Assessing Pull-Up ROM with Deep Learning Pose Estimation

Cameron Boydd

Abstract

The goal of this project is to estimate if a pull-up taken from video is a full range-of-motion or "perfect" pull-up. This project utilizes a pre-trained deep learning human pose estimation model to extract key points and assess the form of the pull-up exercise. By analyzing the 2-dimensional spatial relationships between these key points and using vector arithmetic and trigonometry, the program calculates the angle of arm extension and flexion. To ascertain the completeness of the pull-up movement, the program compares the calculated arm extension or flexion angle against a predefined threshold indicative of full extension or flexion. A "perfect" pull-up is completed when the subject is in the bottom state in full-extension, moves to the top state at the predefined threshold for flexion, and back to the bottom state in full-extension.

1. Introduction

Strength training is important for enhancing muscular strength and overall well-being. Among exercises, the pull-up stands out as a fundamental compound movement that targets multiple muscle groups, including the back, shoulders, and arms. However, doing a pull-up with proper form, characterized by a full range of motion, can be challenging.

This program addresses the challenge of assessing pull-up form using human pose estimation techniques. The objective is to accurately detect key body points, such as the shoulders, elbows, and wrists to determine the completeness of each pull-up.

To achieve the objective, the program utilizes the pre-trained deep learning model "Caffe," trained on the COCO dataset. The COCO dataset offers a large number of annotated images providing detailed keypoints for human pose estimation.

The experimentation started with custom data collection and annotation with a small dataset of people performing pull-ups in varying conditions. However, due to limitations in model training the approach switched to utilize a pre-trained model for improved performance.

As a result, the project demonstrates success with the system successfully processing input frames and generating heatmaps indicating body keypoints. By analyzing the spatial relationships between these keypoints, the system calculates the angles of arm extension and flexion, enabling real-time assessment of pull-up range-of-motion.

In the subsequent sections of this report, it delves into the technology, experimental setup, results, and conclusions.

2. Related Work

This project used a couple of open-source materials. The open-source materials used are OpenCV, Numpy and a pre-trained deep learning model called "pose-iter-440000.caffemodel."

2.1. OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a wide range of functionalities for processing images and videos. OpenCV is written in C++ and includes interfaces for Python, Java, and MATLAB/Octave, among others. OpenCV was used for many of its functions. In the project it loads the deep-learning model, opens a camera and/or loads a video, reads a frame, sets the input image for the neural network and performs forward pass inference on the input image. It also draws on the output frame for the displayed video. It draws circles representing detected key points, adds text annotations to the frame, draws lines connecting key points to form a pose skeleton and displays the resulting frames.

2.2. NumPy

NumPy, short for Numerical Python, is a powerful library in Python used primarily for numerical computing. In this project, NumPy is primarily used for numerical computations involving vectors and matrices. The coordinates of key points are initially extracted as tuples and then converted into NumPy arrays. This conversion allows easy manipulation and computation of vectors. NumPy arrays support element-wise subtraction, resulting in the calculation of vectors representing the upper arm and forearm. The dot product between the upper arm and forearm vectors is computed using NumPy's np.dot() function. The magnitudes of the upper arm and forearm vectors are computed using the np.linalg.norm() function. NumPy functions np.arccos()

and np.degrees() are used to calculate the angle between the upper arm and forearm vectors in radians and then convert it to degrees.

2.3. Caffe

Caffe (Convolutional Architecture for Fast Feature Embedding) is an open-source deep learning framework developed by Berkeley AI Research (BAIR) and community contributors. In this project, the deep learning Caffe model is used for human pose estimation. Included is the .caffemodel file that contains the trained weights of the neural network and the .prototxt file that describes the network architecture. The input frame is turned into a blob suitable for the model, then a forward pass is applied and saved as the output which contains the predicted key point heat-maps.

3. Data

This project uses the pre-trained deep learning model "Caffe." This model was trained on the COCO dataset. The COCO dataset contains a large collection of images with complex scenes and diverse backgrounds. The images are sourced from everyday contexts covering a wide range of scenarios. Each image in the COCO dataset is densely annotated with bounding boxes around object instances, segmentation masks delineating object boundaries, keypoints indicating specific body joints or landmarks, and captions describing the image content. COCO annotations include keypoints for human pose estimation, specifying the coordinates of keypoints such as joints in each annotated person instance. The COCO 2020 Keypoint Detection Task supplied the COCO train, validation, and test sets containing more than 200,000 images and 250,000 person instances labeled with keypoints.

Source : COCO 2020 Keypoint Detection Task

The formatting of the data includes x number of image files with an equal number of annotation files that correlate to each image file. The annotations in the case of keypoint detection would be a list of x and y coordinates (Ex: "keypoints": [x1, y1, x2, y2, ..., xn, yn]. Using the preprocessed COCO dataset, the model learns to predict the locations of keypoints based on input images, aiming to minimize a loss function that measures the discrepancy between predicted keypoints and ground truth annotations.

4. Experiments

The first experiment started with custom data collection and annotation. Experimenting started with 10 images of people wearing different colors, doing pull-ups in different angles and in different ranges of the repetitions. For each image, a correlating .txt file containing x and y coordinates of the keypoints located at the shoulder joints, elbow joints, and wrists was created. The tool used for annotating keypoints was CVAT, an open-source annotation tool for labeling data. The closest attempt to training a model with custom data was using the YOLO-pose model and training it on this data. This ended up failing. Downloading other models to train and downloading public data sets also ended up failing. This is why a pre-trained model was used. This project required a well-trained model to capture the keypoints of the input and reach the goal of the project.

5. Results

The results of the project align with the goals. The program successfully passes input frames as blobs through the model producing heatmaps as outputs.

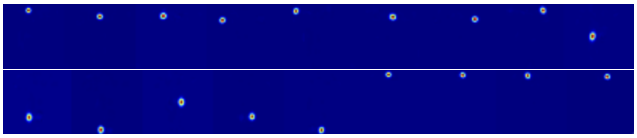

Figure 1. 18 heatmaps indicating 18 different keypoints

Using these heatmaps, the keypoints necessary for pull-up detection are extracted. By analyzing the spatial relationships between these keypoints using vector arithmetic and trigonometry, the program calculates the angle of arm extension and flexion. These angles determine whether the person in the video is in the top state or bottom state of the pull-up. The pull-ups are incremented when the person enters the bottom state from the top state. If the person cuts the repetition short by not moving with full range-of-motion it will not increment.

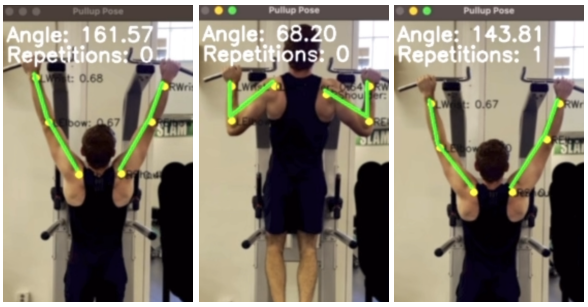

Figure 2. Bottom state, to top, to bottom +1 rep increment

6. Conclusion

The proposed project aimed to leverage human pose estimation techniques to analyze pull-up range-of-motion with the goal of assessing the completeness of each repetition based on the angles of the elbows. By utilizing a pre-trained deep learning model trained on the COCO dataset, the project successfully extracted key joint points from input frames and calculated the angles of arm extension and flexion to determine if the pull-ups were done with a full range-of-motion.

Through the generation of heatmaps and extraction of relevant keypoints, the program calculated the spatial relationships between shoulder, elbow, and wrist joints. By implementing vector arithmetic and trigonometric calculations, the program identified the transition between the top and bottom states of the pull-up.

While initial attempts at custom data collection and model training proved challenging, the decision to utilize a pre-trained model helped achieve the project's objectives. By building upon these open-source resources, the project successfully demonstrated the feasibility of using pose estimation for real-time evaluation of exercise form.

References

[1] Coco dataset: All you need to know to get started. V7. https://www.v7labs.com/blog/coco-dataset-guide

[2] Dmitry Golovkin, "CNN Caffe Models," Available at: https://github.com/foss-for-synopsys-dwc-arc-processors/synopsys-caffe-models/blob/master/README.md

[3] Gholamalinezhad, H., Khosravi, H. (2020, September 16). Pooling methods in Deep Neural Networks, a review. arXiv.org. https://arxiv.org/abs/2009.07485

[4] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T. (2014, June 20). Caffe: Convolutional Architecture for fast feature embedding. arXiv.org. https://arxiv.org/abs/1408.5093