

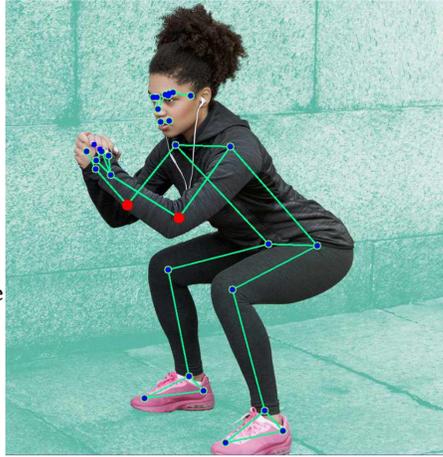
Introduction

The aim of this project is to create a program that could view a squat and give feedback to the user about their form.

To achieve this, we used OpenCV, a computer vision package for python that allowed us to use some common webcams to capture and manipulate frames.

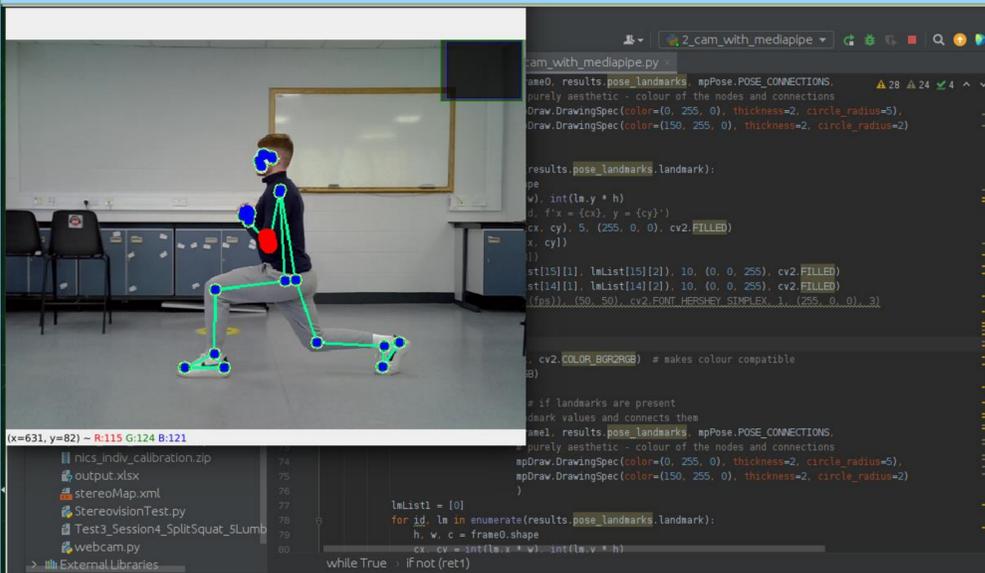
The Python package MediaPipe was used and allowed us to map a model onto a human body in each frame. with OpenCV, we can show these models to the user in real time.

Using both of these packages together is the basis for this project. They provided the tool to analyse the user's movement in enough detail that the program could provide feedback in real time.



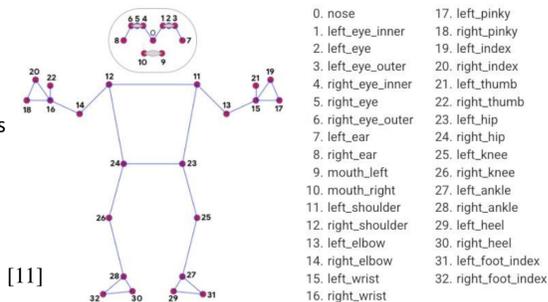
[10]

What does OpenCV and MediaPipe do?



OpenCV allows the program to load up videos or live web cameras in our case. It allows us to go through the video feed frame by frame, draw the skeletal outline over the person on each individual frame, then show this frame on screen.

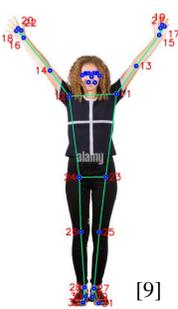
MediaPipe allowed us to gather the coordinate data for the core "landmarks" of the body. Using OpenCV, the program draws circles on all of these landmarks and connects them in an appropriate way as shown in the diagram to the right.



[11]

The program repeats this process of placing a skeletal overlay on the body for each individual frame before that frame is shown to the screen.

On a Jetson Nano, this is resource intensive, and the program will deliver a low frame rate. On a desktop PC however, the frame rate is not a problem, and the program runs seamlessly in real time.



[9]

What is analysis of squat form

Deciding what the system would look for to determine what a good squat was a somewhat under looked task for us. We settled on focusing on the Bulgarian split squat as that would be easier for our camera setup to capture fully. We used guidelines from a paper in the NLM [1] as a basis on what an ideal split squat should be. This informed us that the three crucial joints that determine a successful split squat are the angle of the ankle, knee and the hips.



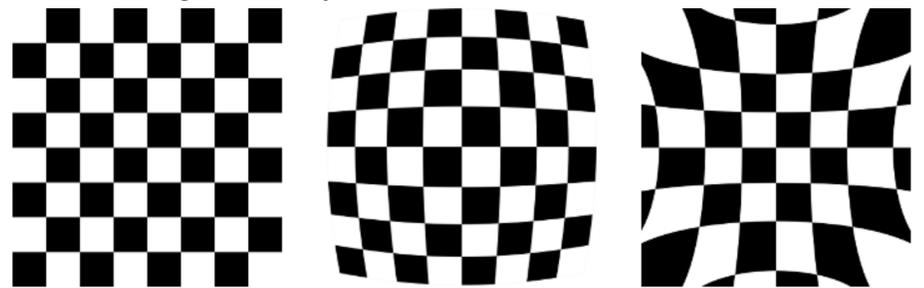
[4]

Objectives

1. Have the program recognise a person and map them
2. Have it perform real time calculations based on the user's squat form
3. Add a second camera that is deferred to if it has better sight of the joint
4. Camera calibration for more accurate computer vision
5. Give feedback to the user

What was achieved:

- 1 A script using OpenCV and MediaPipe was created to map the user
- 2 A number of crucial angles were obtained from the NLM [1], and functions were added to the script which calculated the angles.
- 3 One camera faces the user, and the other camera is to the side. These different angles have better views of different joints, so a function was created to compare the two camera angles confidence that it can see each individual joint. The camera with the higher confidence is what's used for the calculation.
- 4 Calibrating the camera with a calibration script (OpenCV) and a checkerboard pattern allows for the camera to remove distortions native to the camera's hardware and allow for more accurate images to be analysed.



No distortion

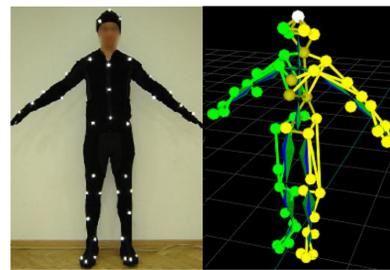
[3]

Negative radial distortion (Barrel distortion)

Positive radial distortion (Pincushion distortion)

- 5 The data is printed to the terminal and saved to an excel file.

Trade offs



[5]

The system that MediaPipe uses to detect people is a pose estimating AI. This means it doesn't truly track all the landmarks on your body but instead places the points on you where the AI believes they should be.

This MediaPipe system isn't as accurate as an autocorrelation function system, but it removes almost all the barriers to use and makes it very appealing. There is no required setup of placing coloured dots or specific shapes on the user as markers,

Results

At the end of our time on this project we had met most of our goals, it could see the user and map them while calculating the angles their joints made while squatting. It saves this information into an excel file.

The system allows for calibration of the cameras, which is particularly useful if the system was set up for long-term use as it has to be calibrated every time the cameras are moved. The system is decently accurate as it uses the best view of the two cameras to calculate each angle.

If there was more time to work on the project, adding a time stamp to each collection of biometric data would help hugely with processing it and providing feedback

	Left Elbow	Right Elbow	Left Knee	Right Knee	Left Hip	Right Hip
1	127.6	162.3	134.3	110.6	134.7	160.6
2	170.8	174.8	73.2	143.7	165.2	165.1
3	105.6	156.3	142.3	22.9	104.6	151.5
4	117.1	151.1	135.9	69.1	129.2	150.6
5	138.0	161.6	148.6	46.3	147.8	44.0
6	135.6	120.2	174.6	46.3	131.4	122.0
7	135.6	120.2	129.7	46.3	131.4	122.0
8	172.5	148.1	86.9	53.6	170.6	159.5
9	176.8	158.8	134.0	40.5	163.2	174.4
10	179.2	158.7	145.5	43.8	177.5	168.8
11	125.6	170.7	120.8	156.7	142.1	171.6
12	128.2	174.1	127.9	143.3	130.6	158.2
13	178.8	174.1	117.8	64.6	174.8	158.2
14	146.5	143.2	165.9	57.0	164.5	157.2
15	176.0	163.1	135.9	65.6	165.4	175.6
16	178.0	164.3	140.2	116.2	178.9	171.6
17	170.3	168.7	119.4	112.3	176.8	174.3
18	136.0	113.9	128.7	175.6	142.3	94.8
19	145.0	149.0	141.8	166.7	169.4	91.1
20	156.7	103.0	172.4	161.5	110.4	57.1
21	177.3	164.7	129.0	77.4	67.8	163.3
22	172.7	143.0	105.4	64.3	176.9	158.8
23	175.1	162.3	136.0	66.9	165.3	175.6
24	132.7	164.7	136.7	112.5	150.1	175.6
25	151.2	173.4	123.1	142.9	172.6	165.4
26	131.2	101.2	148.1	177.1	148.5	82.6
27	168.4	92.4	131.1	159.9	138.2	59.9
28	150.0	84.4	156.4	172.7	80.5	83.3
29	112.7	46.7	170.0	148.4	64.5	50.4

Acknowledgments

- [1] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8136570/>
- [2] https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [3] https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html
- [4] <https://experiencelife.lifetime.life/article/break-it-down-how-to-do-the-bulgarian-split-squat/>
- [5] <https://www.mdpi.com/1424-8220/21/18/6115>
- [6] <https://www.youtube.com/watch?v=l8SBOV1dtYo>
- [7] <https://stackoverflow.com/questions/13437727/how-to-write-to-an-excel-spreadsheet-using-python>
- [8] <https://pythonbasics.org/write-excel/>
- [9] <https://www.alamy.com/stock-photo/open-arms-isolated-full-length.html?sortBy=relevant>
- [10] <https://www.shape.com/fitness/workouts/butt-workouts/30-day-squat-challenge>
- [11] <https://github.com/google/mediapipe/blob/master/docs/solutions/pose.md>
- [12] <https://www.geeksforgeeks.org/camera-calibration-with-python-opencv/>