An Evaluation to Correct Improper Yoga Poses Utilizing Pose Estimation

Ziyue Wang[1], Andy Poggio[2]

[1]Henry M. Gunn High School, [2]independent

INTRODUCTION

Yoga is a familiar exercise people do to improve their physical strength and become aware of their own nature. Many people practice it because there is significant evidence for the health benefits of yoga. For example, yoga reduces lower back pain and functional disabilities associated with back pain (Sawyer 2012). People's incorrect yoga poses cause various side effects such as ankle sprain, stiff neck, and muscle pulls (Arushi 2018). Thus, correcting a person's yoga poses becomes important to maintain body health and avoid injuries.

To solve this problem, the software classifier OpenPose (a tool to determine if the pose is correct or not) from Github is used to help identify incorrect forms of yoga poses. Specifically, a technique called pose estimation is used, which can convert people in a video into a skeleton structure. In the structure, the joints--calling keypoints--are highlighted and connected with lines. With this structure, keypoint positions can be extracted. Using a machine learning algorithm, it is possible to identify if the position is correct or not.

DATA AND FINDINGS

The keypoint positions (the positions of joints) and connections are stored in JSON files in an output folder as a large array of data. The data is then normalized, as different images have different sizes. The data was divided by the length of the torso so every value is between 0 and 1. Also, if a keypoint has a confidence of zero, meaning that point is not confirmed by the machine, it is not counted as the part of the data.

The purpose of normalization is to make the data easier to use. After normalization, the data is well prepared. For example, the data of one image looks like this (coordinates x, y, confidence):

The Normalized Data: [0.00000000e+00 0.00000000e+00 0.00000000e+00 9.76734243e-01
6.40010964e-02 9.75391317e-04 9.94255787e-01 6.40453315e-02
9.29657728e-04 1.00000000e+00 9.86485014e-02 7.81688151e-04
9.85483903e-01 9.87512580e-02 1.41105729e-04 9.56485574e-01
6.40015197e-02 9.46670251e-04 9.45108265e-01 9.87984562e-02
6.96449869e-04 9.39256110e-01 1.19006171e-01 1.06352605e-04
9.73883302e-01 1.24805413e-01 8.46727502e-04 9.85578088e-01
1.24833986e-01 8.31537297e-04 9.82710215e-01 1.56477955e-01
7.44130648e-05 0.00000000e+00 0.00000000e+00 0.00000000e+00
9.56573410e-01 1.24754617e-01 8.14907863e-04 9.22010771e-01
1.30506238e-01 2.30701909e-04 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 9.85424641e-01
3.80572114e-02 6.60442711e-04 9.62512342e-01 3.80409143e-02
8.12709865e-04 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

Figure 1: Data showing 2D coordinates of keypoints of the model pictured

Figure 2: Keypoint positions identified for correct and incorrect upward facing dog and plank poses. D (Cao 2017) (Simon 2017) (Wei 2018) (Knoi 2018) (Motz 2014)

ACKNOWLEDGEMENTS/REFERENCES

Special thanks to Andy Poggio for helping to make this project possible.

1. Cao, Zhe; Simon, Tomas; Wei, Shih-En; Sheikh, Yaser; "Real-Time Multi-Person 2D Pose estimation using part affinity fields," arxiv.org; *Computer Vision and Pattern Recognition*, 2017, https://arxiv.org/abs/1611.08050.

2. Simon, Thomas; Joo, Hanbyul; Matthews, Iain; Sheikh, Yaser; "Hand Keypoint Detection in Single Images using Multiview Bootstrapping," arxiv.org; *Computer Vision and Pattern Recognition*, 2017, https://arxiv.org/search/?query=Hand+Keypoint+Detection+in+Single+Images+using+Multiview+Bootstrapping&searchtype=all

3. Wei, Shih-En; Ramakrishna, Varun; Kanade, Takeo; , Sheikh, Yaser; "Convolutional Pose Machines," arxiv.org; *Computer Vision and Pattern Recognition*, 2016, https://arxiv.org/abs/1602.00134.

4. Jake Krol, "10 exellent platforms for building mobile apps", mashable.com, August 17, 2018,

5. Motz, Erin, "Do this, not that:: 5 poses that everyone does wrong and how to fix them (With Pictures)", erinmotz.com, Jul 5, 2014.

RESEARCH METHODOLOGIES

Existing datasets of correct and incorrect yoga poses were collected from images on a yoga journal online. The researcher collected data by taking labeled images of correct and incorrect yoga poses from existing datasets. After transferring the images into my computer, an existing pose estimation program--OpenPose--is used to output the positions of joints as 2D coordinates and the connections of the joints as vectors). Using the existing data, it is easy compare and contrast the traits of correct and incorrect poses.

After this, a machine learning classifier will be built to learn a way to classify the correct and incorrect poses and it is building in progress. Specifically, the classification algorithm will be trained from data. The researcher labeled each training image as correct or incorrect. For the two kinds of exercises that appear later in this paper, the researcher will find the probability for getting the position right or wrong. Then, the machine learning algorithm called anomaly detection is implemented. If the probability is smaller than a certain value, then the pose can be considered incorrect. From the data implemented, the computer can perform a simple regression to find a function that determines if it is true or false. Once the data is trained, a test set can be made to test the accuracy of the training (to prevent overfitting, which means that a function fits the existing data well but cannot find patterns given a new set of data).

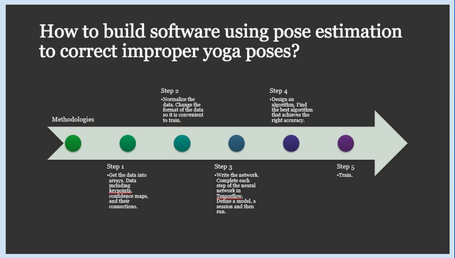

Figure 3: How to build software using pose estimation to correct improper yoga poses?

CONCLUSIONS, ANALYSIS, AND NEXT STEPS

The findings show a clear difference between correct and incorrect poses. The pose estimation algorithm was run on both the chair pose and the plank pose. In the chair pose, the neck should be straight and aligned with the body while the eyes are facing forward. Also, the two arms should be parallel and invisible when looking from the left or right. In contrast, most incorrect poses have eyes facing upward and have an angle between the neck and the torso. In the plank pose, the body should be in a straight line, parallel to the ground, and the upper and the lower handling the body should be in 90 degrees. While in the incorrect pose, it is clear that the whole body (torso and legs) is inclined in a upside-down V-shape. Also, the upper and lower arms are not at 90 degrees.

In the future, the results will be evaluated by a machine learning algorithm; therefore, the training accuracy of the algorithm will be measured in the future. That is to say, the percentage of images that are classified as the same as the initial label will be recorded as an evaluation of my experiment.

The best result will be measured after the training is done. First, I plan to write the source code in python and use it on a laptop computer with GPU to compile and run it. Using the data, the machine learning algorithm can be run with the compiled code to extract features and get the best result.

Next steps include building a phone application that can monitor people's poses when taking a photo. The user can put the phone in almost anywhere to use this app. Then, the user can use the self-capture function to do it.

Figure 4: Future goals are to create a mobile app for greater portability.