– Independent Work Report Fall 2021 –

Using Keypoint Detection to Classify Form in the Back Squat Exercise

Darren Zheng
Advisor: Dr. Alan Kaplan

Abstract

In the popular sport of weightlifting, learning the correct form for exercises can help lifters, novice and advanced, avoid injury. However, traditional form correction, such as hiring a personal trainer, can be inaccessible to a large portion of lifters. The approach described in this independent work project uses AlphaPose, an open source computer vision library, to detect keypoints on a lifter's body when performing the back squat exercise with correct and incorrect form. These keypoints are then used to train various machine learning models, which then are used to classify squat form and are evaluated. In this project, logistic regression outperformed ridge classification and a multi-layer perceptron in classifying good and bad squat form. This application provides a framework in which exercise form can be improved with easily accessible software, enabling lifters to master proper exercise technique and avoid injury.

1. Introduction

It is difficult to deny the health benefits of exercise in people's lives. Weightlifting, or strength training, is a popular form of exercise among Americans. Only falling behind walking in terms of popularity, an estimated 8.9 percent of Americans participate in weightlifting. This exercise is ahead of running (8.8%) and even swimming and other water sports (7.3%) [11]. Strength training's popularity is unsurprising when its benefits are examined. The American Cancer Society claims that some benefits of strength training include increased muscle mass, stronger bones, increased joint flexibility, improved weight control, and better balance [17]. Despite these essential health benefits, strength training without the proper form and technique can lead to severe injury. According to a study published in the British Medical Journal by Bengtsson et al, there is a common belief that

when it comes to the squat, bench, and deadlift, lifters' "injuries might be related to the excessively heavy loads, the large range of motion during the exercises, insufficient resting times between training sessions and/or faulty lifting technique" [1]. Having improper lifting technique may lead lifters to experience an abrupt injury, such as a muscle rupture, while performing an exercise, or they unknowingly could be incrementally damaging crucial parts of their body, which can lead to mysterious pains and aches. Novice lifters should learn the proper form for all exercises, and even intermediate to advanced lifters should periodically assure that their form is still adequate for their safety and longevity.

Among numerous exercises that lifters could perform, the back squat, or popularly known as the squat, is one of the popular and beneficial to everyday health. Squatting helps with leg strength, mobility, and balance by strengthening the gluteus maximus, hamstrings, and quadricep femoris muscles [19]. Moreover, training these muscles benefit daily tasks, such as walking, lifting heavier objects, and sitting [10, 16]. In order to perform this exercise, as shown in Figure 1, the squatter starts in a standing position, with feet flat on the floor and placed slightly wider than shoulder-width apart, and toes slightly pointed outwards. Knees and hips should be fully extended, and a barbell is balanced over the squatter's back, placed on top of the trapezius. As shown in Figure 2, the descent of the barbell is initiated by a simultaneous flexion of the hips, knees, and ankles. Next, as shown in Figure 3, the squatter should descend to the lowest point at which their joints are still comfortable, usually when the hip joint is at the level of or slightly lower than the knees. After the lowest point of the descent is reached, as shown Figure 4, the squatter initiates the ascent with the simultaneous extension of the hips, knees, and ankles, while assuring that their knees are positioned at the same or wider angle than that of their feet. The end of a repetition of a squat should mirror the initial position in Figure 1, making the squatter ready for the next repetition.

Figure 1: Initial squat position

Figure 2: The descent of the squat

Figure 3: The lowest point of the squat

Figure 4: The ascent of the squat

Because the squat recruits multiple large muscle groups, there are numerous opportunities for the squatter to incorrectly perform the exercise. Shown in Figure 5, one common mistake that leads to improper form occurs on the ascent of the exercise: when the squatter's knees cave inwards, or towards each other [10, 16]. Keeping the knees angled outwards and away from each other maintains a stable and strong foundation from which power can be transferred during the lift. When this foundation is compromised, unnecessary load is placed onto the squatters knees and can lead to severe injury. As shown in Figure 6, another common mistake in squatting is also on the ascent: when the squatter extends at their ankles and knees at a faster rate than that of their hips [10, 16]. This mistake leads to the squatter's upper body leaning too far forward and places necessary load on the lower back, which can lead to spinal injuries. Although there are additional form mistakes that occur on the descent of the squat, this paper will focus on these two common mistakes on the ascent of the exercise.

Figure 5: The knees of the squatter caving in on the ascent

Figure 6: The ascent being initiated by the extension of the knees before the extension of the hips

As discussed above, given the popularity of the sport and importance of proper form, it is beneficial for squatters to have easy access to a source for form critique. As personal trainers are not readily accessible to all, this feedback can be also achieved through software, which can extract information of the body position of the lifter through time. A common tool for such is keypoint detection,

a problem in computer vision in which the position and the orientation of an object is detected through a series of keypoints. Keypoints are parts of an object that are essential in the identification of an object and its position. Pose estimation uses keypoint detection on a human body to estimate the position and orientation of one or multiple people. AlphaPose is an open-source, multi-person keypoint detection application [4, 9, 21]. The goal of this project is to use automation to detect then classify form in dynamic weightlifting exercises, particularly the barbell back squat.

2. Related Work

Various methods have been implemented and evaluated in efforts to provide active sports participants with form feedback. The following outlines some of the previous approaches in categories of video capture, video processing, and trained models.

2.1. *Video Capture*

Escamilla et al used 2-D and 3-D camera techniques to perform a bio-mechanical analysis of 3 differing squat stance widths to determine the optimal width. Using two synchronized Sony HVM 200 video cameras, an external light source, and a 3-D calibration frame, Escamilla et al recorded squatters at a masters' level powerlifting competition, meaning all subjects were at least 40 years old [3].

Yasser et al used a thermographic camera on the Microsoft Kinect Xbox 360 to record lifters performing the deadlift, squat, and shoulder press to analyze form [22].

Reynolds et al attached Microsoft Kinect and surface electromyography (sEMG) probes to lifters' compression shorts to record the lifters' movement throughout an exercise. With the combination of these technologies, they were able to notify the lifter when improper form was performed and when their muscles were approaching fatigue [15].

In the project described in this paper, one iPhone 8 plus camera was used due to economic limitations; this could lead to a loss of information, as only one angle of the squat is being recorded. However, the same angle is roughly maintained during capture of all testing data. Moreover, the squatters sampled were in their early twenties, and the squats were not taken from a powerlifting competition.

2.2. *Video Processing*

Researcher Luke Genung took videos of himself deadlifting and combined those with existing videos of deadlifts from various sources. The deadlift videos were converted into thousands of JPEG image frames. To clean the data and reduce the amount of background noise, Genung crops out the background by filling it in with a single color before training a TensorFlow model to analyze form [5].

After performing keypoint detection with OpenPose, Trevor Phillips used time series to process the keypoint data. Then, he used the K Nearest Neighbors algorithm to classify squat form. Phillips was able to achieve an accuracy of 90% when classifying exercise type, but only an accuracy of 65% when classifying exercise form [13].

Srisen et al enhanced the Kinect sensor, a motion tracking device popularly used in weightlifting pose tracking, by implementing a joint correction process for the hands, feet, and knees. They used fast cross-correlation to match templates of the video and the Lucas-Kanade algorithm to obtain the optical flow of an entire subimage [18].

Zhi-chao and Zhang trained a fully convoluted network in order to extract the foreground and remove background interference from videos of people lifting. Additionally, they used a convolution

neural network to classify the foregrounds of the lifting videos. They concluded that their algorithm was effective in selecting high image quality key pose frames [23].

Kuruppu et al used the ASUS Xtion PRO, a low cost 3-D gaming camera, to record weightlifters performing the snatch exercise. Then, they used OpenNI/NITE, an open source joint tracking process, to capture the skeleton of the lifter through each frame of the output from the Xtion PRO. They also used the Correlation Coefficient Template Matching algorithm to track the joints of the lifter in a 2-D video. Taking the 2-D and the 3-D video outputs after joint tracking, they validated that the two processes resulted in comparable outputs. Additionally, the three phases of the snatch exercise were recorded and tracked separately [8].

In the approach described in this paper, the observed exercise was the squat instead of dead-lift, snatch, etc. Only the concentric, or ascent, portion of the squat was utilized in pose detection and training the models. Keypoint detection was used to reduce the background noise. Additionally, the models in this approach were trained on those keypoints instead of the frames from the videos alone.

2.3. *Trained Models*

Kanase et al used keypoint detection, vector geometry, and dynamic time warping to analyze the technique of four exercises - bicep curl, front raise, shoulder shrug, and shoulder press. Kanase et al used OpenPose to perform keypoint detection on a series of exercise videos. Then using dynamic time warping and vector geometry, form feedback was produced [7].

Yasser et al used Naive Bayes, KNN, and Fast DTW models to provide lifters with feedback on form. They found that the Fast DTW model outperformed other classification models, and for dependent user movements, it could achieve recognition with 100% accuracy [22].

Dr. Sofia Julie trained a convolution neural network in order to study the effect of scapula dyskinesis, a deviation from the normal position of the scapula, under the influence of weightlifting [6].

Using Keras, Cheung et al created a Recurrent Neural Network - Long Short Term Memory (RNN-LSTM) to classify deadlift form. They were able to classify form with 72.5% accuracy [2].

Phillips trained a Recurrent Neural Network with exercise data because it is effective for use on sequences of data, such as frames of a video. Phillips also implemented a 3-D Convolutional Neural Network (CNN) to train the data to result in an accuracy score of 69.23% [14].

For the project described in this paper, AlphaPose was used for keypoint detection, and instead of using dynamic time warping to deal with differing video lengths, frames of the squat videos were extracted. Also, instead of using vector geometry, a neural network was trained in hopes of detecting less obvious features that can lead to poor squat form.

3. Approach

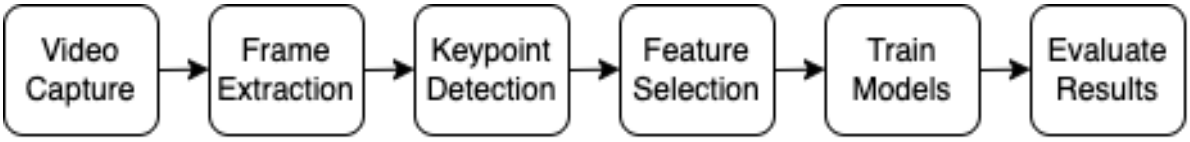

Figure 7: Flow of this application

The flow chart in Figure 7 illustrates the approach taken for this project. In studying the position of the body during a lifting exercise, there are various avenues for gathering data. In this project, two dimensional video capture was chosen because it is easily accessible, replicable, and inexpensive. Other possible methods of data collection include using thermal cameras, physical body sensors, or three dimensional cameras. After video capture, each video needed to be separated into its subsequent frames. An issue with training models on multiple videos of a moving body is the

question of how to compare two temporal sequences that vary in speed. One method for avoiding this issue is Dynamic Time Warping, which is an algorithm that Kanase et al used [7]. In this approach, this problem was avoided by extracting relevant frames (more about how these are defined in Implementation) from each video, and creating the dataset from those frames alone. Next, keypoint detection was used on each frame to extract the features of each body position in each frame. Keypoint detection provided the position of various keypoints of the body, such as the right knee and the right hip. An alternative to keypoint detection is to use the frame as sample images alone, but this includes various confounding factors present in the background of the image. In order to eliminate this background noise, Genung merely covered the background by coloring over it and training his models with those images [5]. In this project, keypoint detection was chosen to eliminate background noise because it extracted the position of only the keypoints of the lifter's body, thus only extracting foreground features; additionally, training models on keypoints instead of images minimized computational times when fitting models. After keypoint detection, the dataset was created by allocating each frame as a sample and each keypoint produced as a feature. Additionally, each frame was labeled as 1 if it was extracted from a video exhibiting good form and a 0 if the frame was extracted from a video exhibiting bad form. After the dataset was created, feature selection was performed in order to identify a subset of features that was most influential in labeling each sample. In other words, feature selection would identify the body keypoints whose positions were most important in classifying a repetition as good or bad form. After feature selection was performed, various machine learning models were trained, and the predictions of these models were evaluated. Because a set of classification models were trained, the best performing one could be identified.

4. Implementation

4.1. *Video Capture*

Videos were captured with the camera of an iPhone 8 Plus. The phone was positioned on a tripod roughly four feet above the ground and at a 45 degree angle from the front of the lifter. The lifters

were comprised of four different subjects, all healthy male undergraduates. Each lifter conducted multiple sets of weighted back squats, with each set containing ten repetitions. In roughly half of the sets, the lifters performed squats with good form, and in the other half of the sets, the lifters performed squats with bad form. Within the squats with bad form, roughly half of the repetitions demonstrate the lifter's knees caving in, as shown in Figure 5, and the other half of the bad repetitions demonstrate the lifter's upper torso being bent too far forward, as shown in Figure 6. Each one of these sets were recorded as one singular video on the iPhone.

After recording all of the sets, each video containing a set was trimmed to produce a new video for each individual repetition performed in the set. Each new video showing a repetition was trimmed on Apple images' edit tool, has a duration of one second, and only shows the ascent of the squat, as that was where the poor technique occurred. After creating these new videos containing each individual repetition, 166 videos of good squat repetitions and 91 videos of bad squat repetitions were collected. These videos were then uploaded to two Google Drive folders organized by form.

4.2. *Frame Extraction*

The remaining portion of this implementation was performed in Google Colab. To prepare the repetition videos for classification, a script was executed to extract frames from each video. Each video is roughly one second long and recorded at 60 frames per second. Using the OpenCV library, the frames of the first and last 25% of the videos were discarded and the rest of the frames were kept. This was because the frames that are towards the middle of the video are more significant, and most of the videos, regardless of good or bad squat form, started and ended similarly. The frames that were kept from each video were the relevant frames, and will be the only ones used moving forward. The cutoff value of 25% was arbitrary, and was determined from some trial and error. Figure 8 shows an example of the first relevant frame extracted from a good form video. Figure 9 shows an example of the last relevant frame extracted from the same good video. All frames that

followed the first relevant frame and preceded the last relevant frame were also extracted as relevant frames.

Figure 8: An example of a first relevant frame

Figure 9: An example of a last relevant frame

After running the frame extraction script on all of the repetition videos, 5172 good frames and 3523 bad frames were extracted for a total of 8695 frames. These frames were then uploaded to two Google Drive folders separated by whether the frames were extracted from a good or bad squat repetition.

4.3. *Keypoint Detection*

After extracting the frames, keypoint detection was then performed on the frames by using an open source application named AlphaPose. AlphaPose is an open source, accurate pose estimator that produces 17 keypoints representing different body parts for a given person in an image. AlphaPose produces visual results with the keypoints overlaid on top of the original frame as well as a textual representation of the keypoints, providing three values for each keypoint: the x coordinate, the y coordinate, and a c-value, which is a confidence score up to 100 that represents the degree of

confidence with which the keypoint was detected.

The x and y coordinates are relative to the image frame, not the lifter. For each of the 8695 total frames as input, AlphaPose produced an image of the original frame with the 17 keypoints overlaid and lines connecting them to form a skeletal figure, as shown with an example in Figure 10. AlphaPose also produced a textual representation of the keypoints in the format of a json file, as shown with an example in Figure 11. The good frames were inputted into AlphaPose first, and that produced a json file containing all of the keypoints information for the good frames, and then the same procedure was done on the bad frames.

Figure 10: Visual result of keypoint detection on a squat frame using AlphaPose


```
[{"image_id": "good_1.jpg", "category_id": 1, "keypoints": [861.9473266601562,
1649.6337890625, 0.8930714726448059, 916.3292846679688, 1631.506591796875,
0.9418668150901794, 861.9473266601562, 1595.251953125, 0.8735575675964355,
1061.347900390625, 1685.888427734375, 0.9608651399612427, 843.8200073242188,
1613.379150390625, 0.48017531633377075, 1260.748291015625, 1885.2890625,
0.7518171072006226, 934.4566040039062, 1776.525146484375, 0.7414545416831197,
1641.422119140625, 2012.18017578125, 0.7676333785057068, 843.8200073242188,
1685.888427734375, 0.6112838387489319, 1532.658203125, 1903.416259765625,
0.7889202833175659, 807.5653686523438, 1613.379150390625, 0.7012014389038086,
1423.894287109375, 2284.090087890625, 0.7553151249885559, 1188.239013671875,
2229.7080078125, 0.7612645626068115, 1188.239013671875, 2592.25439453125,
0.8580217957496643, 861.9473266601562, 2392.85400390625, 0.8999889492988586,
1369.51220703125, 2954.80078125, 0.8491950631141663, 1079.47509765625, 2737.27294921875,
0.9301170706748962], "score": 2.9922189712524414, "box": [615.353515625,
1383.1622314453125, 1018.8800048828125, 1856.2376708984375], "idx": [0.0]}, {"image_id":
"good_2.jpg", "category_id": 1, "keypoints": [847.9235229492188, 1632.9747314453125,
0.8775139451026917, 921.0931396484375, 1596.389892578125, 0.9421643614768982,
847.9235229492188, 1578.0975341796875, 0.8968108892440796, 1067.432373046875,
1651.26708984375, 0.9663987159729004, 847.9235229492188, 1578.0975341796875,
0.43840187788009644, 1268.6488037109375, 1870.7760009765625, 0.7424153089523315,
939.3855590820312, 1761.021484375, 0.7296923398971558, 1634.4969482421875,
1998.82275390625, 0.7139930725097656, 829.6311645507812, 1669.5595703125,
0.5015809535980225, 1561.3272705078125, 1889.068359375, 0.7131101489067078,
793.0463256835938, 1596.389892578125, 0.6720899939537048, 1433.280517578125,
2273.208740234375, 0.751531183719635, 1195.479248046875, 2218.33154296875,
0.7497570514678955, 1195.479248046875, 2602.47216796875, 0.8340876698493958,
866.2159423828125, 2401.255615234375, 0.8842099905014038, 1378.4033203125, 2968.3203125,
0.8398277163505554, 1085.724853515625, 2730.51904296875, 0.9215768575668335], "score":
2.9726648330688477, "box": [642.2939453125, 1364.076416015625, 978.3238525390625,
1873.1423339843751], "idx": [0.0]}, {"image_id": "good_3.jpg", "category_id": 1,
"keypoints": [855.0453491210938, 1604.279541015625, 0.8872259855270386, 909.01806640625,
1568.2977294921875, 0.9370658993721008, 855.0453491210938, 1568.2977294921875,
```


Figure 11: Textual results of keypoints detection on good squat frames using AlphaPose

Next, the two json files containing the keypoint detection results for the good and bad frames were converted into two dataframes using the Python package Pandas where the columns were the x-coordinates, y-coordinates, and the c score of each keypoint and the rows were the frames [20]. Then, a single column was added at the end of both dataframes that was a 1 for each frame that came from a good squat video, and a 0 for each frame that came from a bad squat video. Then, these two dataframes were combined to create the final dataset with dimensions of 8695 rows by 52 columns. A small subset of the resulting dataset is shown in Figure 12.

left_knee_c	right_knee_x	right_knee_y	right_knee_c	left_ankle_x	left_ankle_y	left_ankle_c	right_ankle_x	right_ankle_y	right_ankle_c	form
0.858022	861.947327	2392.854004	0.899989	1369.512207	2954.800781	0.849195	1079.475098	2737.272949	0.930117	1
0.834088	866.215942	2401.255615	0.884210	1378.403320	2968.320312	0.839828	1085.724854	2730.519043	0.921577	1
0.824671	855.045349	2395.879150	0.898314	1376.781494	2953.597168	0.822202	1088.927002	2737.706299	0.916123	1
0.836506	846.695068	2390.498047	0.945954	1374.694702	2954.911377	0.842649	1083.384521	2736.428711	0.896197	1
0.866355	858.630981	2389.067139	0.937431	1365.004517	2951.704346	0.850100	1083.685913	2726.649414	0.888231	1
...	...	...	...	...	...	...	...	...	...	...
0.051053	1348.152588	1806.991821	0.041072	1332.644775	1853.515503	0.099702	1309.382935	1838.007568	0.138897	0
0.799568	771.944702	2506.733154	0.861998	1194.999146	3025.936279	0.861506	925.782654	2833.638672	0.978105	0
0.061687	1299.030029	1266.017944	0.054533	1283.387817	1938.634155	0.167484	1275.566650	1930.812988	0.216879	0
0.809335	773.137207	2511.637939	0.876011	1210.531372	3006.083496	0.858139	944.291443	2834.929199	0.966439	0
0.850456	769.087280	2507.671387	0.882931	1194.189209	3010.064453	0.822205	923.669800	2836.159180	0.929911	0

Figure 12: A subset of the final dataset of frames and their keypoint detection results

Using the resulting dataset, the feature set and label set were then created. The feature set came from slicing the dataset pictured in Figure 12 by keeping all the columns that include the x-coordinates, y-coordinates, and the c score of each keypoint for all samples. Moreover, the label set was created by slicing the dataset by only keeping the column labeled "form". After creating these sets, the training and test sets were created. A function from the Sci-Kit Learn library, a machine learning library built in Python, was used to complete this task with a classic 80/20 training test split [12]. The result is a training set comprised of two dataframes, one for the features (x-train) and one for the labels (y-train) of 6956 random samples, and a test set comprised of of two dataframes, one for the the features (x-test) and one for the labels (y-test) of the remaining 1739 samples.

4.4. *Feature Selection*

After creating the training and test sets, feature selection was performed to identify a subset of features on which the labeling of samples are highly dependent. Using Lasso feature selection from Sci-kit Learn with an alpha of 0.7, a subset of features were selected.

4.5. *Train Models*

Next, a variety of classification models were trained on the training set. The first model is logistic regression, chosen because of its popularity as a classifier and max iterations was set to 5000. Next, ridge classification was used because it treats multi-colinearity within the features, which is helpful because there exists linear relationships between the features in the dataset, since they represent highly connected positions of body parts. The ridge classifier was trained with an alpha of 1. Finally, a multi-layer perceptron, or an MLP, was trained because as a neural network, it can identify hidden patterns and correlations in raw data, which is useful because relationships between certain body parts, are important to classify squat form. The MLP was trained with with 100 hidden layers. After all three models were fit to the training set, they were used to predict the labels of the test feature set.

4.6. *Evaluation Metrics*

After using the three classifiers to predict labels for the feature test set, two evaluation metrics were used to determine the effectiveness of the models. First, an accuracy score was calculated for each model by dividing the number of correctly predicted samples by the total number of samples in the test set. The accuracy score was presented as a percentage, and offers a representation of the raw percentage of samples that each classifier predicted correctly. Then, confusion matrices were created for each of the classifiers. For each model, confusion matrices display four statistics: the number of positive labels that the model predicted correctly, the number of positive labels that

the model predicted incorrectly, the number of negative labels that the model predicted correctly, and the number of negative labels that the model predicted incorrectly. Confusion matrices give an evaluation of how a model performs in terms of the number of false positives, or incorrectly predicted samples that are labeled positive, and false negatives, or incorrectly predicted samples that are labeled negative. In this case, positives samples were labeled as 1, representing good squat form, and negatives samples were labeled as 0, representing bad squat form.

5. Results

5.1. *Feature Selection*

The resulting subset of features resulting from feature selection are listed in Figure 13. These features hold the most weight for determining the labels for samples in the training set. There are a total of 19 of extracted features out of the original 52 features. Each feature represents an x-coordinate, y-coordinate, or a c-value for a body part. For example, only the y-value of the nose was extracted, which makes for one feature, while both the x and y coordinate for the right elbow were extracted, which makes for two features.

Nose (y)	Left shoulder (x)	Left hip (y)
Right eye (y)	Right shoulder (x)	Left knee (x & y)
Left ear (y)	Right elbow (x & y)	Right knee (x & y)
Right ear (y)	Left wrist (x & y)	Left ankle (x & y)
	Right wrist (x)	Right ankle (x)

Figure 13: Resulting subset of features after feature selection

5.2. *Accuracy Scores*

After training the three classification models on the training set and predicting the labels of the test set, logistic regression yielded an accuracy score of 99.60%, ridge classification yielded an accuracy score of 99.02%, and the multi-layer perceptron yielded a score of 98.80%. The logistic regressor outperformed the ridge classifier, which outperformed the MLP. The average of these three accuracy scores is 99.14%.

5.3. *Confusion Matrices*

The confusion matrices of the three models are pictured in Figure 14. For each confusion matrix, the upper left square represents the number of true negatives, the upper right square represents the number of false positives, the lower left square represents the number of false negatives, and the lower right square represents the number of true positives.

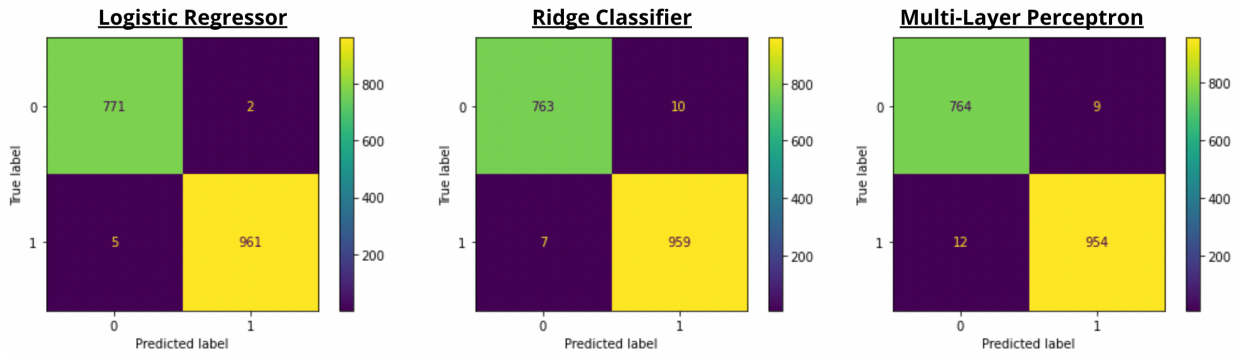

Figure 14: Confusion matrices of the three classification models

The logistic regressor yielded a false positive rate of 0.12% and a false negative rate of 0.29%. The ridge classifier yielded a false positive rate of 0.58% and a false negative rate of 0.40%. The MLP yielded a false positive rate of 0.52% and a false negative rate of 0.69%. Between the three models, the average false positive rate is 0.41% and the average false negative rate is 0.46%. Ridge was the only model that produced a higher false positive rate than false negative rate on the test set.

6. Evaluation

Among the 17 selected features resulting from feature selection, no c-values were extracted. Therefore, the x and y coordinates of the body parts are more heavily weighted than c-values in predicting the labels of the test set samples. A possible explanation for this is that the c-values produced by keypoint detection are all relatively high with low variance among different body parts, which does not provide much additional helpful information in model prediction. Since the c values are empirically and intuitively not as essential as the x and y coordinates of the keypoints in model prediction, one could consider training and testing the models on solely the x and y coordinates.

Among the three models trained, the logistic regressor yielded the highest accuracy score, making it the best model evaluated in terms of the raw number of correctly predicted samples. However, an average accuracy score among all models of 99.14% was observed. This incredibly high score signifies that all the models performed well on average, and keypoint detection combined with machine learning models can be a viable and practical approach to providing lifters with accurate form feedback.

After using confusion matrices to evaluate the false positive and negative rates of the three models, the logistic regressor produced the lowest false positive rate and the lowest false negative rate. This observation is not surprising since the logistic regressor produced the fewest number of incorrectly predicted samples. Moreover, the logistic regressor and the MLP both produced lower false positive rates than false negative rates, while the ridge classifier produced a higher false positive rate than a false negative rate. The former is preferable, as false negatives mean that although the lifter's form was actually good, the model labels it as bad, which does not engender relatively detrimental implications. However, a false positive would mean that although a lifter's form is bad, the model labels it as good, which could lead to the lifter continuing to perform bad form while thinking that it is good, leading to possible injuries. Although low false positive rates and false negative

rates are desired, a lower false positive rate is more essential in this application, which logistic regression and MLP provide. After evaluating the three models with accuracy scores and confusion matrices, the logistic regressor performed the best when it comes to predicting squat form from this self-generated dataset, as it has the highest accuracy score, and yields a lower false positive rate than false negative rate. Logistic regressor is a simple model when compared to the MLP, which is a neural network. Although the logistic regressor outperformed the MLP, the margin with which it did was small: an difference in accuracy scores of 0.80%. Neural networks require large dataset sizes to produce more accurate results, so perhaps a larger dataset may have engendered a slightly higher accuracy score for the MLP.

7. Conclusion

Ultimately, the application described in this independent work project shows that the goal of using automation to improve exercise form is entirely possible. This approach shows that machine learning models trained on keypoint locations, produced by keypoint detection, can engender high accuracy scores for classifying back squat form. It can be reasonably assumed that the work done for this project with the barbell back squat is transferable to other weightlifting exercises, given the proper camera angles and video capturing technique. With this assumption, this application presents a framework which easily accessible software can use to provide accurate exercise form correction to lifters for a variety of dynamic weightlifting exercises.

Some limitations of this work include the video capturing equipment and technique, keypoint detection usage, model selection, sample bias, and the types of poor technique surveyed. To capture squat videos, a singular 2-dimensional iPhone 8 Plus camera was used due to financial limitations. Using a 3-dimensional camera or multiple 2-dimensional cameras from various angles could lead to additional features that can predict exercise form more accurately. Also, more samples could be collected to allow higher performance in certain models, as many neural networks perform

better with a larger training set. In addition to video capture, keypoint detection was used in this application by training models on the raw x coordinates, y coordinates, and c-values of each body part. There was also an inherent bias in the subjects recorded for this project. Only four subjects were recorded, and they were male and in the age range of 20-21. More effective models should be trained on a larger age range as well as females to be more useful to a more diverse population of lifters. Moreover, only 2 types of bad technique, caving knees and being bent too far forward, both of which were only from the ascent, were used for training the models. To produce a more robust model, more forms of bad technique should be used to be able to inform lifters of other poor techniques they could be repeating. Future work could include training models on the images of the keypoints being connected by straight lines or the distances between relevant body parts. Finally, this application trained three models, the logistic regressor, the ridge classifier, and the MLP. Future work can implement a larger variety of classification models that would lead to higher accuracy. Moreover, future models trained could predict the problem of bad squat form and how to correct it.

References

[1] V. Bengtsson, L. Berglund, and U. Aasa, "Narrative review of injuries in powerlifting with special reference to their association to the squat, bench press and deadlift," *BMJ Open Sport Exercise Medicine*, vol. 4, no. 1, Jul 2018. [Online]. Available: https://bmjopensem.bmj.com/content/4/1/e000382

[2] S. Cheung, H. Li, P. Mak, and A. Tang, "Workout form classifier," *Rutgers University*.

[3] R. Escamilla, G. Fleisig, T. Lowry, S. Barrentine, and J. Andrews, "A three-dimensional biomechanical analysis of the squat during varying stance widths," *Medicine Science in Sports Exercise*, vol. 33, 2001.

[4] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.

[5] L. Genung, "Ai-powered personal trainer to check weightlifting form," *Medium*, 2020.

[6] S. Julie, "The use of convolution neural network algorithms in the biological image of weightlifting of scapula dyskinesis," *Malaysian Sports Journal*, vol. 1, 2019.

[7] R. R. Kanase, A. N. Kumavat, R. D. Sinalkar, and S. Somani, "Pose estimation and correcting exercise posture," *ITM Web of Conferences*, vol. 40, 2021.

[8] G. Kuruppu, T. Fernando, S. Kodituwakku, and U. Pinidiyaarachchi, "Marker less tracking of weightlifting snatch movement using xtion pro live," *University of Peradeniya*.

[9] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," *arXiv preprint arXiv:1812.00324*, 2018.

[10] G. Myer, A. Kushner, J. Brent, B. Schoenfeld, J. Hugentobler, R. Lloyd, A. Vermeil, D. Chu, J. Harbin, and S. McGill, "The back squat: A proposed assessment of functional deficits and technical factors that limit performance," *Strength and Conditioning Journal*, 2014.

[11] B. of Labor Statistics, "Sports and exercise among americans," *The Economics Daily*, 2016.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] T. Phillips, "Exercise classification with machine learning (part i)," *Towards Data Science*.

[14] ——, "Exercise classification with machine learning (part ii)," *Towards Data Science*.

[15] N. Reynolds, C. Corbo, J. Gardner, and B. Sherman, "Dft: Digital fitness trainer."

[16] B. Schoenfeld, "Squatting kinematics and kinetics and their application to exercise performance," *The Journal of Strength and Conditioning Research*, 2010.

[17] S. Simon, "5 benefits of strength training," *American Cancer Society*, 2019.

[18] P. Srisen, S. Auephanwiriyakul, N. Theera-Umpon, and S. Chamnongkich, "Kinect joints correction using optical flow for weightlifting videos," *IEEE*.

[19] N. Sugisaki, S. Kurokawa, J. Okada, and H. Kanehisa, "Difference in the recruitment of hip and knee muscles between back squat and plyometric squat jump," *Plos One*, 2014.

[20] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.

[21] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose Flow: Efficient online pose tracking," in *BMVC*, 2018.

[22] A. Yasser, D. Tariq, R. Samy, and M. Hassan, "Smart coaching: Enhancing weightlifting and preventing injuries," *International Journal of Advanced Computer Science and Applications*, vol. 10, 2019.

[23] C. Zhi-chao and L. Zhang, "Key pose recognition towards sports scene using deeply-learned model," *Journal of Visual Communication and Image Representation*.