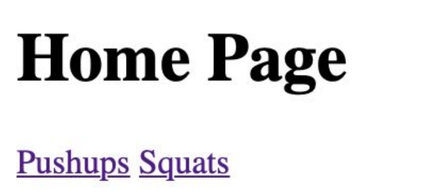
For Approach #1:

- NumPy (To assist with calculating angles between the joint landmarks)
- Flask (To give a selection screen between the two exercises to the users)

For Approach #2:

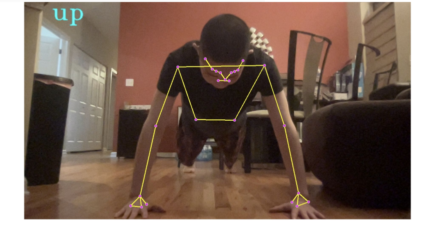
- pandas (To export training data to csv files, predict exercise poses and analyze data)
- scikit-learn (To train the prediction models using the exported csv file through machine learning)

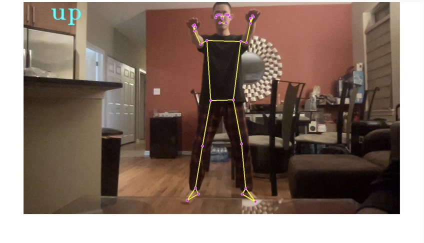
Implementation (Approach #1)

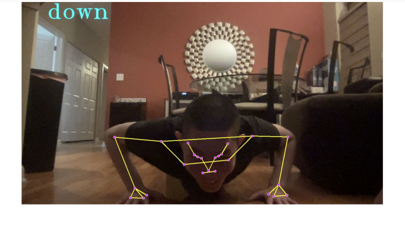


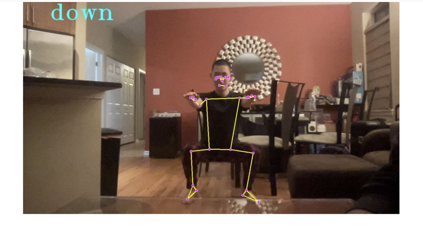
Home Page

Pushups Squats







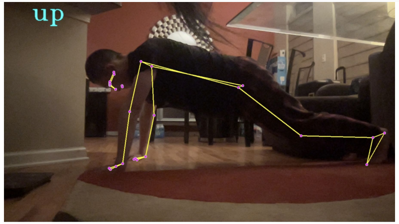


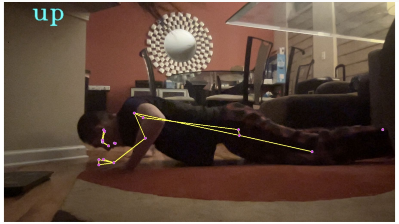
Process

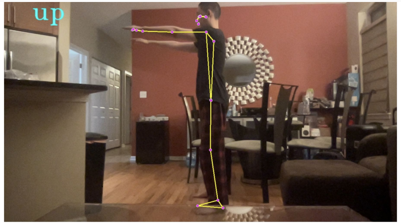
- Users select which exercise to track
- UI Window appears for selected exercise
- Video frame of user is converted to a RGB image and processed using OpenCV
- User’s pose landmarks are extracted from the RGB image using MediaPipe Pose
- Angles between certain pose landmarks are calculated with the help of NumPy
- Calculated angles are used to estimate the user’s exercise status and shown on the UI

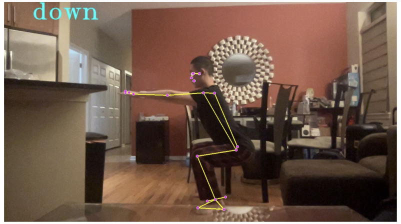
Result (Approach #1)

- “Up” and “Down” status for pushups recognized properly:
  - For front view
  - Not for side view
- “Up” and “Down” status for squats recognized properly:
  - For front view
  - For side view
- Recognition for both exercises from the diagonal view is unreliable

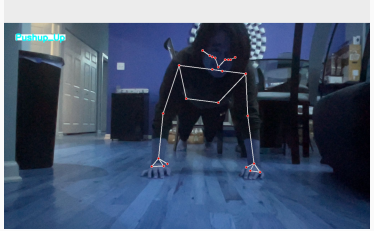


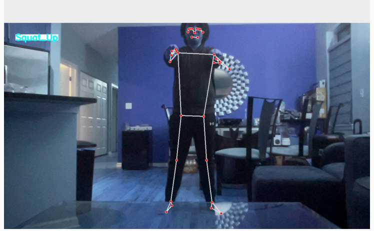


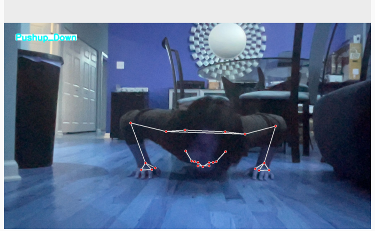


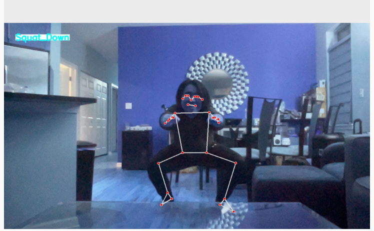


Implementation (Approach #2)







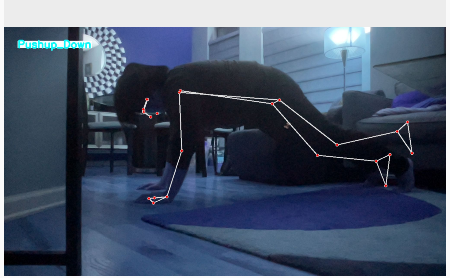


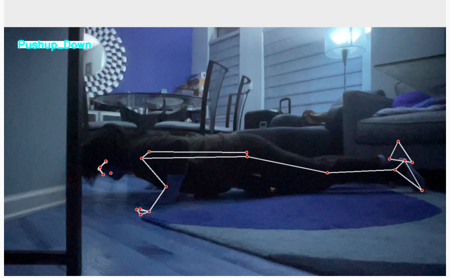
Process

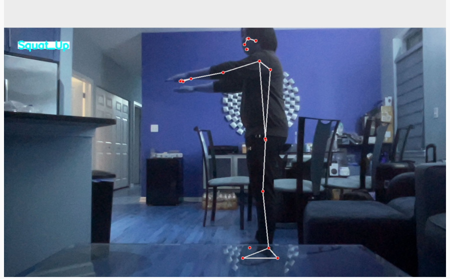
- 20 different images gathered for each exercise for each status (80 images total)
- Images have pose landmark data extracted from them into a csv file using pandas
- The csv file is used to train the prediction model using scikit-learn
- A single UI window opens up and pose landmark data extracted from video frames
- The landmark data is used with trained model to predict both the user's current exercise pose and status, which is shown on UI

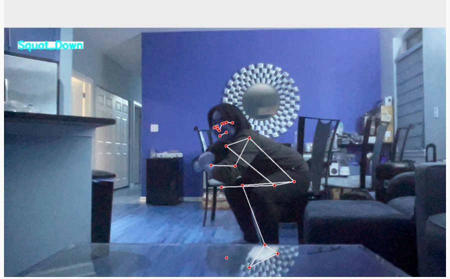
Result (Approach #2)

- Both the exercise poses and the status recognized properly for front view
- For the side view:
  - Pushups and its status have some difficulty being recognized properly
  - Squats and its status can be recognized, but very specific
- Recognition for both exercises and status from the diagonal view is more reliable than Approach #1, but not so much









Conclusion

- Approach #1 seems to be more reliable in recognition at the moment
    - Possibly due to confidence level (joint angles) for pose recognition not being too strict
- Approach #2 could become a lot more reliable and better than Approach #1 given the right dataset:
    - Larger datasets on kaggle.com and such either had missing details or did not separate “Up” and “Down”
    - No need to switch modes/windows to detect different exercise poses
    - More exercise types can be added easily through just training with dataset
    - Recognition more reliable from every angle with the right variety in dataset
- More improvements can be made to the app for better usability:
    - Usage of external webcam for better quality video and positioning
    - Additions to the UI such as exercise counter, “difficulty” sliders, etc…

Thank You!