Classification of Sitting Postures using MediaPipe and Machine Learning

Xin Ying Chew

***Abstract*** *— Human pose detection is becoming more prevalent in our daily lives, ranging from healthcare to workout trainers. And as people spend more time sitting down, many people are developing bad sitting postures. With recent developments in machine learning and computer vision, the correction for bad sitting posture can be easily achieved without physical exercises through real-time computer supervision. In this paper, I propose a machine learning model based on MediaPipe that eliminates the need for the entire human torso to be visible in the frame. The model takes on frame sequences from the desktop camera and classifies the sitting posture of the human in the frame. An accuracy of 100% was achieved for sitting posture recognition when tested using the testing data after repeated retraining.*

# INTRODUCTION

Bad postures have been a problem since the past when we labor over house chores and fieldwork. Now, technological advancements have led to the development of mobile phones and iPads, giving us another reason for hunching over by drawing us to tower over our phones. This bad habit of hunching can be carried over to times when we are simply sitting in class or talking with our friends. Children may not realize the severity of bad postures or even the fact that they are not sitting properly. Kids and adults alike may prefer slouching over sitting straight as it requires less energy. However, hunching and slouching can cause back pain, neck pain, headache, disrupt sleep and digestion, and decrease motivation [1]. Therefore, the long-term health damage of bad postures is worse than the short-term comfort it provides and it is an issue that we must confront.

As our daily life is increasingly dependent on sitting work, many have adopted bad sitting postures. While there are existing models for predicting human postures, most of them require the entire human to be visible in the frame [2], which makes them not as user-friendly because not everyone has a separate webcam or a large enough space to be able to capture their entire body in their office or home. Thus, this project aims to develop a machine learning algorithm that can determine bad sitting postures using images of the user’s upper half body, simply obtained through a desktop camera. Specifically, I will be aiming to classify four different kinds of sitting postures — straight, hunch, slouch, and slant — with straight defined as sitting upright, hunch as leaning forward, slouch as leaning backward, and slant as leaning sideways.

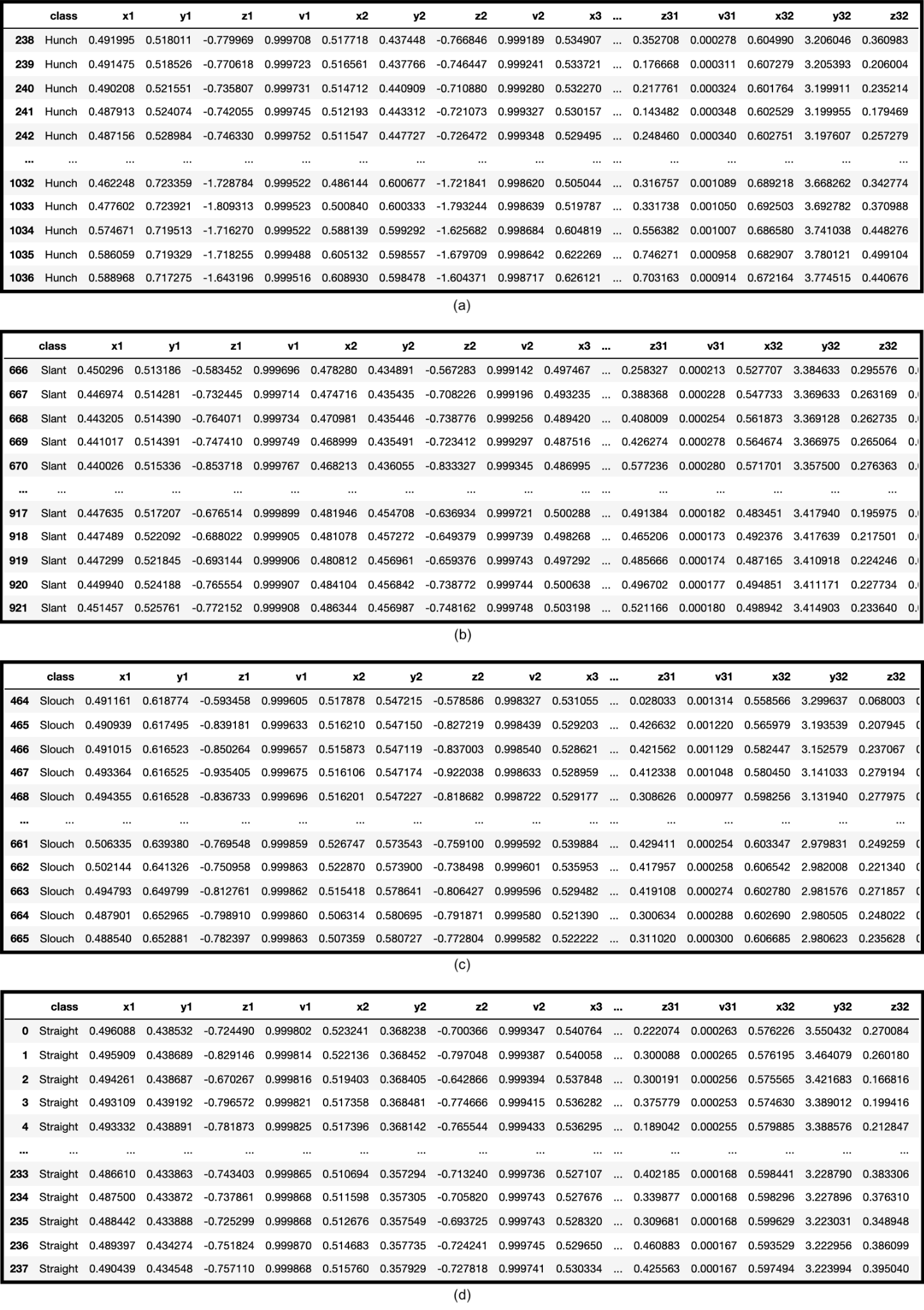
## A. Data Collection, Description, and Visualization:

As aforementioned, the raw data inputs are the BGR (blue-green-red) images [3] from the built-in camera on the user’s desktop through OpenCV. Using the MediaPipe Holistic Pipeline, which utilizes the pose, face, and hand landmark models to generate a total of 543 landmarks [4], the person captured by the camera will be labeled from the tip of their nose, all the way down to the heel of their foot.

MediaPipe is extremely helpful in that it outputs a floating-point number for the x, y, z coordinate, and visibility (the likelihood of the landmark being visible in the image) of each landmark [4]. Compared to the BGR color values of each pixel, these coordinates provide more in-depth information on the person’s body position and allow for a more accurate model in detecting different poses.

While the MediaPipe Holistic Pipeline includes the face and hand landmark models as well, only the pose landmark model was ultimately used. This is because the pose landmark model gives a general location for the palm and important facial features (such as the mouth, nose, and eye) already and detailed tracking of the hands and facial features is unnecessary for pose detection. As a result, 33\*4 = 132 data points were collected for each sample, with the 33 landmarks from the pose landmark model each having 4 values: the x, y, z coordinates, and visibility [4].

Combining OpenCV and the MediaPipe Holistic Pipeline, the 132 data are collected for each image of a class and written into a CSV file. Fig. 1 illustrates some of the data collected for each class, with x1 being the x-coordinate of the 1st landmark (the nose), y2 the y-coordinate of the 2nd landmark (the inner left eye), z3 the z-coordinate of the 3rd landmark (left eye), and v33 the visibility of the last landmark (the right foot index) [5], etc.

****Fig. 1. Sample data for the classes “Hunch” (a), “Slant” (b), “Slouch (c), and “Straight” (d).

In order to visualize the data, the x average was calculated using:

X avg = (x1 + x2 + x3 + … + x21 + x22) ÷ 22,

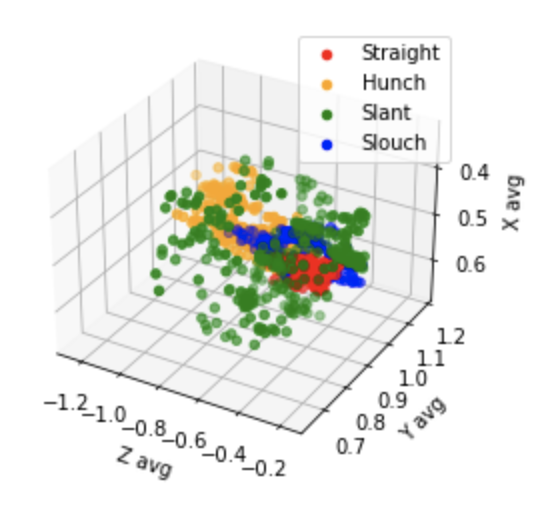
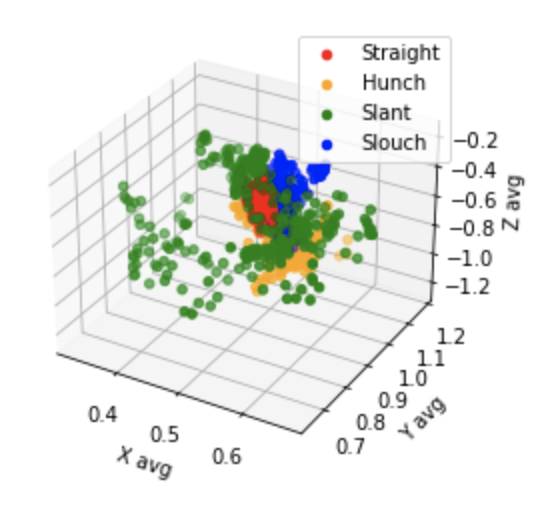
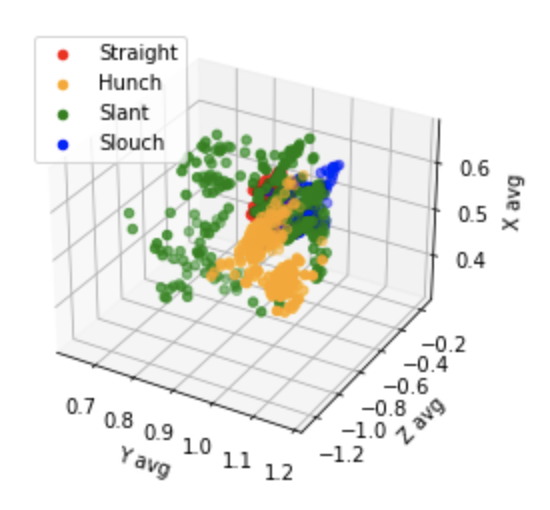
the y average calculated using:

Y avg = (y1 + y2 + y3 + … + y21 + y22) ÷ 22,

and the z average calculated using:

Z avg = (z1 + z2 + z3 + … + z21 + z22) ÷ 22.

The landmarks 23 to 33 are excluded because these are the landmarks that constitute the bottom half of our body (from hip to toe) and are not visible in the camera, thus, do not provide important information about the user’s postures.



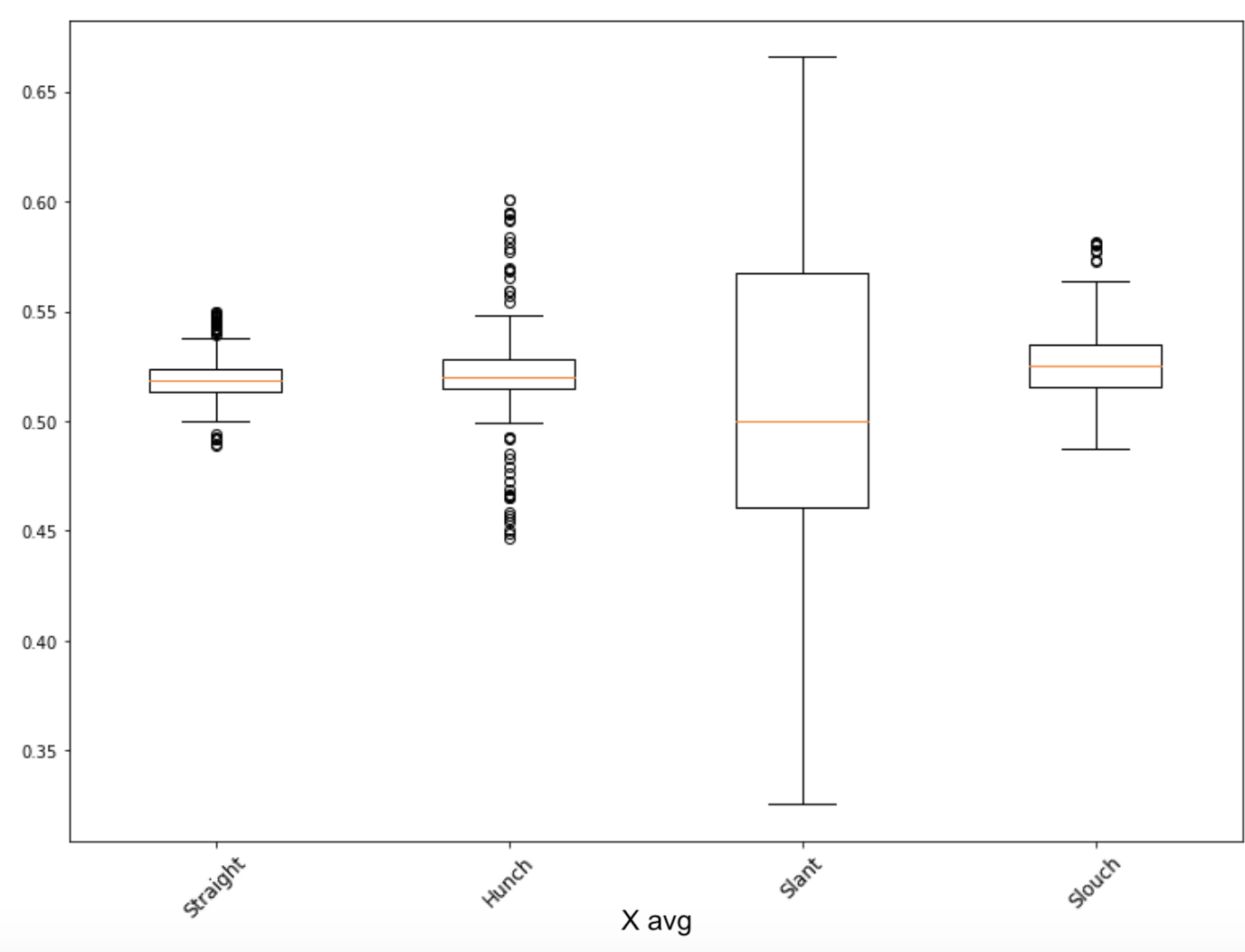
(a)

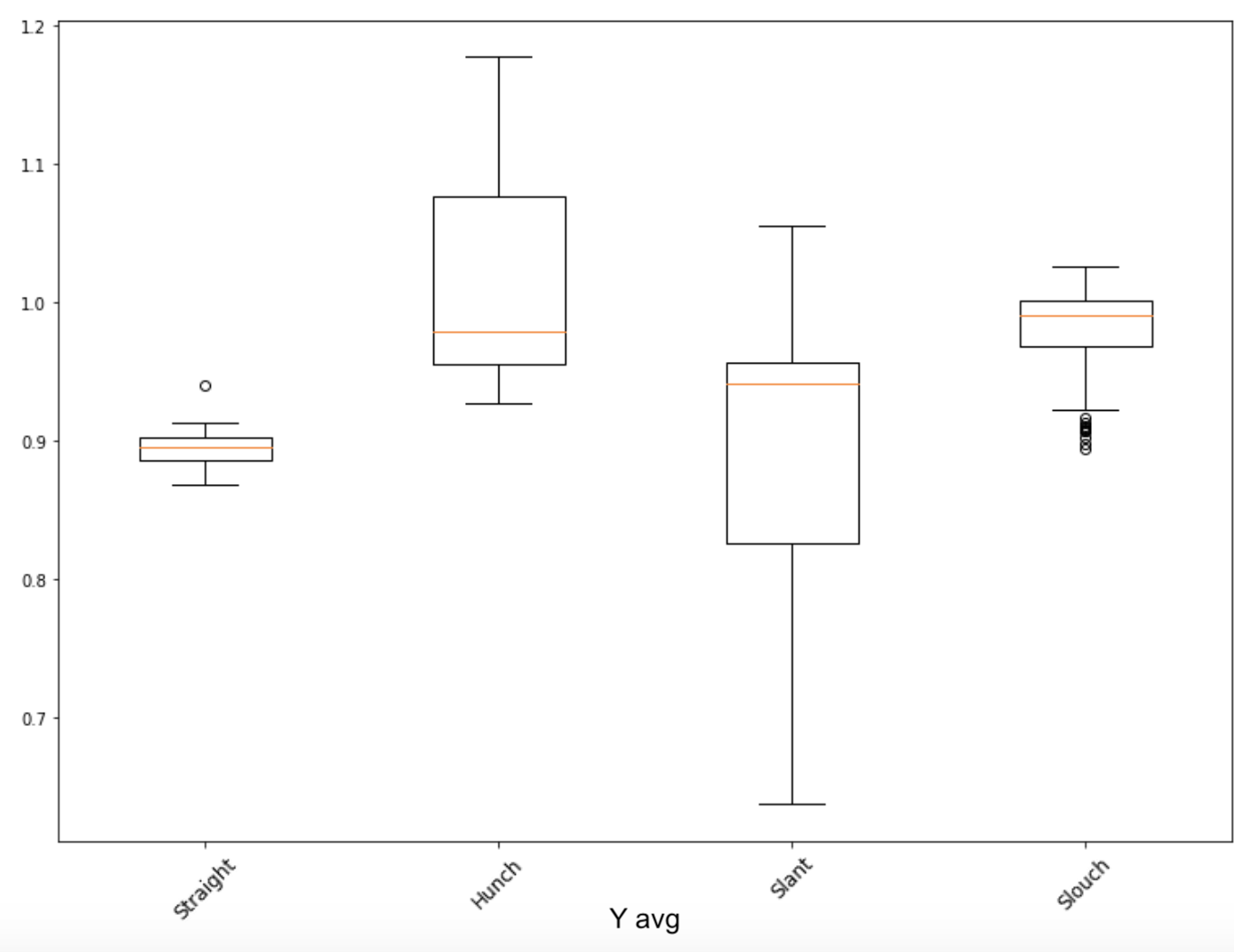
(b)

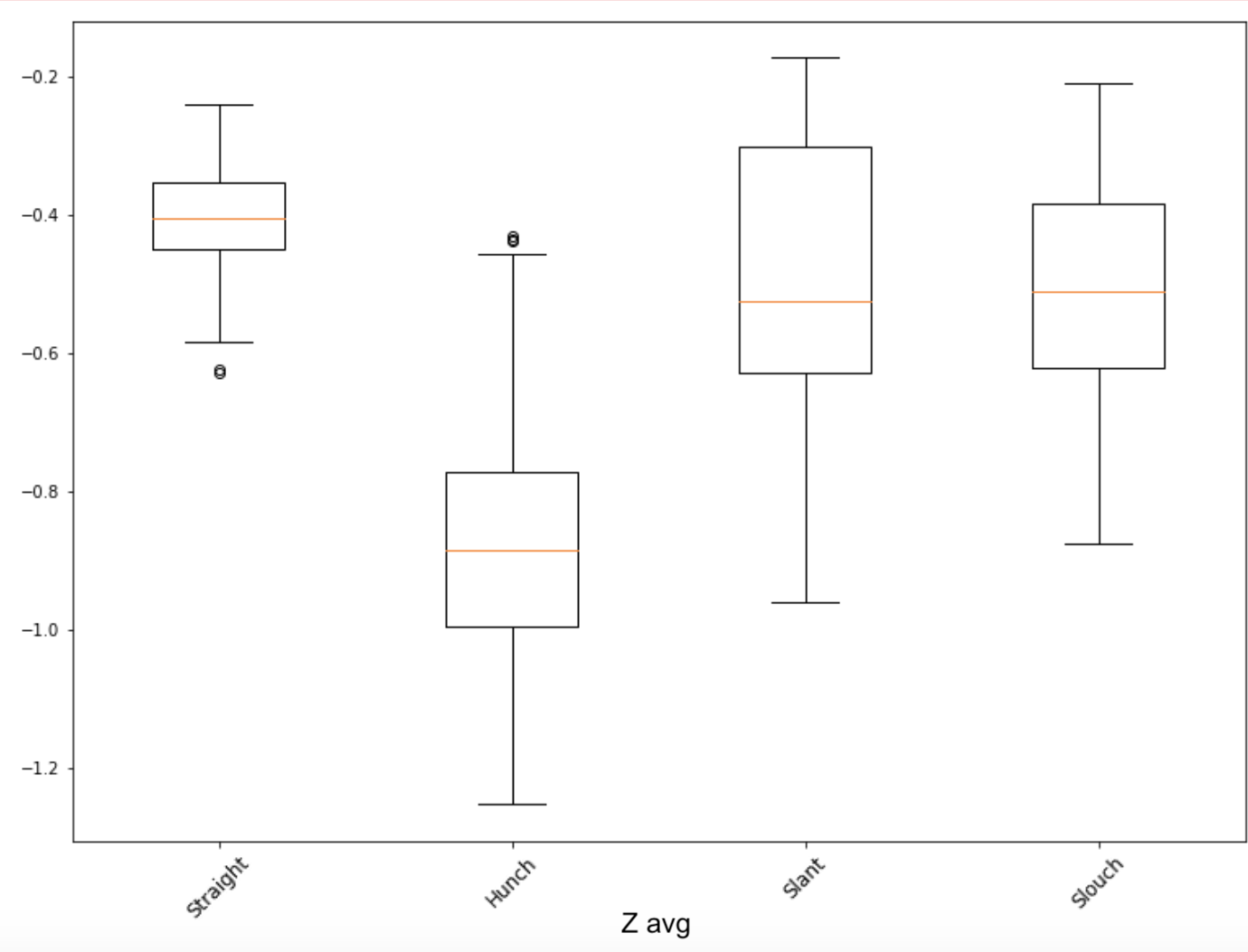
(c)

Fig. 2. 3D scatter plots of the same x, y, and z averages with rotated axes.

As depicted in Fig. 2, the data for the class “Slant” (green data points) has a high spread, due to the nature of the posture itself: leaning from side to side. Because the class “Slant” takes on such extreme values, it is harder to visualize the other classes, mainly the “Straight” and “Slouch” classes (red and blue data points), as they are surrounded by the “Slant” class. As is evident from Fig. 3, 4, and 5, the classes “Straight,” “Slouch,” and “Hunch” take on a relatively smaller range of values. Specifically, the class “Straight” has the smallest range on all three axes, and this is most likely because there is only one way to sit upright. The z coordinate measures depth: the further a landmark is from the screen, the closer its z coordinate is to zero, and vice versa. The “Hunch” class would have z values much smaller than zero compared to the other class, as shown in Fig. 5, because when the user hunchs over, his or her head is closer to the desktop camera.

  
Fig. 3. Box plot of the x averages for different sitting postures.

  
Fig. 4. Box plot of the y averages for different sitting postures.

  
Fig. 5. Box plot of the z averages for different sitting postures.

# RELATED WORKS

Traditional methods to stop slouching include exercising, stretching [6], and wearing an orthopedic posture brace [7]. Exercises and stretches are time-consuming and require immense self-discipline, so it may not be ideal for people who do not even have enough willpower to sit straight. On the other hand, orthopedic posture braces are clinically proven to improve postures, but only when given the appropriate brace and put on correctly. Furthermore, prolonged use of posture braces can worsen the issue by allowing the user to become over-dependent on the brace for keeping them upright, weakening their back muscles. Instead, posture braces work because they serve as a reminder for its user to sit up [7].

This makes a computer-aided posture tracking that can distinguish good sitting postures from bad sitting postures a perfect replacement for traditional methods. Computer-aided posture tracking is, like posture braces, a nudge to the users to sit up straight, but without giving them the option to lean on something else, forcing the users to rely on their muscles. Real-time tracking also reduces the need for the users to exercise their back muscles, as by consciously sitting correctly after being reminded by the computer, the users are constantly using and training their muscles to be constantly supporting their back to be upright.

Newer technology involves wearable textile sensors, inertial and magnetic sensors attached to the human body, depth cameras, Kinect sensors, and sensors attached to office chairs which collect data on the users, such as their forward inclination, distance to the computer, and relative skeletal angles [2]. Nevertheless, not only can these extravagant sensors be seen as limiting and hindering the user’s movement, but they may also be deemed unnecessary and inaccessible for the working middle class. In addition, these camera-based systems have demanding requirements for distance, lighting, and non-occlusion [2], making them inconvenient and unworkable for busy students or adults.

There have been many previous works that track human postures using RGB-depth images. For example, Huang & Gao used 3D joint coordinates captured by the depth camera to reconstruct realistic 3D human poses [2]. Liu et al. used a 3D Convolutional Neural Network called 3D PostureNet, employing Gaussian voxelization for the skeleton to represent posture configurations [8]. Pham et al. used Deep Convolutional Neural Networks based on the DenseNet model to represent spatio-temporal patterns of skeletal movements and perform human activity recognition [9]. Sengupta et al. proposed mm-Pose which detects 15+ skeletal joints using mmWave radar reflection signals. With the radar-to-image representation, a forked CNN architecture was used to predict the real-world 3D positions of the skeletal joints [10].

However, none of these methods work when the lower part of the body is not visible, i.e., the occlusion problem [2]. To address this, I will be training a machine learning algorithm that is capable of classifying different sitting postures solely based on a person’s shoulder and head position with the help of the MediaPipe Holistic Pipeline.

# SOFTWARE AND DEVELOPING TOOLS

## A. Software

I will be writing my code and implementing my model on Jupyter Notebook.

## B. Libraries

Numerous Python libraries, predominantly, the MediaPipe Holistic Pipeline, OpenCV, Pandas, and Scikit-learn.

As mentioned before, the MediaPipe Holistic Pipeline provides the pose, face, and hand landmark models that help recognize and label the key joints of the human in the frame, supplying the data (x, y, z coordinate, and visibility) I will be using to train my machine learning classification model. OpenCV is crucial for image processing and rendering the results onto the screen. Pandas helps with reading CSV files and working the tabular data within CSV files as data frames. Scikit-learn will be used to build custom machine learning pipelines and models. NumPy and Matplotlib were also used for manipulating and visualizing the data. Lastly, Simpleaudio contributes to audio playing when the user is in the wrong position.

## C. Laptop/Desktop Setup

The user’s desktop is placed an elbow’s length away so that the head and chest of the person are fully in the camera’s view, as Fig. 6 shows.

  
Fig. 6. The desktop setup.

## D. Sensors/Hardware

The data will be collected through the macOS Camera.

# LIST OF MILESTONES

## A. Edge Impulse

Initially, I used EdgeImpulse to try to classify images between “Straight” and “Hunch.” Although the model was able to reach an accuracy of 80%, as illustrated in Fig. 7, its performance during live classification was worse. Moreover, the image inputs were not processed, which means the model is unable to distinguish between the person and the background. Consequently, the model’s performance is even more likely to decrease in unseen scenarios when the background is new.

  
Fig. 7. Edge Impulse model accuracy score and confusion matrix.

## B. MediaPipe

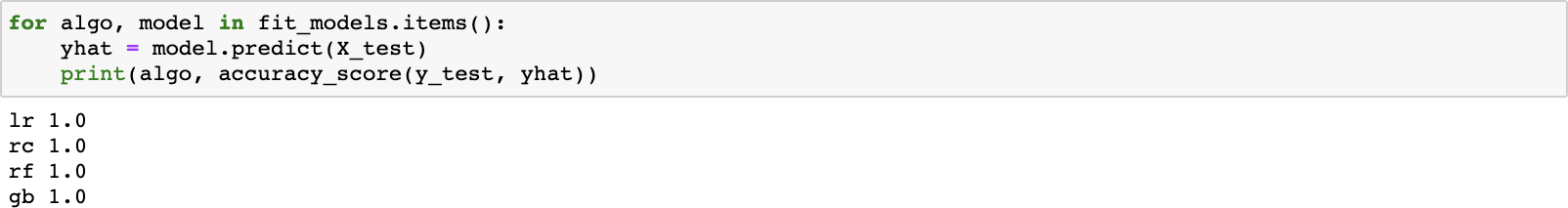
After the realization that my model’s accuracy can increase significantly if the person in the camera is already identified and labeled, I did a little research and found the MediaPipe Python library which provides various human landmark models such as the face, hand, and pose landmark models. Altogether, they form the MediaPipe Holistic Pipeline, which is capable of labeling hand, face, and full body landmarks in real-time and return the coordinates of each landmark. At first, I decided to use the holistic model which simultaneously tracks 33 pose, 42 hand, and 468 facial landmarks [4]. Since there is an x coordinate, y coordinate, z coordinate, and visibility score for each landmark, I ended up with (33 + 468)\*4 = 2,004 features for each sample. Although I merely had to run the data through Scikit-learn’s ML models, 2004 features were a bit too much data.

Trying to make my model more efficient and accurate, I decided to collect data again with only 33 pose landmarks, resulting in 33\*4 = 132 features per sample, a number significantly less than the previous 2,004 features.

## C. Training and Retraining

In the first training set, I collected 922 samples, with 226 from “Hunch,” 256 from “Slant,” 202 from “Slouch,” and 238 from “Straight.” The samples are randomly split into training and testing data sets with a ratio of 7 to 3. Then, I created four different machine learning pipelines. They are logistic regression, ridge classifier, random forest classifier, and gradient boosting classifier. Through each of the four pipelines, I also passed through a preprocessing function called standard scalar which standardizes the data so that no one feature overshadows the other features.

After training and testing the model with pre-collected data, I tested the data in real-time and realized it has trouble recognizing certain “Hunch” and “Slant” positions. So, I went back and collected 179 more samples, with 115 for “Hunch” and 64 for “Slant,” retrained the model with the now 1,101 samples, and achieved an accuracy score of 1.0 for all models when testing with the testing data set, which is illustrated in Fig. 8.

  
Fig. 8. Accuracy scores for the logistic regression (lr), ridge classifier (rc),   
random forest classifier (rf), and gradient boosting classifier (gb) model.

# RESULTS AND DISCUSSION

As perfect as the models may seem with accuracy scores depicted in Fig. 8, these numbers actually do not tell us as much about our actual data. Remember, there are a total of four classes. That means as long as the sitting posture is “Straight,” even if the algorithm believes the pose is 40% “Straight,” 20% “Slant,” 20% “Slouch,” and 20% “Hunch,” it will still accurately classify the posture as “Straight.”

So although the final result is accurate, the algorithm believes that there is a 60% chance that the posture is *not* “Straight.” With that in mind, I tested each of the four models out in real-time to select the final best model, one that can confidently declare the class of a posture. Here, “confidently declare” is defined such that the chance of the declared class will still be greater when comparing it against the combined chance of the other three classes, or that the chance of the declared class is greater than 50%. The linear regression model turned out to have the best result during live classification and became the model that is used in the algorithm to alert the user of a bad posture.

There are many limitations to my method. The major one being the fact that there is only one subject, me. This may have influenced the validity of the results, as there is no range in heights, and this model could have grown over-dependent on the absolute difference between the coordinates of specific joint landmarks. Furthermore, I struggle with sitting properly and have grown more accustomed to bad postures than feeling uncomfortable in them. So although I have no problem sitting up straight, I may decide that a slight “Hunch,” due to my bad habit of excessive hunching, is still a “Straight,” and affect the precision of the model. This is especially evident when I was testing out the model during live classification. I thought I was sitting upright yet the model classified my pose as “Hunch” and it took me a while to figure out that I still had a slight hunchback.

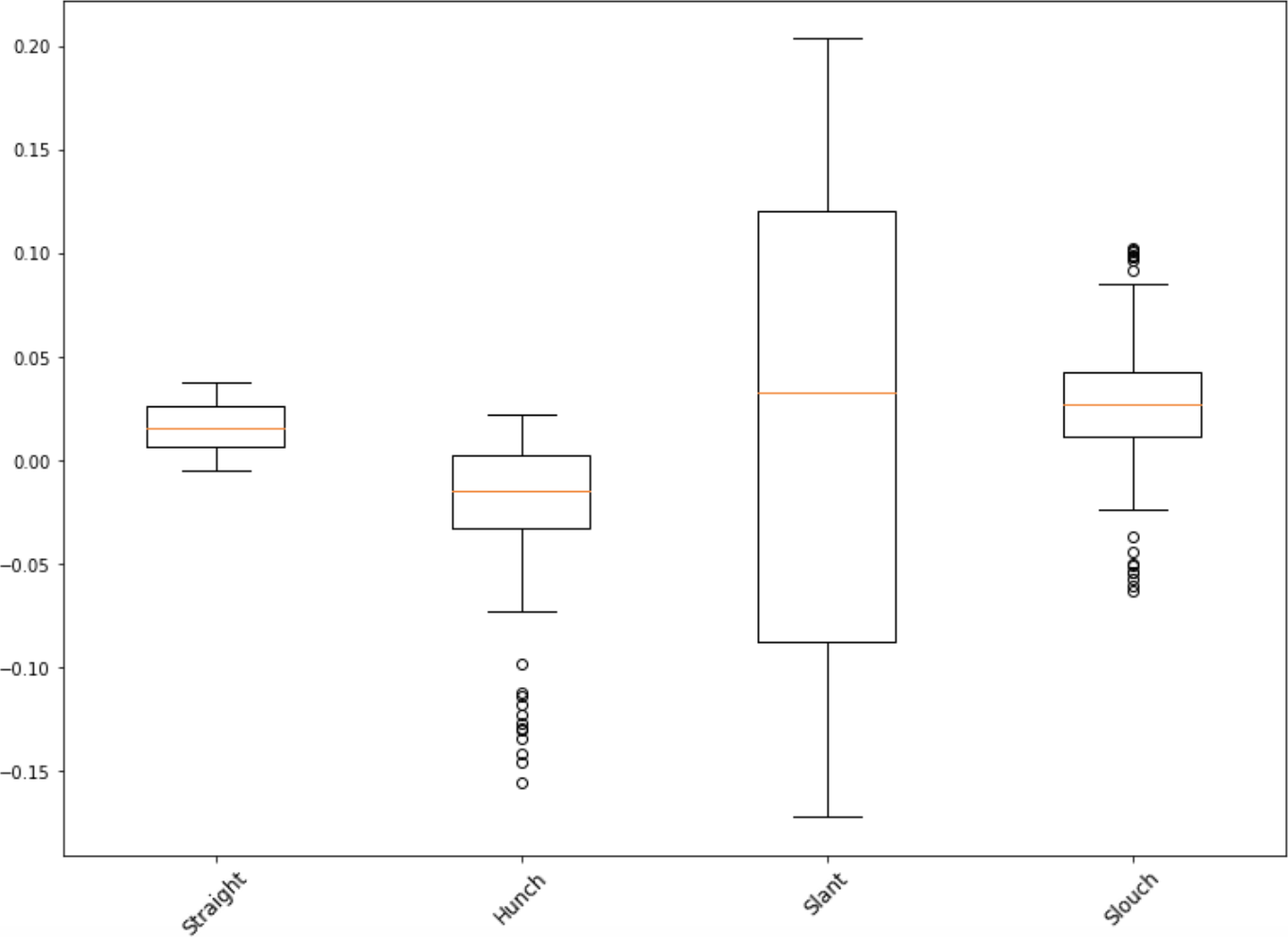
# CONCLUSION

I have proposed a machine learning classification algorithm that uses logistic regression and is built on top of the MediaPipe Holistic Pipeline for human sitting posture recognition. Based on my testing data, the logistic regression classification algorithm achieved an accuracy score of 100% in predicting the four sitting posture classes: “Straight,” “Hunch,” “Slant,” and “Slouch.” Unlike other related work, my method does not depend on outside sensors other than the computer or having an entire human skeleton visible in the frame, and can perform the human sitting posture classification when only as little as the chest and shoulders or the human torso is visible in the frame, making my approach more accessible and convenient for real-time human posture classification for students and office workers. Further improvements in dataset collection methodology can be made to account for different body shapes and disabilities.

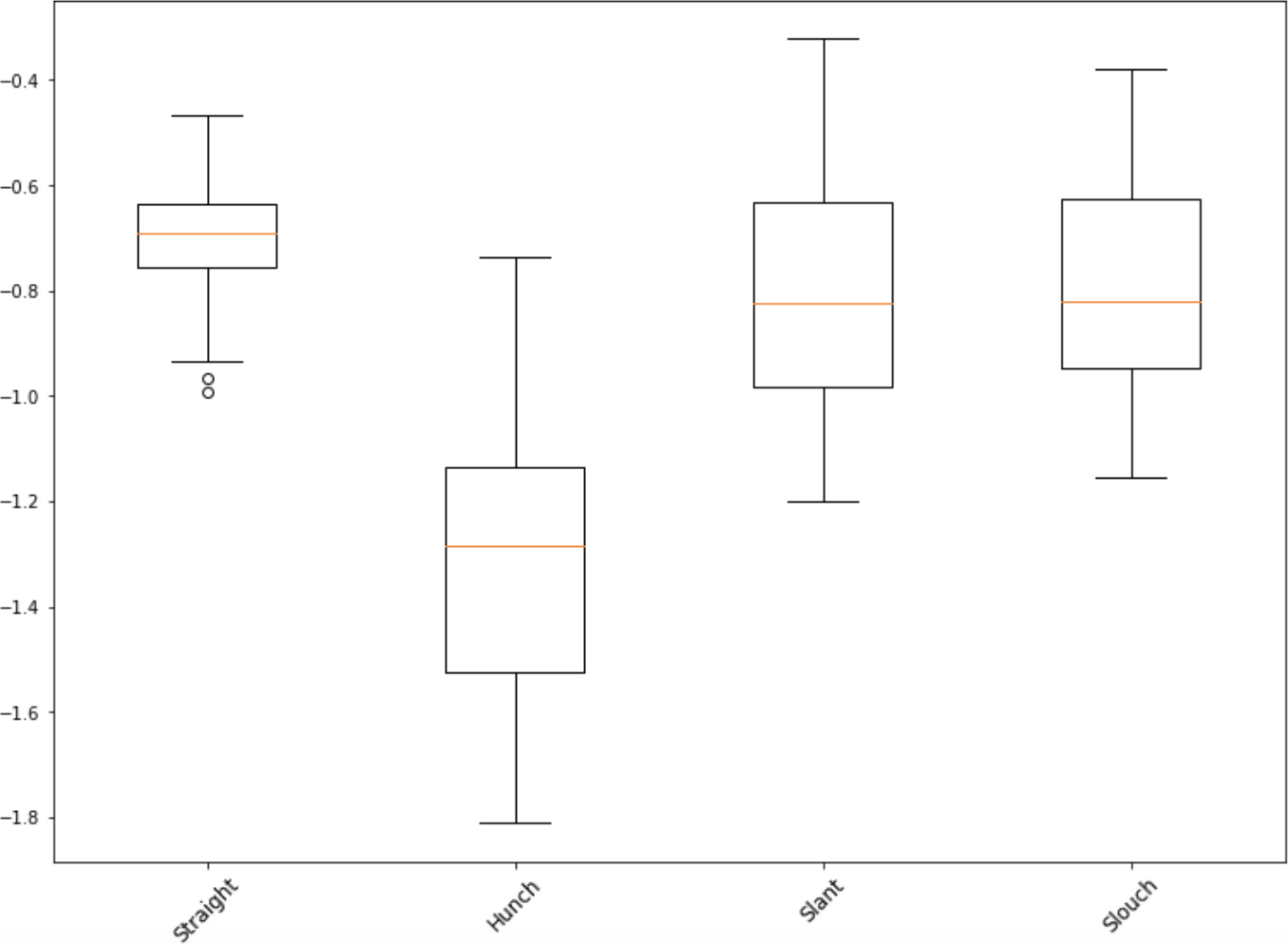
# FUTURE WORK

As mentioned previously in the Section V. of Results and Discussion, a major limitation of this model is the lack of subjects. This greatly restricts the model’s flexibility as it is only given one very specific instance, me, to train on. In the future, I can extend the subject group to include various ages, body types, gender, ethnicities, disabilities, and occupational groups to improve the results for real-world cases.

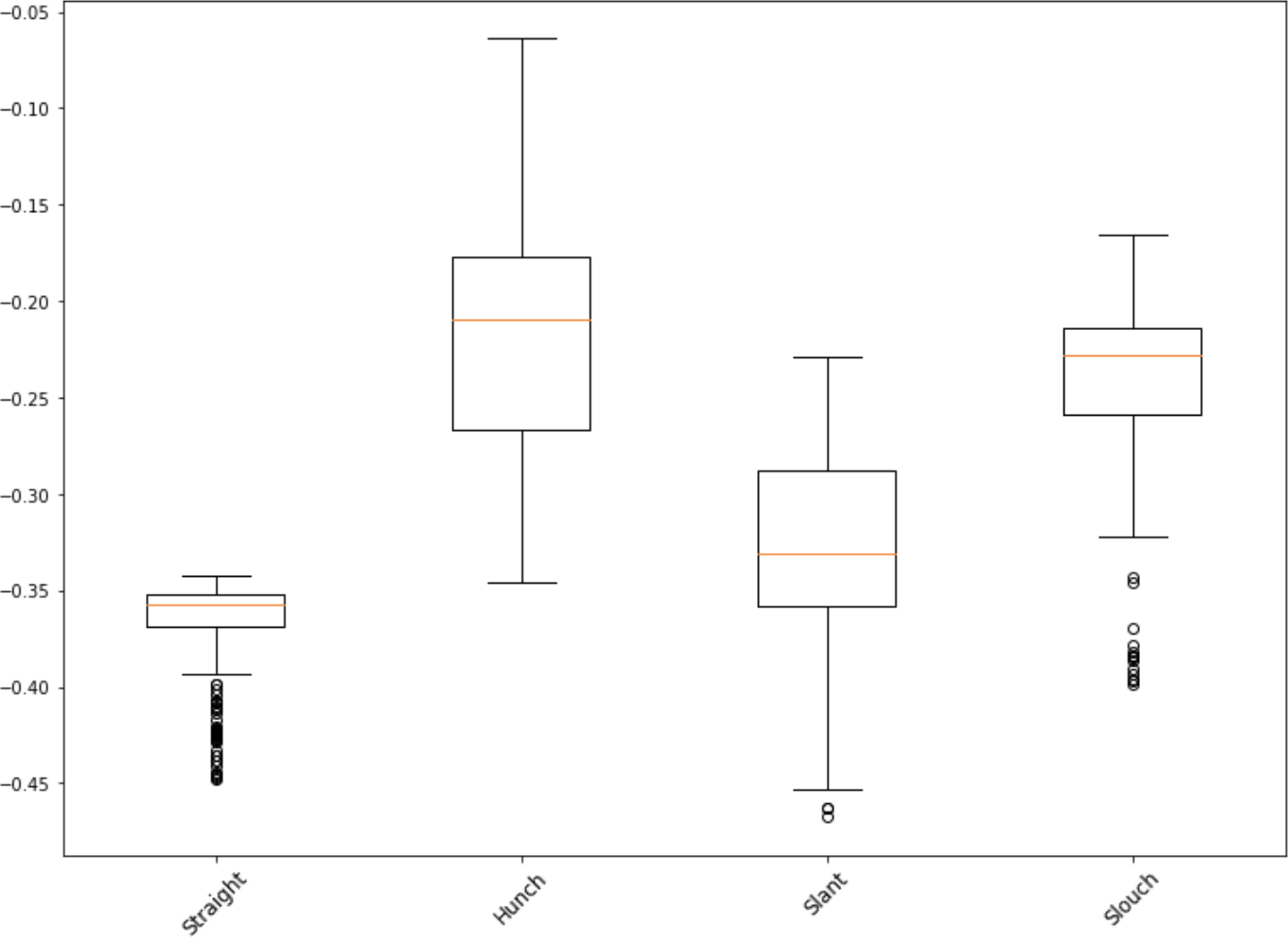
If given more time, I can also try to further manually process the landmark coordinates and differing features to optimize the model. For example, to distinguish “Slant” from the other classes, I can take shoulder\_height\_difference = y12 - y13 (the y coordinate of the left shoulder minus the y coordinate of the right shoulder), because when a person is slanted, their shoulders would be lopsided and the value for shoulder\_height\_difference will be further from 0, as illustrated in Fig. 9.

  
Fig. 9. Box plot of the y12 - y13 values for different sitting postures.

To distinguish “Hunch” from the other classes, I can just take z1, the z coordinate of the nose, because when a person is hunched over, their head would be closer to the screen and have smaller z coordinate values, shown in Fig. 10.

  
Fig. 10. Box plot of the z1 values for different sitting postures.

Lastly, to distinguish between the last two classes, “Straight” and “Slouch,” I can take y\_difference = y1 - (y12 + y13)/2, which is the y coordinate of the nose minus the average between the y coordinate of the right shoulder and left shoulder). This might work because when a person is slouched over, their shoulders tend to move upward and their head downward. So a person who’s slouching would have a smaller y\_difference value than a person who’s sitting up straight, as proven in Fig. 11.

  
Fig. 11. Box plot of the y1 - (y12 + y13)/2 values for different sitting postures.

# REFERENCES

[1] “6 Effects of Poor Posture on the Body.” *Integrated Rehabilitation Services*, Integrated Rehabilitation Services, 9 Aug. 2017, [integrehab.com/blog/back-pain/6-effects-poor-posture/](http://integrehab.com/blog/back-pain/6-effects-poor-posture/).

[2] Kulikajevas A, Maskeliunas R, Damaševičius R. 2021. Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ Computer Science* 7:e442 [doi.org/10.7717/peerj-cs.442](http://doi.org/10.7717/peerj-cs.442).

[3] “Image File Reading and Writing.” *OpenCV*, [docs.opencv.org/master/d4/da8/group\_\_imgcodecs.html](http://docs.opencv.org/master/d4/da8/group__imgcodecs.html).

[4] “MediaPipe Holistic.” *Mediapipe*, Google LLC, 2020, [google.github.io/mediapipe/solutions/holistic](http://google.github.io/mediapipe/solutions/holistic).

[5] “MediaPipe Pose.” *Mediapipe*, Google LLC, 2020, [google.github.io/mediapipe/solutions/pose.html](https://google.github.io/mediapipe/solutions/pose.html).

[6] Leung, Kenneth. “10-Minute Workout to Fix Bad Posture (&amp; Tips).” *BuiltLean*, BuiltLean LLC, 7 Apr. 2018, [www.builtlean.com/fix-bad-posture/](http://www.builtlean.com/fix-bad-posture/).

[7] Chandler, Meredith. “Do Posture Correctors Work? Here’s What Our Experts Think.” *Ergonomics Health Association*, Ergonomics Health Association, 7 May 2020, [ergonomicshealth.com/do-posture-correctors-work/](http://ergonomicshealth.com/do-posture-correctors-work/).

[8] Liu, Jianbo, et al. “3D POSTURENET: A Unified Framework for SKELETON-BASED Posture Recognition.” *Pattern Recognition Letters*, North-Holland, 28 Sept. 2020, [www.sciencedirect.com/science/article/abs/pii/S0167865520303615](http://www.sciencedirect.com/science/article/abs/pii/S0167865520303615).

[9] Pham, Huy Hieu, et al. “Spatio–Temporal Image Representation of 3d Skeletal Movements FOR View-Invariant Action Recognition with Deep Convolutional Neural Networks.” *MDPI*, Multidisciplinary Digital Publishing Institute, 24 Apr. 2019, [www.mdpi.com/1424-8220/19/8/1932](http://www.mdpi.com/1424-8220/19/8/1932).

[10] Sengupta, Arindam, et al. “mm-Pose: Real-Time Human Skeletal POSTURE Estimation Using MmWave Radars and CNNs.” *IEEE Sensors Journal*, IEEE, 6 May 2020, [ieeexplore.ieee.org/document/9083948/](http://ieeexplore.ieee.org/document/9083948/).