Pose Estimation: Final Report

Group D:
Aniket Chandak
Ashraf Saber
David Bui
Deanne Kshipra Charan
Hardik Kumar
Siddartha Thentu

Abstract:

For our project we are tasked with creating an application that involves the computer vision problem of pose estimation. Pose estimation is the problem of being given input data of either an image or video and identifying the human joints found in the input data. There are many subset problems in human pose estimation such as 2D single/multi person pose estimation, 3D single person/multi person, video input and single image input. After looking through the State of the Art (SOTA) approaches to pose estimation we chose the SOTA HRNet and decided to use this as the backbone of this project.

HRNet is the SOTA approach for 2D single/multi person pose estimation on videos and images. We use a simplified code version of HRNET called SimpleHRNet and pretrained HRNet models to perform pose estimation on a yoga poses dataset, then feed those coordinates to train an MLP to classify any “in the wild” yoga pose picture it sees. Using a modified version of a webcam demo in SimpleHRNet we can perform real time yoga pose classification by feeding each frame as a single picture to the trained MLP and outputting a sample yoga pose image of what the MLP thinks the person is doing.

Introduction:

Pose estimation is one of the most important techniques in computer vision which involves identifying poses from input data. The scope of pose estimation is finding the human joints. It is not identifying the human body in the given data. The input data can be in the form of an image or video. As we deep dive into the problem of human pose estimation, we realize that there are different subproblems to solve. Pose estimation can be further divided into 2D/3D single person, multi person, and crowd pose estimation. The other challenge we encounter is the type of data we are working with which vary between images and videos. The research in this field is vast, both in terms of width and depth. For our project we chose one solution to the problem of pose estimation and built a working application around it. Our proposed idea is to classify Yoga Poses. We aim to classify images of individuals practicing Yoga and identify the

Yoga technique in the image using pose estimation. However, first we need to figure out an approach and that is where we started with our literature survey.

Literature Survey:

The aim of our literature survey was to traverse through the breadth of this technique and to provide a rudimentary understanding of various subtopics by examining their respective State of the Art (SOTA) approaches to pose estimation and pick one for our project. For our literature survey we examined the following problem areas for pose estimation, 2D single person, 2D multi person, 3D single person pose estimation on an image, 3D single person pose estimation on a video, and 3D multi person pose estimation. We first start with the discussion of the 2D single person pose estimation.

In their paper, Su et al. [1] summarized the previous work in the area of human pose estimation. Research in the area falls into one of two categories. The first is single-stage pose estimation and the second is multi stage pose-estimation. Multi-stage approaches have higher accuracies as they use multiple layers for pooling and convolution. Su et al. [1] proposed a new method called Cascade Feature Aggregation (CFA) for human pose estimation. This method cascades several hourglass networks for better accuracies. In their paper, Su et al. [1] explain that their suggested method is better than previous work in this area.

Another method of single person pose estimation is discussed in paper [2] where the author talks about solving the 2D pose estimation on motion heavy dataset. The result of the approach is a motion transfer between a highly skilled dancer and an ordinary target subject.

In our literature survey, we explored different problems in pose estimation approaches (2D pose estimation and 3D pose estimation). For 2D pose estimation, some papers discussed the limitations of the existing approaches when a dataset had a property like motion [2] associated with it. Other limitations included detecting multiple person poses in a crowd dataset [3]. The approaches discussed for 3D pose estimation make use of off the shelf 2D key point detectors. Christoph Feichtenhofer and David Grangier’s approach [4] employs a fully convolutional architecture that performs temporal convolutions over 2D key points for accurate 3D pose prediction in video. This approach is compatible with any 2D keypoint detector and can effectively handle large contexts via dilated convolutions. Compared to the approach [5] it provides higher accuracy, simplicity, as well as efficiency, both in terms of computational complexity as well as the number of parameters.

From the 5 papers in our initial literature survey, we chose the 3D human pose estimation in video with temporal convolutions and semi-supervised training paper as the basis for our project. Unfortunately due to data set acquisitions issues were forced to use a different a different pose estimation model for our project. After exploring the other papers in our literature survey we were either unable to properly run the code or unable to acquire the dataset and began a search for another pose estimation approach that had both workable code and an available dataset. Our search lead us to two code repositories for OpenPose and HRNet. Both approaches used the public MSCOCO dataset and we were able to get the code running on both AWS instances and local instances. So, it became a matter of us choosing whichever one was more accurate for us. As the table illustrates [Appendix A], HRNet proved more accurate,efficient and easy to work with the key points. We could visualize the joint predictions on OpenPose and HRNet and the qualitative results are shown [Appendix B]. Since the accuracy and localization in joint prediction were much better with HRNet when compared to OpenPose, we decided to take HRNet as our base model. We found a simplified re-implementation of HRNet called SimpleHRNet which had the code for inference and visualization [10]. Now that we had a state of the art approach for pose estimation with code and pretrained models pre-trained models we proceeded to begin work on the application of our project.

Experiment to train MLP:

The dataset we decided to use is a public yoga pose dataset that contained 10 classes of various yoga poses[9]. However, we only used 8 classes when training our MLP. Using SimpleHRNet code and it’s pre-trained models, we first extract the joint coordinates for each yoga pose from the yoga images dataset using pose estimation. We then normalize all the joint coordinates to make them independent of the original resolution (Check get_keypoiny_data.py for this implementation). The normalized coordinates are flattened into a vector and fed as input to train a multilayer perceptron(MLP)(Check MLP.py program for this implementation). A short definition of a multilayer perceptron is that it’s a basic feedforward neural network. Breaking down the definition of an MLP, feedforward means the edges in the neural network do not form a cycle. A perceptron is a type of linear classifier. So, a multilayer perceptron can be thought of as a non-cycling multilayer linear classifier. Then feed these coordinates(key points) as a vector to a multi-layer perceptron and train the MLP to classify and identify the yoga pose.

GridSearchCV and 5-fold cross-validation:

To find the optimal/best hyperparameters to train MLP, we are using the GridSearchCV library. The library includes formulating a grid having a set of possibilities for hyperparameters. Then the model is trained on each possible value in the grid in each iteration, and accuracy is calculated for each grid value. Parameters are selected which gives the highest accuracy. (Check grid_search.py program for this part)

The best parameters found for our model were as follows:

- activation: *“tanh”*
- hidden_layer_sizes: *“(30,30,30)”*
- learning_rate: *“invscaling”*
- max_iter: *“10000”*
- solver: “*adam*”

The best accuracy achieved is ***0.9332****.*

Moreover, K-Fold cross-validation has also been incorporated in GridSearchCV module. It is used to smoothen out any bias in match data, maximize the number of match scores and avoid overfitting. But as observed in a lot of cases, there can be a lot of imbalances in the data target classes. Hence, *StratifiedKFold* is used. This returns stratified folds where each set contains roughly the same percentage of samples of each target class, in turn warranting the preservation of comparative class frequencies in each train and validation fold. StratifiedKFold with 5 splits has been implemented while training our model.

YogAI Application using trained MLP:

With the MLP fully trained we are ready for our application. The aim of our application is to suggest the yoga pose to the user based on his current position. For e.g., If user is doing a Tree Pose, based on the key points MLP model will identify the nearest possible pose. Our application will give a suggestion based on the results of MLP (Check app.py for implementation of this part). SimpleHRNet comes with a webcam program that does real-time pose estimation using footage from the webcam.

For each frame, pose estimation is performed and the results are drawn on the frame itself. The coordinates output from the pose estimation being performed on each frame are fed into are fully trained MLP model and the MLP model outputs a confidence score of what pose is being done in that particular frame. We use that score to output the results as an example image of what yoga pose the person in that particular frame is doing.

Conclusion:

We were able to successfully create a working application of that classifies Yoga Poses using real time webcam footage with good accuracy on the predicted poses. See the GitHub link to see the full code [11]. Follow the instructions to get the application working and do some yoga poses for some good exercise.

References

[1] Zhihui Su, Ming Ye, Guohui Zhang, Lei Dai, Jianda Sheng. Cascade Feature Aggregation for Human Pose Estimation. arXiv:1902.07837v3, 2019

[2] Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros, Everybody Dance Now. arXiv:1808.07371v2, 2019.

[3] 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.

[4] Dario Pavllo, Christoph Feichtenhofer, David Grangier, Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In CVPR 2018.

[5] Federica Bogo and Angjoo Kanazawa and Christoph Lassner and Peter Gehler and Javier Romero and Michael J. Black(2016): Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In: 14th European Conference on Computer Vision(2016).

[6] Abu Hassan, Mohd Fadzil & Hussain, A & Md Saad, Mohamad Hanif. (2018). Polygonal Shape-based Features for Pose Recognition using Kernel-SVM. Journal of Telecommunication. 10.

[7] Lv, Fengjun, Nevatia, Ramakant Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost. Computer Vision -- ECCV 2006

[8] Shahbudin, Shahrani. (2012). A Simplified Shock Graph for Human Posture Classification Using the Adaptive Neuro Fuzzy Inference System. Journal of Information and Computational Science. 9. 2035–2048.

[9] Yoga Pose Dataset https://www.amarchenkova.com/2018/12/04/data-set-convolutional-neural-network-yoga-pose/

[10] Opini Stefan, Simple HR-Net, (2019), Github repository, https://github.com/stefanopini/simple-HRNet

[11] Our github link

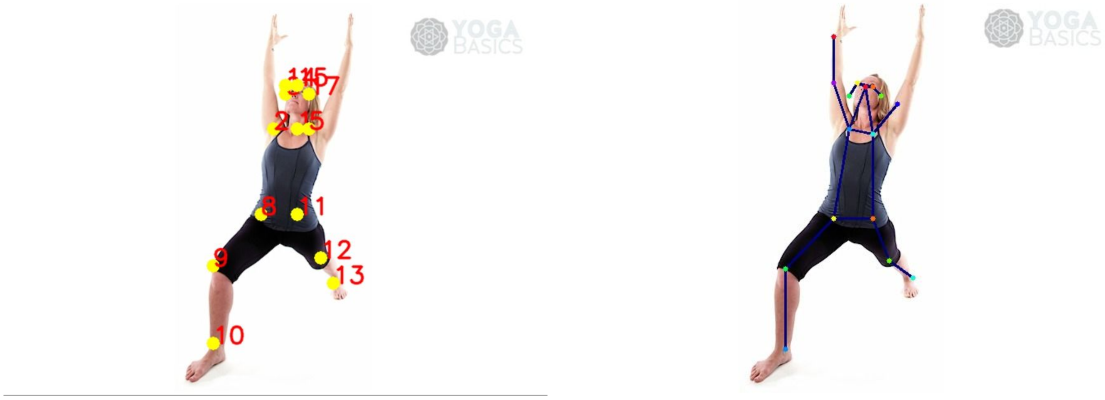
https://github.com/bbdavidbb/YogAi

Appendix

Appendix A
Comparisons of various pose estimation methods

COCO	HRNet	CPN	Open Pose
Average Precision	77	73	65.3
Input Size	384x288	384x288	384x288

Appendix B:
Openpose output vs HR-Net output



Appendix C:
2D Pose Estimation vs. 3D Pose Estimation

2D Pose Estimation		3D Pose Estimation	
2D Single Person	**2D Multi Person**	**3D Single Person**	**3D Multi Person**
● Single Stage ● Multi Stage (Higher Accuracies): Cascade Feature Aggregation (CFA) ● Video: motion transfer between a highly skilled dancer and an ordinary target subject ● HRNet	● 2 Methods: (Top Down/ Bottom UP) ● Crowd Pose ● HRNet	● SMPLify ● 3D Single Person with Video	● DetectNet ● RootNet ● PoseNet