# DepthPerception: A Deep Learning Framework to Assess Squat Depth in Powerlifting

Rishi Chandra[1], Michael Tao[2],
[1]Johns Hopkins University, [2]University of California Los Angeles

## SOURCE CODE

https://github.com/rishic3/DepthCheck.git
https://github.com/michaelltao/FrontSquat.git

### PROBLEM BACKGROUND and OBJECTIVE

One of the three events in the sport of powerlifting is the squat. A central requirement for the validity of a squat is for the lifter to descend to an adequate depth [1], determined by the subject's hips descending below the plane formed by their knees, shown in **Fig. 1** below.

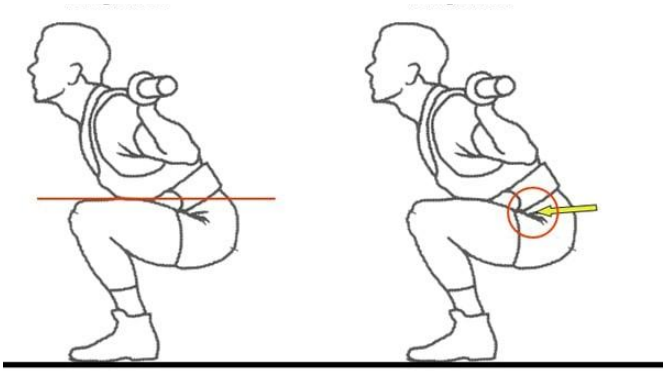

**Fig. 1** - Diagram portraying the USAPL (USA Powerlifting) federation's standards for depth.

Judging squat depth is a difficult task for several reasons. Body structure, including thigh width, femur length, and joint insertions, can differ significantly across lifters, and these visual discrepancies can be subject to human judging bias. The movement often occurs in the space of a few seconds, and the deepest instance of the squat is fleeting and can be difficult to recall, due to blinking or a lapse in short-term memory. Powerlifting environments are often highly distracting, as judges are required to simultaneously give commands, and side spotters can visually occlude the lifter.

Outside of powerlifting, squat depth assessment is fundamental to non-competitive fitness applications. Proper squat depth is crucial to ensure proper range of motion and encourage adequate joint mobility in competitive and amateur lifters alike, as squat depth is an important indicator of mobility [2] and a primary driver of muscle hypertrophy [3].

The overarching objective of our project, aptly named DepthPerception, is to utilize deep learning and computer vision techniques to automatically assess squat depth in powerlifting settings given a squat video input. The framework is intended to replace human judging in powerlifting events, and to be applied in non-competitive gym settings as an assistive tool to encourage proper form.

## INITIAL GOALS and GOALS ACCOMPLISHED

Our initial goals were as follows:

1. At the base level, the completed framework will be able to accurately classify depth from a standardized camera height and angle (ideally, a direct side angle) with no obstructions. This would suit a competitive powerlifting judging application, where the camera angle and field of view can be standardized by the event organizer.
2. We will explore strategies to make the model robust in various video angles and perspectives, as well as against bodily occlusions. This would suit non-competitive applications such as public gyms, where filming from an ideal camera angle without occlusions might not be feasible.

We ultimately accomplished the following:

1. We developed a framework that classifies squat depth of an input video from a standardized camera height and angle (direct side angle) with no obstructions.
2. We developed a framework that classifies squat depth of an input video from varying camera angles and perspectives by mapping estimated bodily landmarks to 3D coordinates.
3. We implemented our own neural network trained on a squat dataset with depth annotations to make depth classifications directly from front-angle pose estimation landmarks, achieving an accuracy of 0.85 on the test set.
4. We estimated a real-world metric for subject displacement from depth by computing a pixel-to-centimeter relationship by relating a user-inputted height and the subject bounding box, produced by an object detection model.

## METHODOLOGY — Pose Estimation Framework

To compute pose estimations for our input video, we utilized two open source pose estimation frameworks, namely the OpenPose model [4] developed by Carnegie Mellon University and Google's BlazePose model [5] within their MediaPipe library. We found that OpenPose's landmark detection was more robust in cases of overlapping body parts and background noise. We found BlazePose to be more robust to motion blur and yielded a faster computation time. Most notably, BlazePose also produces a real-world 3D estimation of pose landmarks, which OpenPose lacks.

Given these findings, we produced two separate frameworks using pose estimation for depth classification, implemented as ***sideAngle.py*** and ***main.py*** respectively in the accompanying source code (see DepthCheck repository).

***sideAngle.py*** utilizes OpenPose to classify depth using the estimated pose landmarks in image coordinates, suitable for squat videos taken at a standardized orientation (camera pitch of zero) around hip height. We parse the input video into frames, compute the estimated landmarks across frames, and identify the frame containing the lowest hip landmark. We then output a depth classification based on whether or not the hip coordinates are below the knee coordinates in the image.

***main.py*** utilizes BlazePose and its accompanying 3D estimations to classify depth in real-world coordinates, suitable for squat videos of varying camera orientation and position. Like the first approach, we parse the input video into frames, compute the estimated landmarks across frames, and identify the frame containing the lowest hip landmark. We then utilize the BlazePose estimation of these landmarks in 3D world coordinates. We compute the plane intersecting the heel and foot landmarks, serving as a base plane of reference. We then

compute the distance from the hip and knee landmarks from this base plane to determine whether the hip descended below the knees, thereby outputting a depth classification.

Both of the above frameworks utilize Yolov3 object detection [6] to compute a depth discrepancy (distance above or below depth) in real-world metrics. The frameworks prompt the user to optionally input a subject height in centimeters. We apply the pre-trained Yolov3 model to detect subjects within the first video frame. We store the size of the bounding box of this subject in pixels, and compute a centimeters-per-pixel relationship by dividing this value by the inputted height. If multiple subjects are detected, we select the centermost subject in the frame for subsequent analysis. This centimeters-per-pixel relationship is used to convert discrepancies from depth to a value in centimeters.

## METHODOLOGY — Neural Network Classification

Our third method employs TensorFlow to build a neural network trained on a powerlifting squat dataset sourced from Kaggle [7] to directly make squat classification predictions. The source code can be found in the FrontSquat repository.

The published dataset consists of 240 videos, half of which were valid, each represented as a directory of frames, ranging from 50-200 frames each. Each frame contains the MediaPipe landmark estimations for the given frame as a Numpy array. The frames are extracted from front-facing squat videos that have been labeled as either valid or invalid squats. We use this annotated dataset to train the neural network to classify squat depth given a sequence of MediaPipe landmarks.

The network architecture is a sequential network comprising 3 dense layers of dimensions 32, 64, and 1. The first two layers use the ReLU activation function, while the output layer uses a sigmoid function to produce a binary output. The network is trained with Binary Cross-Entropy loss and SGD optimization, and achieved an accuracy score of 0.85 over the test set.

To use the network to classify an external video, the video must be processed into the same format as the training set. Thus, we developed an algorithm—see ***getKeypoints.py*** in the FrontSquat repository—to produce a concatenated numpy array containing the X and Y coordinates of each landmark in each frame of the video. This array can then be used as input to the trained model, producing a depth classification for the given video.

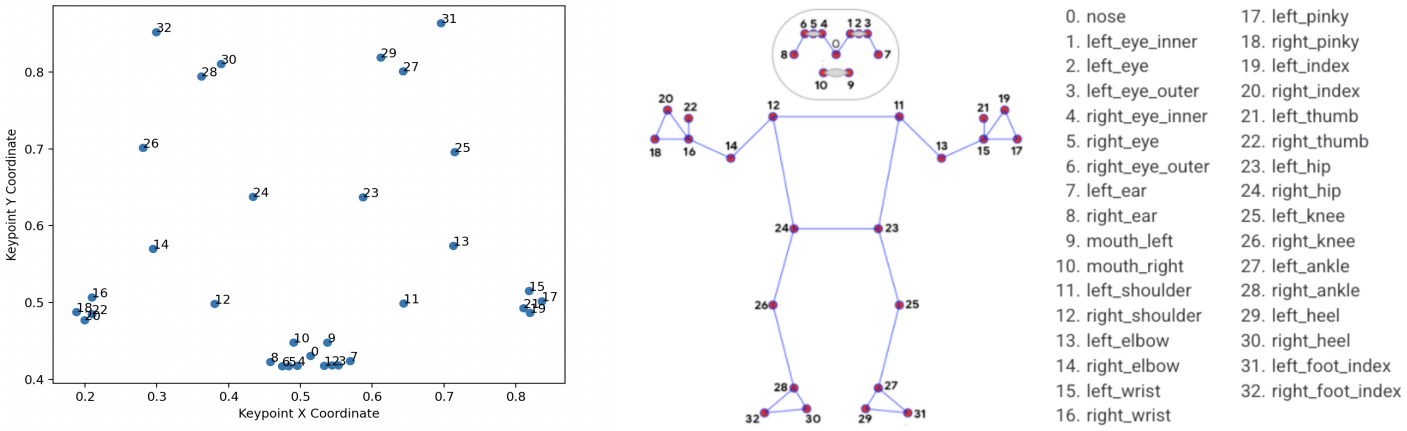


**Fig 2** - Sample plotted depiction of a frame's landmark coordinates (left) and the landmark each number corresponds to (right)

## CHALLENGES and LIMITATIONS

Though our first framework, ***sideAngle.py***, yields consistent and accurate results, it is limited to videos taken from straight-on side-angle perspectives, as it relies on image coordinates for depth calculations.

Our second framework, ***main.py***, relaxes this restriction by computing depth relative to the feet of the lifter, allowing for flexible camera angles and positions. However, this method has its own limitations—erroneous predictions or occlusions of a subject's feet may prevent a reliable calculation of the floor plane. Additionally, we rely on BlazePose's predicted mapping of landmarks in real-world coordinates, which is subject to noise, such as background clutter and bodily overlap.

Our neural network implementation was designed to improve upon ***main.py*** by foregoing the need to compute the base plane of the subject and map landmarks to real-world coordinates, instead making a direct classification from the landmarks of each frame. The resultant model is unconcerned with noisy foot predictions or incorrect 3D mappings, and can independently identify the deepest frame and learn the features of a valid squat. However, the dataset only contains videos taken at a direct front angle, and is therefore limited to classifications on front angle squat videos. Additionally, the dataset is relatively small and the model is therefore prone to overfitting.

## FUTURE IMPROVEMENTS

The main limitation of our neural network implementation lies in the dataset. Although there are many widely available fitness and squat datasets, few to none are labeled with depth classifications for powerlifting applications. The Kaggle dataset we employed in this project contains the output of MediaPipe's pose estimation solution rather than actual videos of squats, which is still limited by the accuracy of the pose estimations.

The foremost next step for our project would involve curating a large annotated dataset of squat videos, containing varying subjects, camera angles, occlusion conditions, etc., each labeled as depth or not depth. We may then train a convolutional neural network to classify the depth of an input video directly. This foregoes the need for pose estimation and coordinate geometry, allowing the model to independently learn the relevant features that determine a valid squat in the presence of background noise and varying camera angles.

For the architecture of this proposed model, we could make use of a pre-trained pose estimation model such as OpenPose or BlazePose, freezing the convolutional feature encoding layers, and changing the output layers to produce a depth classification on the input rather than estimating landmark coordinates—thereby leveraging the learned features regarding bodily position and applying them to make a depth classification.

Additional changes that would improve the accuracy and robustness of the framework include:

1. Detecting prominent features of the lifter—i.e. knee sleeves, squat suit, etc.—in inputs with multiple subjects to select the appropriate subject for depth analysis
2. Image segmentation to isolate the appropriate subject, making the model more robust to background clutter and visual occlusion
3. Utilizing multiple camera angles to reconstruct a 3D representation of the subject, creating a more robust 3D mapping of the subject in world coordinates for depth classification

## INDIVIDUAL CONTRIBUTIONS

Rishi was primarily responsible for the pose estimation framework source code, including the object detection, video parsing and depth plane calculations. Michael was primarily responsible for implementing and training the neural network. Both members were equally involved in project ideation, code debugging, data curation, and model evaluation across both repositories.

## REFERENCES


1. "USAPL Rulebook Home USA Powerlifting." ***USA Powerlifting Technical Rules***, 2021, www.usapowerlifting.com/wp-content/uploads/2021/04/USAPL-Rulebook-v2021.1.pdf.
2. Endo, Yasuhiro, et al. "**The Relationship between the Deep Squat Movement and the Hip, Knee and Ankle Range of Motion and Muscle Strength**." *Journal of Physical Therapy Science*, vol. 32, no. 6, 2020, pp. 391–394., https://doi.org/10.1589/jpts.32.391.
3. Kubo, Keitaro, et al. "**Effects of Squat Training with Different Depths on Lower Limb Muscle Volumes.**" *European Journal of Applied Physiology*, vol. 119, no. 9, 2019, pp. 1933–1942., https://doi.org/10.1007/s00421-019-04181-y.
4. Cao, Zhe, et al. "**OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields.**" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, 2021, pp. 172–186., https://doi.org/10.1109/tpami.2019.2929257.
5. Bazarevsky, Valentin, and Ivan Grishchenko. "**On-Device, Real-Time Body Pose Tracking with MediaPipe Blazepose.**" *Google AI Blog*, 13 Aug. 2020, https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html.
6. "**Predict with Pre-Trained YOLO Models.**" *03. Predict with Pre-Trained YOLO Models - Gluoncv 0.11.0 Documentation*, https://cv.gluon.ai/build/examples_detection/demo_yolo.html#load-a-pretrained-model.
7. Aboosalih, Ayoob. "**Powerlifting Squat Dataset.**" *Kaggle*, 7 May 2022, https://www.kaggle.com/datasets/ayoobaboosalih/powerlifting-squat-dataset.