CS231N Final Report: Automating Powerlifting Judging through Keypoint Detection

Rishi Alluri
Stanford University
allurir@stanford.edu

Tarun Kumar Martheswaran
Stanford University
tarunkm@stanford.edu

Rahul Krishna Thomas
Stanford University
rt03mas@stanford.edu

Abstract

Powerlifting is an international strength sport with hundreds of thousands of competing athletes around the globe. As the sport continues to grow in recent years, it has become increasingly clear that powerlifting could benefit greatly from the incorporation of Artificial Intelligence techniques. While past work has demonstrated the power of CNN-based architectures in extracting keypoints from human movement, little to none has been done to apply these techniques to powerlifting meets.

We utilize human movement detection to determine the validity of an athlete's three lifts during a meet: squat, bench, and deadlift. Through the construction of a novel dataset and utilization of both hard-coded feature predictions and neural networks, we validate our results on the human-determined lifting labels. This work is a promising first step in automating the judging process and advancing powerlifting technologically.

1. Introduction & Related Work

Powerlifting is a strength-based sport that focuses on three compound lifts: the squat, bench, and deadlift. Currently, a powerlifting competition, or "meet", is where athletes come together to perform their maximum squat, bench, and deadlift, for just one repetition. Each individual has three attempts for each lift, by which only the maximum valid attempt for each lift is counted. The validity of a lift, in a meet setting, is determined by three human judges positioned at different view angles of the lifter. Each resultingly assigns a white or red light, according to their opinion of whether or not the lift satisfied all definitions of a valid lift [4]. The ranking is then done by summing a lifter's maximum valid attempts.

Currently, powerlifting meets are entirely human-driven, and the sport more generally has a massive lack of artificial intelligence presence. There have been, however, promising studies in computer vision applications to the broad lifting space. Rosenhaim, in his graduate thesis, assessed the quality of several different lifts using Human Action Recognition (HAR), Human Action Prediction (HAP), and Human Action Evaluation (HAE) [10]. Lin and Jian developed the Weight-Training Pose (WTPose) algorithm to determine in real-time if a lift is being done with poor technique. They pioneered a real-time detection and correction system using OpenPose and algorithmic techniques [6]. Chariar and co-authors used MediaPipe to similarly establish key landmarks on a lifter during squat, in an effort to correct form and prevent injury [7]. These studies provide a great technical framework to transfer to professional powerlifting meets. However, they each have their own individualistic data preprocessing, which often comes with specialized cameras and setups.

As competing powerlifters ourselves, we and several other athletes find that the judging in meets can often be highly subjective. So, we propose basing judging off heuristics based on joint tracking. This would eliminate some level of subjectivity that often leads to complaints from athletes, but also can give judges a basis to defend their own decisions. However, every reliable sport automation task begins with a good dataset. The NFL, for example, has several of these datasets [1], but also more niche sports such as badminton have been thoroughly explored with vision [9]. An extensive search revealed none exist for powerlifting, especially ones that can be translated into a real-time judging system.

We begin with the construction of a novel powerlifting dataset - a pipeline from a Youtube live stream to segmented attempts with labeled joint keypoints. This was generated using several vision-manipulation techniques to segment to the best of our ability. Using this unique dataset (termed PL-Vision), we test accuracy based on comparing our generated labels to the judge decisions. The input to our model is then the normalized keypoint coordinates for wrist, elbow, shoulder, etc., across all frames of a lifter's squat, bench, or deadlift attempt. We pass this through a hard-coded heuristic model for an unsupervised approach. We also consider a supervised approach that makes use of data augmentation and a convolutional classifier acting on 4D keypoint data. We then closely analyze the scenarios in which there are discrepancies between our labels and judge decisions, to determine if there is sufficient reasoning for a lifter to contest. Our work provides a great first step to incorporating Computer Vision into the sport of powerlifting. We hope to make PL-Vision public to give AI enthusiasts the opportunity to improve upon our current results.

2. Methods

2.1. Building Novel PL-Vision Dataset

2.1.1 Source of Data

The data used to build the dataset is publicly available YouTube videos of official powerlifting competitions and meets. So far, we have been using the 2024 Powerlifting America Classic Open Nationals videos [11]. Each video contains competitors doing three attempts of squat, bench, and deadlift. The validity of each lift is indicated by at least two out of three white lights received from the three respective judges. Furthermore, on failed lifts, a flag appears under the lights indicating the reason for lift failure. The summary of the flags for each lift is in Table 1, shortened to relevant factors.

Table 1: Summary of flags for each lift

	Red Card	Blue Card	Yellow Card
Squat	Insufficient Depth	Downward Movement	Incomplete Lift
Bench	Insufficient Elbow Depth	Downward Movement	Incomplete Lift
Deadlift	Unlocked Knees, Elbows Not Back	Downward Movement	Incomplete Lift

This is by no means an exhaustive list of rules that lay grounds for red lights: for a full list, see the IPF technical rules [4].

2.1.2 Data Preprocessing

The first step is to downsample the original YouTube videos from the original 25 frames per second to 12 frames per second to reduce computational work for additional downstream preprocessing steps. From here, the next step of our pipeline was to split the full YouTube video frames into individuals' lift attempts. In each professional meet video, the convention is to put the lifter's name in a textbox at the bottom of the screen, including information such as their weight class, current weight attempt, and previous weight attempts which can be seen in Figure 1. As a result, we use PyTesseract, an optical character recognition (OCR) Python wrapper[2]. PyTesseract utilizes Google's Tesseract OCR engine, which employs a deep LSTM neural network architecture to recognize diverse fonts, styles, and layouts. The LSTM model processes the input image sequentially, leveraging its recurrent connections and gating mechanisms to accurately transcribe the text content.

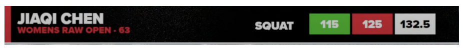


Figure 1: Example of a textbox containing the lifter's name, weight class, and weight attempt information extracted using PyTesseract.

For the next step, each lift needed to be further trimmed down to precise start and end points so that each video only includes the entirety of the lifting motion. When a lift is about to start, the camera angle cuts to the front angle. We created a distance metric leveraging a masked target frame and Mean Squared Error (MSE) calculation on unmasked regions to robustly identify the start frame of weightlifting attempts in video data by finding the frame with the lowest distance from the static background elements of the target frame while ignoring dynamic components like the lifter's body. The end of each lift is marked by the display of the judges scoring lights. Using the same method as described below, when the lights are detected, the endpoint is marked and the video is successfully trimmed.

2.1.3 Lift Validity (Label) Extraction

After each lift, three circular lights appear on the screen, which are white if the corresponding judge deems the lift valid and red otherwise. The lift is valid if it receives at least two white lights. Below each red light, there is also a flag from the colors $\{\text{red}, \text{blue}, \text{yellow}\}$, which indicates the reason for lift failure (e.g. downward motion, improper depth, etc.). To extract the number of white lights, we convolve a square filter over each frame in the video, and mark the locations where the average color across the filter is close to white (under a certain threshold). This approach is meant to account for differing sizes and positions of white lights on the screen across meets and lifts. A visualization of the filter hits on a cropped portion of the video is in Figure 2.

We can further extract flags as follows. We follow the same approach as above, but now mark the locations where the average filter color is nearly red. For each such light location, crop the video to a surrounding region, and then compute the maximum blue and yellow pixel counts (b, y) over all frames in the video. By clustering these pairs across all videos (through a simple KNN with $k = 3$, the number

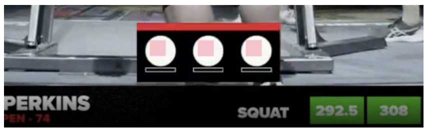


Figure 2: Visualization of the filter hits on a cropped portion of the video for detecting white lights.

of possible failure flags), alongside our computed number of white lights above, we can deduce the number and location of red, blue, and yellow cards.

2.1.4 Feature Extraction: Pose Tracking Keypoints

Body position tracking is crucial to determine the validity of lifts, as it captures relative movements between joints that may violate powerlifting conventions. We use MoveNet, a pre-trained deep network for real-time human pose estimation, to track joints for the duration of the lift[3]. MoveNet utilizes several convolutional layers to extract spatial patterns and time dependency of joints from frame-by-frame human poses, and thereby output relevant pose coordinates for each frame. These coordinates represent specific body parts or joints, as overlayed in Figure 3. Formally, MoveNet outputs a set of 12 keypoints, each represented by a labeled 3D point (x, y, s) corresponding to each joint's position (x, y) and confidence score s. Example plots of keypoint positions (x, y) over time for individual joints are shown in Figures 4(a), 4(b), 4(c). We also visualize all plotted $(x, y, s) \in \mathbb{R}^3$ over 12 example videos in our dataset in Figure 5, where colors represent joints (left hip, left shoulder, etc.). Here, we see pronounced clustering of both keypoint positions and scores across joint classes.

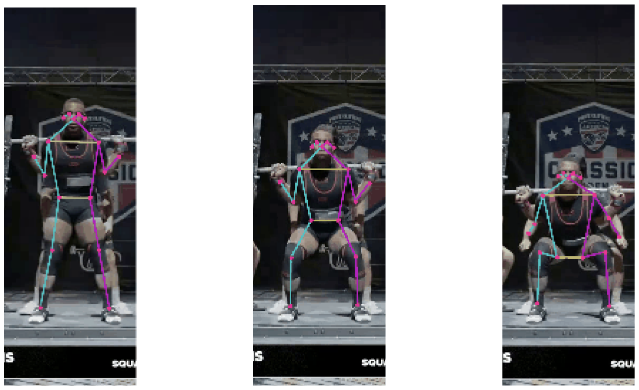
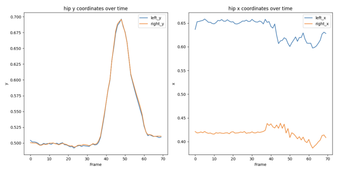


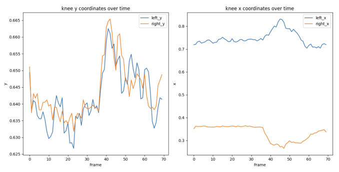
Figure 3: We overlay MoveNet keypoints on a lifter.

2.1.5 Feature Extraction: Barbell Detection Keypoints

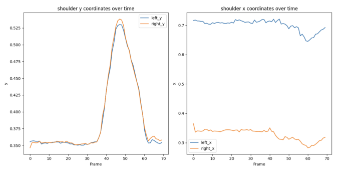
One of the key aspects of lift disqualification is any downward motion of the barbell during the movement. In other words, we need one continuous motion to complete the lift. So, we feel it necessary to extract the coordinates of the barbell during the lift. Currently, we are using a barbell tracker established by Github user Marticles [8]. Here, a bound-



((a)) Tracked hip keypoints



((b)) Tracked knee keypoints



((c)) Tracked shoulder keypoints

Figure 4: We plot tracked keypoints x and y coordinates over time for different joints.

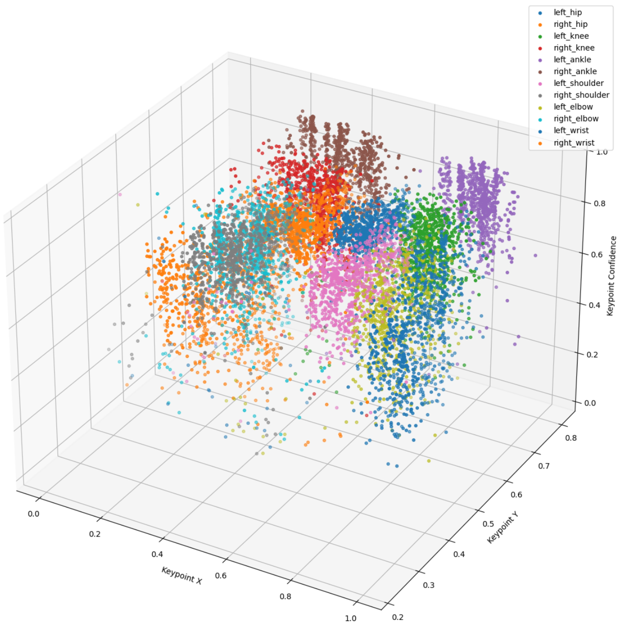


Figure 5: We sample all frames from 12 dataset videos and plot the keypoint coordinates and scores (x, y, s).

ing box is created around the moving object, and the center point is tracked. We currently estimate the center point as a midpoint of joints, and resultingly create a bounding box around this, as in Figure 6.

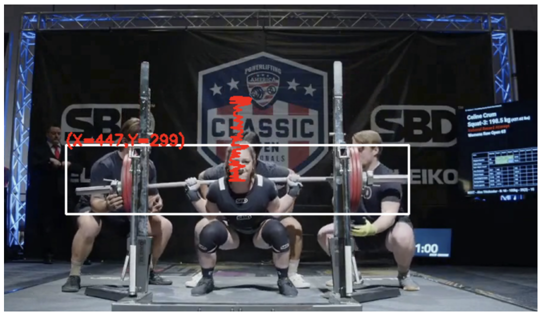


Figure 6: Visualization of the barbell tracking.

2.2. Lift Validity Prediction

2.2.1 Hard-Coded Heuristic

We formed heuristics for the squat, bench, and deadlift that aligned well with the IPF Technical Rulebook [4]. Here, we define the joint keypoint measurements as each a function of f, the frame number.

Table 2: Names and Variables

Name	Variable
left/right_hip	l_h/r_h
left/right_knee	l_k/r_k
left/right_ankle	l_a/r_a
left/right_shoulder	l_s/r_s
left/right_elbow	l_e/r_e
left/right_wrist	l_w/r_w

Each joint keypoint time series was smoothed with a Savitzky-Golay smoothing filter, which finds convolution coefficients C_i. Each smoothed point in a time series Y_j is defined as a linear combination of the convolution coefficients and a sliding window of the observed keypoint values y across frames:

$$Y_j = C_1 * y_1 + C_2 * y_2 + ... + C_j * y_j + ... + C_w * y_w,$$

where w is the size of our sliding window.

Then we encode the following rules for Squat, Bench, and Deadlift as described in the IPF Technical Rules. **Squat depth** was encoded as

$$l_h{}^y(f) - l_k{}^y(f) > 0$$

and

$$r_h{}^y(f) - r_k{}^y(f) > 0$$

having a solution $a < f < b$, where a, b are two other frames from the attempt. Put simply, we require the hips to cross the knee y coordinate at two points, where the hips remain below the knees for $b - a$ frames. Similarly, the **bench elbow depth** is encoded as

$$l_s{}^y(f) - l_e{}^y(f) > 0.006$$

and

$$r_s{}^y(f) - r_e{}^y(f) > 0.006$$

having a solution $a < f < b$.

Here, we use 0.006 to represent the fact that a parallel elbow-shoulder line is still a valid lift, while a parallel hip-knee line is considered invalid. Visuals of depth on the squat and bench are shown in Figure 7.

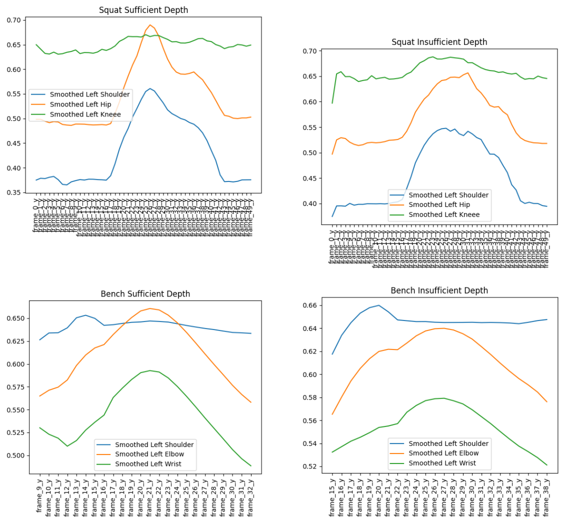


Figure 7: Visualizations of depth on squat and bench

For testing of downward motion across all lifts, we experiment with using barbell tracking and joint tracking. Barbell tracking proved to be far too noisy to differentiate between slight downward motion and simple kinks in our data, so we used the joint keypoint data. On each lift, we use the joint closest to the barbell to track this: squat is shoulder, bench is wrist, deadlift is wrist. One challenge across all three lifts was to distinguish between downward motion and a lift that failed on strength. In the former, the bar will sink down slightly and then continue to move up at a similar speed for squat and bench. Meanwhile, in a lift failed on strength, the bar sinks down and proceeds up at a considerably faster speed, as the spotters make contact with the bar and apply force to move it up to its starting point. So, after the bar reaches hip/elbow depth, we record the average rate of change of the relevant joint until downward motion is found. If the average rate of change of the barbell y position after the downward motion is more than 1.5 times the rate before, this is recorded as a failed lift, or yellow card. Otherwise, this is a blue card for downward motion 8.

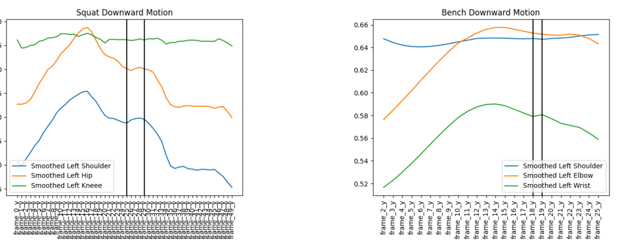


Figure 8: Visualizations of downward motion on squat and bench

In the case of the deadlift, a lift that fails on strength will mean the bar is dropped back to the ground. We hypothesized that this should be much simpler to catch.

Finally, we faced an important design consideration with the center judge. We had left and right joint keypoints, but no succinct way to make a decision on the center judge. As a result, we decided to err on the side of caution and make our center prediction red if at least one of the left or right predictions was red. In the case that one of the left or right predictions was white and the other was red, the center prediction matches the red light, as well as the associated card color.

2.2.2 Addressing Class Imbalances

Hard coding heuristic features is an unsupervised classification approach, so it does not require label data. However, to implement a supervised approach, there are key issues we must address with *class imbalance* in our dataset.

Of the squats, deadlifts, and benches captured in our dataset, over 80% receive 3 white lights, and over 90% are valid. Furthermore, some combinations of flags (e.g. one red card and two yellow cards) never show up, and many more only occur once or twice for a particular lift. This imbalance makes it difficult to train a model that does not simply output valid lifts at test-time.

Unfortunately, due to the infrequency of flags like the blue card (downward motion is rare compared to strength failures) and combinations of differing cards (e.g. it is uncommon for one judge to deem insufficient depth and another to call incomplete lift), there is no easy fix on the data collection side. Thus, we *upsample* minority classes to mitigate the effects of class imbalance. We repeat data points such that each combination of three flags appears around the same number of times across our dataset (after the train-test split, as we only upsample the training set). This does not introduce variation in the minority class, but when combined with another technique, it does: *data augmentation*.

Augmention increases the variability of the upsampled minority classes, and can later improve generalization and robustness of the model. We augment by sequentially applying four transformations: scaling keypoint x values, scaling keypoint y values, translating keypoint x values, and translating keypoint y values. When performing a transformation Γ on an existing dataset D, the new dataset is $\texttt{concat}([D, \Gamma(D)])$. The scales are absolute values of i.i.d. $\mathcal{N}(0, 1)$ samples; the translations are too but are also scaled by the mean of keypoint x and y values. Note the lift flags are *invariant* to these transformations. They are not invariant to scales or translations in the keypoint confidence.

It is worth noting that we can also optionally smooth the keypoint data using the Savitzky-Golay filter. Although this does not address class imbalance, it allows us to better cluster data points. We can see this from forming t-SNE embeddings in Figure 9 on a subset of the training data after using PCA to reduce feature dimensionality to 40, for the data classes $\{\text{naive}, \text{upsample}\}$ and both smoothed and unsmoothed keypoints. Qualitatively, the improved clustering from upsampling becomes clearer after smoothing keypoints.

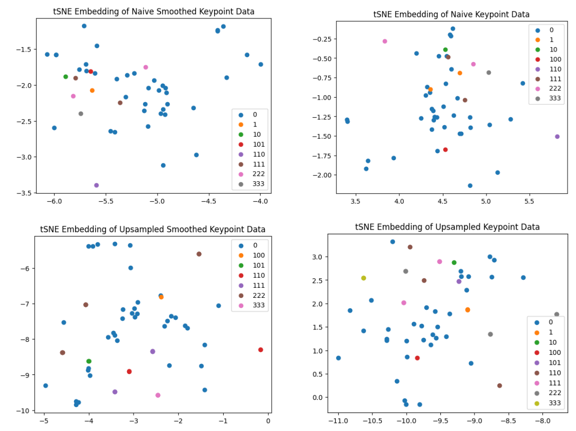


Figure 9: t-SNE embeddings on 4 keypoint data variations.

2.2.3 Deep Learning Keypoint CNN

Our network takes in a $4D$ input of keypoints $K \in \mathbb{R}^{N \times 3 \times f \times k}$, where N is the number of videos (after upsampling and augmentation, as in the previous section), f is the number of sampled frames, k is the number of keypoints, and 3 is included as each keypoint contains an x-coordinate, y-coordinate, and a confidence score (see Section 2.1.4). The goal is to predict three lift flags for each video, which requires an output $F \in S^{3 \times N}$, where $S = \{0, 1, 2, 3\}$ denotes labels corresponding to white (valid lift), red card, blue card, and yellow card, respectively. To achieve this, we independently train 3 deep neural networks $\Phi_i : \mathbb{R}^{N \times 3 \times f \times k} \to S^N$, $i = 1, 2, 3$, to minimize cross-entropy loss between $\Phi_i(K)$ and F_i for each i.

The architecture of each Φ_i is motivated by CNNs: each video has keypoint data of the shape $(3, f, k)$, which resembles the shape of image data, so we infer convolutional layers may perform better than a fully-connected network. This may seem surprising because in practice, $f = 63$ and $k = 12$ are much smaller than width and height of images. Nevertheless, convolutional layers still allow us to capture 2D spatial relationships along the (f, k) axes. Intuitively, convolutional kernels learn relations between keypoint data at neighboring and symmetric joints (given the labeling of joints) as well as relations between keypoint data over frames. Furthermore, applying many convolutional filters allow us to capture different relationships between x-coordinates, y-coordinates, and confidence scores (corresponding to the first shape dimension 3) of keypoints, which can all be relevant in predicting lift validity.

The full architecture is shown below in 10. We pass through a convolutional layer with padding 1 and kernel 3, a max pool with kernel 3, a ReLU, and another convolutional layer with the same padding and kernel. Finally, we pass through two linear and ReLU layers of hidden dimensions 1024 and 256, and then project into an output dimension of 4, the number of classes for flags.

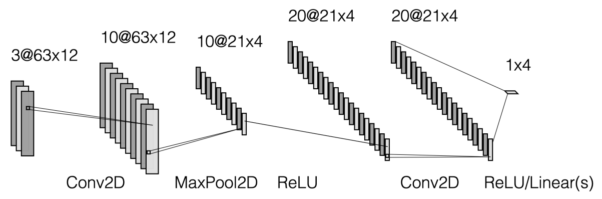


Figure 10: Visualization of the keypoint CNN architecture. Figure produced using Alex Lenail's NN-SVG tool [5].

3. Results & Discussion

Through our data preprocessing pipeline, we obtained videos from 4 different sessions of lifters competing in the Powerlifting America Nationals. The metrics we use are based on the judge-given lights from the left, center, and right. This resulted in ≈ 150 videos for each lift, each with an associated label.

3.1. Hard-Coded Heuristic Results

We calculate 2 metrics: the proportion of total lights correct (PTL), and the proportion of valid/invalid decisions correct (PV). Note for a red light to be correct, we require that our prediction generate the red class, and also the card attached to the red light reliably.

The results from the Squat, Bench, and Deadlift hard-coded heuristics are in Table 3. All results were quite promising. Diving deeper into the lifts our heuristics were not able to reliably predict revealing 3 categories of classification error.

Table 3: Accuracy for each lift

	PTL	PV
Squat	83%	86%
Bench	74 %	79 %
Deadlift	81 %	82 %

First, there were several lifts where our joint key points were convincing enough to display meeting of depth requirements, but the human judges saw otherwise. The opposite was also true in some cases- where the human judges passed the lift but we were not convinced the hip joints crossed below the knees or the elbows below shoulders.

Secondly, the downward motion hyperparameter faced some difficult tradeoff choices. Here, we let n represent the minimum number of frames the lifter was in downward motion for it to be flagged by our model. Making n too small meant we would predict simple keypoint tracking noise as downward motion. However, making it too large would lead to missing out on several lifts that had less obvious downward motion than others.

Finally, there are several cases we missed in which the lifter did not properly follow the audible commands. When performing the attempt, the lifter must listen to the judge exclaim "Start", "Press", "Rack", etc... in order for their lift to be valid. As we did not factor in the audio in this project, these were often wrongly counted as good lifts.

On a lift-wise basis, the misclassifications squat and bench were split between these three categories. For deadlift, on the other hand, almost all of our misclassifications were when we predicted a successful lift, but the lifter failed on strength with a red light with a yellow card. From just the keypoints, we were unable to determine when the lifter successfully locked his or her knees, or if the lack of strength forced them to drop the barbell early. We furthermore attempted to track when a lifter locked their knees by searching for a point where the slope between the ankle and knee became less in magnitude than the slope between knee and hip. While this physiologically makes sense, the noise in the data made this very unreliable.

Ultimately, the hard-coded heuristic performs well overall, but we must take note of the class imbalance mentioned in Section 2.2.2. Most lifts at National Level Meets, which composes our dataset, will receive three white lights. These lifts often have pristine execution, which means the keypoint tracking will easily display a successful lift. Ultimately, it is the **rare cases** when the lifter fails which the heuristic tends to perform worse, giving us a misleadingly high accuracy. Nevertheless, there were of course many cases where we did predict red lights and the cards attached to them.

3.2. Keypoint CNN Results

We only train our keypoint-based CNN on squat data. The reason is that we observed keypoint contributions across all joints are more impactful for squats than, say, bench or deadlift, where not all joints need to be observed to judge lift validity; and so the relevant features are not of high-enough dimensionality to necessitate a deep network. We trained each of our three networks with a learning rate of 0.01 with Adam and a weight decay of 0.01 for 400 epochs. The training and test accuracy are shown in Tables 4 and 5. We report accuracy for each of the 3 lights individually, as well as accuracy for lift validity, and accuracy based on how many videos had *all three* lights predicted correctly (called full light accuracy). Note that the full light accuracy was not considered as a metric for hard-coded heuristic features, but lift validity was as the PV metric.

Table 4: **Training accuracy** for squat videos.

Keypoint Smoothing	Light 1	Light 2	Light 3	Lift Validity (PV)	Full Lights
No	0.667	0.594	0.580	0.580	0.551
Yes	0.662	0.647	0.632	0.6324	0.544

Table 5: **Test accuracy** for squat videos.

Keypoint Smoothing	Light 1	Light 2	Light 3	Lift Validity (PV)	Full Lights
No	0.750	0.833	0.750	0.833	0.583
Yes	0.833	0.750	0.667	0.750	0.667

We observe smoothing keypoint data gives an increase in training accuracy for most individual lights, lift validity, and all lights. However, the effect is less pronounced on test accuracy, with no smoothing achieving better lift validity accuracy and smoothing achieving better full light accuracy. This suggests keypoint smoothing suffers from overfitting to some degree. In any case, the PV metrics for both smoothing and non-smoothing fall below the 86% accuracy reached for hard-coded squat features (see Table 3). Generally, the unsupervised heuristic features seem to perform better than the deep learning approach here. One possible reason is the utilization of barbell tracking in the hard-coded features: integrating this into the deep architecture is a promising area of future work.

4. Conclusion & Future Work

The construction of the PL-Vision dataset is a promising milestone in the process of automating powerlifting judging with computer vision. Both the hard-coded heuristic and our Keypoint CNN revealed the capabilities of joint tracking and are strong baseline models for this task.

There are several steps in our dataset construction process that could use advancements. We faced the limitation of not being able to reliably predict where exactly the start and stop of the lift were. This meant a lot of the data contained noisy estimates of the lifter's joint keypoints before and after they completed their lift. This included their walk up to the platform as well as any movement they may have had after. These extra frames are hurting our modeling potential, so we hope to focus our efforts on extracting lift start and lift stop as reliably as we can. If we are able to do this, we will have the ability to grow PL-Vision seamlessly, as there are a plethora of live-streamed powerlifting meets on YouTube. As we continue developing our pipeline, we furthermore need the ability to detect the actions of the spotters adjacent to the lifter. If we are able to extract the point where spotters touch the barbell, or the plates on the barbell, we will undeniably know if the lifter failed on strength, versus a case of downward motion.

With a larger dataset and more trustworthy keypoint features, we are optimistic that our current hard-coded heuristic and deep learning model will prove to be successful in transforming the way judging is done at powerlifting meets.

References

[1] bronkscottema. Football players computer vision project. `https://universe.roboflow.com/bronkscottema/football-players-zm06l`, 2023. Accessed: 2024-06-01.

[2] Google. Pytesseract. `https://pypi.org/project/pytesseract/`.

[3] Google. Movenet: Ultra fast and accurate pose detection model. `https://github.com/tensorflow/tfjs-models/tree/master/pose-detection`, 2023. Accessed: 2024-05-17.

[4] InternationalPowerliftingFederation(IPF). `https://www.powerlifting.sport/fileadmin/ipf/data/rules/technical-rules/english/IPF_Technical_Rules_Book_2023__1_.pdf`.

[5] A. Lenail. Nn-svg: Publication-ready neural network architecture schematics. `https://alexlenail.me/NN-SVG/LeNet.html`, 2019. Accessed: 2024-06-06.

[6] C.-Y. Lin and K.-C. Jian. A real-time algorithm for weight training detection and correction, 2022.

[7] A. I. S. S. C. A. M. Chariar, S. Rao. Ai trainer: Autoencoder based approach for squat analysis and correction, 2023.

[8] Marticles. Barbell-path-tracker.

[9] mathieu cartron. Shuttlecock computer vision project. `https://universe.roboflow.com/mathieu-cartron/shuttlecock-cqzy3`, 2023. Accessed: 2024-06-01.

[10] A. L. Rosenhaim. Human action evaluation applied to weightlifting, 2023.

[11] YouTube. Powerlifting squat videos. `https://youtube.com/playlist?list=PL85JoAo_KTFurl_VibCv5xscq_NwmaYa8&si=vBaVZti7Pngdn46O`, 2024. Accessed: 2024-05-17.