Pose Estimation and Application in Fitness

Song Jiafei, Wang Xin
ShanghaiTech University
{songjf, wangxin1}@shanghaitech.edu.cn

Abstract

More and more people pay attention to health, fitness exercise is one of the most popular ways, but it could be ineffective and potentially dangerous due to the incorrectly pose performed by users. There are many tips in different exercises, you can hire a fitness coach or learn the physical exercises knowledge online by yourself, yet it is too expensive and complicated. Here, we introduce Pose Corrector, an application to simplify all those procedures, users only need to upload their exercise videos and it can detect the users' exercise poses, point out the mistakes and provide personalized recommendations for every turn of actions. Pose Corrector uses the state-of-the-art technology openpose to detect users poses, then utilizes the key points vector, evaluates the extracted features geometrically and heuristically to give useful feedback. We record a data set of over 17000 frames of correct and incorrect pose in 22 videos, and implement geometry and dynamic time wrapping algorithms for evaluation. It currently covers five common exercises.

1. Introduction

Human 2D pose estimation is a popular and fundamental topic in computer vision. Given a single RGB image, we wish to determine the precise pixel location of important keypoints of the body. Knowing the body structure of human can be beneficial for further high-level computer vision tasks, such as abnormal behavior detection and so on. What's more, there are a lot of applications of pose estimation in human-computer interaction and one famous example is Kinect which is produced by Microsoft for Xbox 360 and Xbox One video game consoles.

And in this project, we will apply pose estimation to the fitness. In strength training and fitness, for example, the squat is a compound, full body exercise that trains primarily the muscles of the thighs, hips and buttocks. You should be very careful about such exercises, wrong actions will do lots of harm to your body and improper actions will have no effect on your muscles. Hence, if we can record the pose of your coaches and yours, it will be more convenient and easier to adjust your actions in daily training with such kind of system.

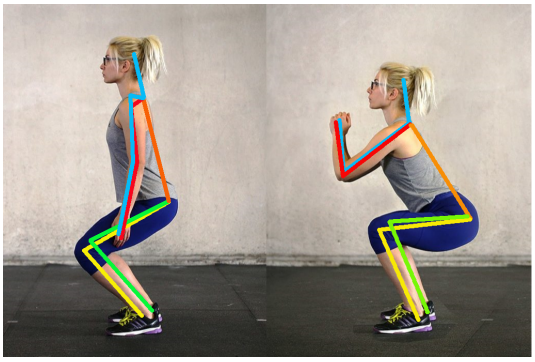

Figure 1. Pose estimation of squatting.

Our purpose is to establish a system to help beginners to learn and correct their actions instead of hiring a fitness trainer in high price. We will try our best to focus on each detail in one action. What's more, we will involve 5 main actions which exercising 5 main muscle group: thigh, chest, back, shoulder and bicep.

2. Related Works

In this part, we will show some paper we read and some related works. Since the method of deep learning has gained such good performance, some traditional [1] methods of pose estimation have been gradually replaced. Recent pose estimation systems have universally adopted ConvNets as their main building block to replace the hand-crafted features and graphical models.

Newell [2] proposed a very interesting network, named Stacked Hourglass Networks. It allows for repeated bottom-up, top-down inference across scales. Also, it used residual module to deal with the vanishment of gradient in such deep networks.

Based on above Stacked Hourglass model, Xiao Chu [3] added multi-context attention mechanism both globally and locally to it. Attention model was quiet reasonable here and achieved some improvements but only a little.

Figure 2. Pipeline of the whole system. After uploading the fitness videos to our system, we first use openpose to extract the skeleton of the body. Then we will evaluate each action and automatically give corresponding feedbacks with the frame of the wrong actions.

Megvii[4] done by Face++ is the leading team in COCO Keypoint Challenge17. They proposed a very complicated network to integrate high resolution information from the earlier layers and semantic information from the deeper layers. Good performance always comes with elaborate model.

One more exciting implementation of single-person pose estimation is [5] done by MSRA. They ranks second after Face++.Their network is quiet simple and only contains a pre-trained 10-layer Resnet and three deconvolution layers. It amazed me that such a simple network can achieve such remarkable performance and also inspires us that how important the features are in pose estimation. And we focus on this paper and do some experiment in milestone.

Another excellent work is named Openpose [6] done by CMU. The performance is quiet good. They combine the confidence maps for part and the part affiity fields together to predict the heatmaps of the joint. The network is also complicated. But the idea of PAF inspires us a lot.

3. Procedure

3.1. Pipeline

Figure 2 shows the whole pipeline for our pose estimator. It includes 5 parts: uploading the videos, keypoints(pose) detecting, extracting useful information, evaluating each action and giving corresponding feedbacks.

3.2. Dataset

We record the first 13 videos of 5 different exercises (barbell curls, dumbbell lateral raise, dumbbell rowing, rope down, squat), including the correct and wrong poses. We store them as MP4 format, 60 frames per second. And then record 9 more videos for testing as a complement. The users can record themselves video of any type of exercises among the function list, theoretically they can use any equipment to record and no requirement for the distance from camera. We recommend you to choose the clearest perspective (side to camera for dumbbell rowing, face to camera for dumbbell lateral raise etc.), and try to keep your whole body visible and inside the picture. Additionally, we might need the user to clip the video on their phones so that it includes only the frames of the action, save it as MP4 format and upload to our Pose Corrector.

3.3. Geometry Basis

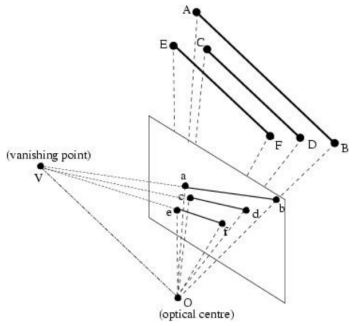

Figure 3. Projecting Geogmetry

While projecting the 3D World to the 2D Image using a pinhole camera, lines which are parallel in 3D space are projected on lines that meet at the horizon. As shown in figure2, lengths and angles will not be preserved after projecting. In our project, we require the people we want to shot with be at the same height of the optical center and the plane of body be parallel to the image plane. Hence, we can get the approximate angle between the limbs for our pose corrector. This is the basis of our whole project.

3.4. Pose Estimator

Before the milestone check, we train the pose estimator using MSRA's simple network, it has already achieved the remarkable performance on our testing videos. After several modifications of network, the accuracy on COCO dataset varies a little. Since the training time is quiet long, then we focus more on our pose corrector instead of further improving the accuracy. After deep consideration, we will take openpose by CMU as our pose estimator.We choose it instead of previous model for learning something interesting in openpose.

As figure4 shows, it applies the bottom-up method, which means detecting keypoints first and then forming the skeleton of each person and the running time will not be increased with the number of people. They combine the confidence maps for each joint and the part affinity fields(PAFs) together to predict the heatmaps of the joint. PAFs is used to

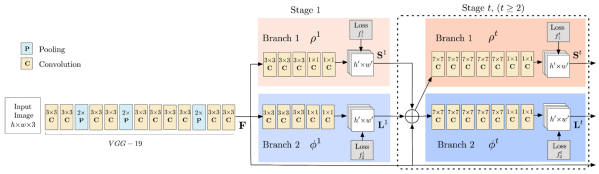

Figure 4. Network of openpose. The upper branch is to calculate the heatmaps of each keypoint, which has 25 in this project. The bottom branch calculate the PAF, which means the direction of the line between two keypoints. Before being sended to the next stage, the input and the upper and bottom output will be concated together.

describe the direction of each pixel in the skeleton. What's more, openpose provides an convenient Python API for getting the keypoint of person.It can both use CPU or GPU, which is very convenient for application.

In this project, we will get 25 keypoints: nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist,middle hip, right hip, right knee, right ankle, left hip, left knee, left ankle, right eye, left eye, right ear, left ear,left small toe, left heel, left big toe, right big toe, right small toe, right heel.

3.5. Dynamic time warping

We have gotten various angle information in each frame, while tracking some angle, we may get the similar pattern as periodic signal. Despite of evaluating the action by some standards, it's also meaningful to compare such pattern with the template action .Dynamic time warping (DTW) is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed and length.

DTW uses dynamic programming to calculate the minimal distance between two series by stretching and shortening the series.The total distance is computed as the sum of absolute distances, for each matched pair of indices, between their values.

DTW has been applied to temporal sequences of video, audio, and graphics data. Hence, in this project, we will try to quantify each part(angle of the limbs) of action and compare with the standard template to determine whether this change of angles is good or bad.

4. Results

4.1. Barbell Curl

Barbell curl is an excellent and classical exercise to train your bicipital muscle of arm. It is suitable for all levels of trainees. The key of such exercise is standing up straight and keeping your upper arm fixed with your body so you will feel your biceps become engaged. While maintaining tension on your biceps, it's also necessary to curl the bar up to shoulder height.

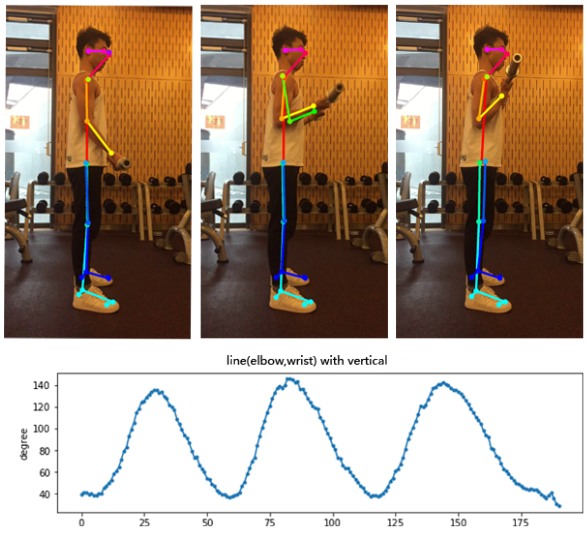

Figure 5. Sequence of barbell curl. Left: start of the whole exercise, you should stand straight. Right: curling the bar up to the shoulder height. Bottom figure shows the periodic variation of the first parameter, this figure includes three turns of such exercise.

The first parameter we are tracking in this project is the angle between the arm(yellow line) with the vertical. This angle can determine whether you have reached the expected height in each turn and can also tell us whether the weight of bar is suitable for you.

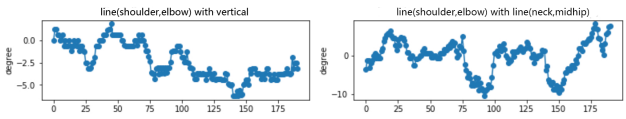

Figure 6. Left: angle between the upper arm with the vertical. Right: angle between the upper arm and the body.

What's more, we also monitor the angle between the upper arm(orange line) with the vertical and the upper arm(orange line) with the body(red line) . See figure 5. These two parameters can determine whether you have kept your body fixed in order to fully engage your muscle. As shown in figure 4, these degrees vary slightly with the time which means you have followed the right instruction.

Finally, we will give the personalized feedback for the trainees,see figure 6. Also we will output the screenshots of the wrong frame in each turn. Such as, if you curl the bar too much, our system will output:*Go up too much! It's better to lower xx degrees"* with the wrong frame for better visualization and understanding of your action.

The whole feedbacks are *"Keep your hands close to your body!"* , *"Keep your body fixed! Do not shake!"* , *"Go up too much!, it is better to lower xx degrees!"*

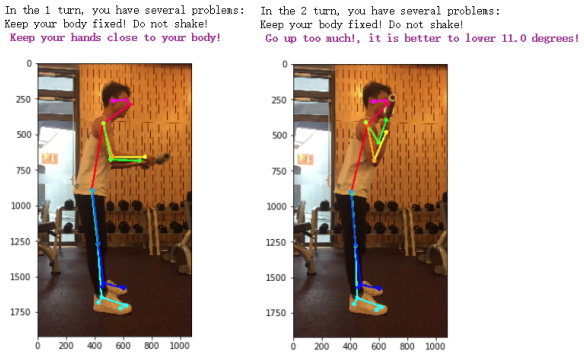

Figure 7. Several examples of wrong screenshots of wrong actions. Our system will give feedbacks automatically with the frame of wrong actions.

4.2. Squat

The squat is a compound, full body exercise that trains primarily the muscles of the thighs, hips and buttocks.However, squatting too much and using over-weight barbell will be greatly harmful for your knee and spine.

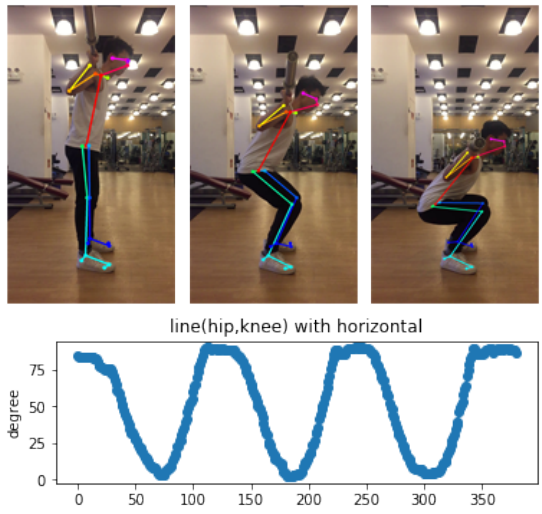

Figure 8. Top: procedure of squatting(from start to the end). Bottom: monitoring the angle between the thigh(green line) and the horizontal, which including three turns of exercise.

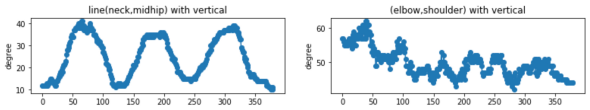

Figure 9. Left: monitoring the angle between the back and the vertical. Right:monitoring the angle between the arm with the vertical.

In figure 7, it means you should squat until your thigh is parallel to the horizontal. In figure 8, it means your arms should be fixed with the body and you should leaning forward a little while squatting.

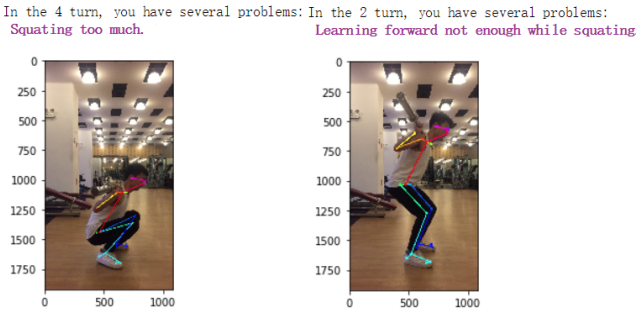

Figure 10. Wrong Actions.Same output format with barbell curling.

We show the sample feedbacks for squatting. The whole feedbacks are: *"Leaning forward too much/not enough while squatting."* , *"Squatting too much/ not enough."* and *"Keep your arms close to your body!"*

4.3. Rope Pulldown

Rope Pulldown is a classical exercise to train the triceps muscle of arm. You should keep your arms fixed with your body and use your forearm to pull down the rope. Since the

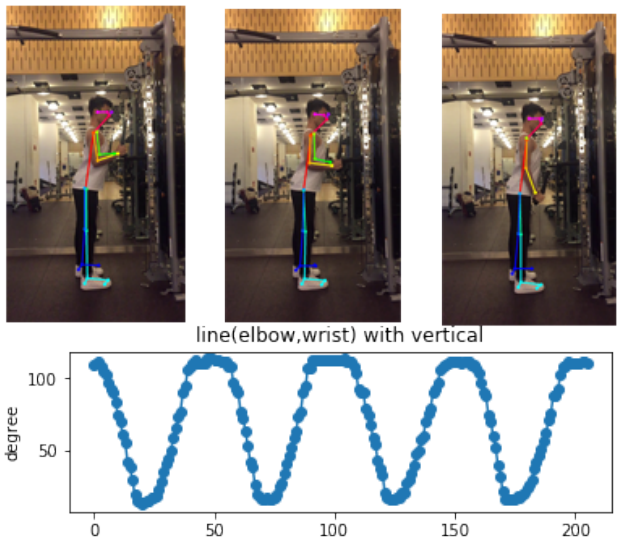

Figure 11. Rope down.Including 4 turns of exercise.

procedure is similar to the barbell curl, hence I will not show more details about such exercise. What's more, the whole feedbacks are: *"Keep your body fixed! Do not shake!"* , *"Leaning too much! Straight a little!"* , *"Leaning your back*

a little!" and *"Go up too much!, it is better to lower xx degrees!"*

4.4. Dumbbell Lateral Raise

Dumbbell Lateral Raise is mainly to exercising the medium bundle of triangle muscle, and it can also benefit to the front and back bundle. The key of this exercise is keeping your feet distance equal to your shoulder width, keeping your arms around 160 degrees to maintain tension on your muscle, dont shrug your shoulders and dont let your elbow above your shoulder.

Figure 12. Top: the procedure of dumbbell lateral raise.

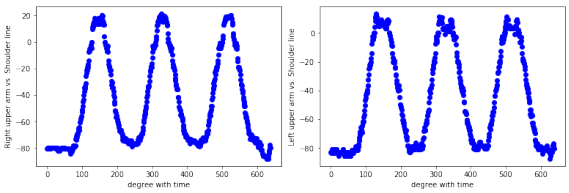

Figure 13. The angle of the both left and right arms with the vertical which includes three turns.

The first parameter we are tracking is the angle between upper arm (both left and right) and shoulder line, see figure 12.This angle can detect whether you stop your raising in the proper position, and tell us whether you let your elbow higher than your shoulder. For the next three parameters ,for the limitation of paper, we will not show the results.The second parameter we monitor is the angle between upper arm and forearm (both left and right), this angle to judge whether you keep your arms straight, empirically speaking it is better to keep it around 160 degrees.The third parameter is the feet distance, we compare it to its own shoulder width, and give according feedback of different error.The fourth parameter is the speed, we monitor your speed of every turn of your action, if you perform too fast or too slow may due to the improper dumbbell weight.

The feedbacks are *"Don't let your right elbow higher than your shoulder!"* and some distance comparison of feet *"God, I can almost build a Great Wall between your legs"*

4.5. Dumbbell Rowing

Dumbbell rowing is to exercise the back muscle, it looks like rowing the boat, we detect the wrist start point y value to determine which side of arm is exercising, the key of this exercise is leaning forward and keeping your upper body fixed, abdomen in, chest cast, and waist straight, keeping your upper arm close to your torso while exercising, pulling up to the top limit position to feel your back tighten up, stay for a few minutes and then put down slowly until your arm perpendicular to ground. For the limitation of the space, we will only describe the parameters we are monitoring without showing the image.

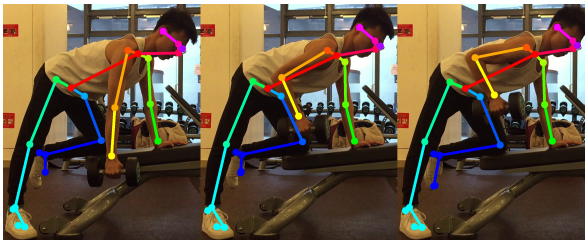

Figure 14. The procedure of Dumbbell Rowing.

The first parameter we are tracking is the angle between torso and the horizontal line, if you dont keep it parallel to the horizontal line within a threshold, we will give you a warning (there also exists the lateral type).The second parameter we monitor is the angle between your upper arm and the horizontal line, to monitor the maximum angle you pull up and the minimum angle you put down. The third parameter is the angle between your upper arm and forearm, this angles minimum value should stay in a reasonable interval, too big means you didn't put in place, while too small means you rotate the dumbbell.The forth parameter is the speed.

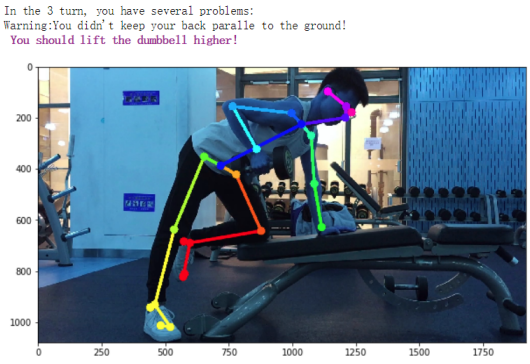

Figure 15. Wrong actions with feedback and screenshot of Dumbbell Rowing.

Figure 15 shows the feedback with screenshot of wrong action. The other feedbacks are: *"Keep your body fixed! Do not shake!"* , *"The angle between your upper arm and forearm is too small/big!"*, *"You should put the dumbbell lower to let your muscle stretch more!"* and the speed feedback *"Too fast!"*

4.6. Results of DTW

For the limitation of space, in this part, we only take squat and barbell curl into account. We calculate the DTW distance with the true and wrong action and assign the label with higher score. Finally, we use confusion matrix to measure the performance of the model.

Squat	precision	recall	f1-core	support
Bad	1.00	0.50	0.67	6
Good	0.75	1.00	0.86	9
Barbell curl	precision	recall	f1-core	support
Bad	0.85	0.65	0.73	17
Good	0.70	0.88	0.78	16

Table 1. Classification of above two actions.

As shown in the above table, we can find the classification by the DTW algorithm works to some extent. But there are still some shortcomings, how many parameters used here will influence the results and how to choose the main parameters is also a problem. Hence, we just have a try to quantify the action. The more will be discussed in the following part.

5. Discussion

Since the core of our project is the application of deep learning to solve the practical problem.The purpose of our project is to help evaluate the quality of fitness. Hence, we do not pay much attention in improving the accuracy of the network. The pre-trained model is quite accurate in generating the keypoints for our project. In conclusion, Pose Corrector can fully use the output of the pose estimator, evaluate the action in both geometry and quantization ways, the good performance in non-labeled test dataset prediction validate this project again. As a result, it can provide feedbacks precisely to every turn, users not only receive the personalized feedback, but can also see the moment picture.

But we still try to understand and reproduce two interesting network. As discussed above, the MSRA constructs the network with the ResNet and three deconvolution layers and achieves the outstanding performance. The key point of this network will be that the keypoints may be among the features extracted by the ResNet. In another network of openpose, it uses direction vectors to help predict the keypoints.

As proposed by teachers, use the 3D keypoints to evaluate the actions of fitness may be reasonable. The complexity will increase greatly and I think the evaluation will be much scientific and accuracy.

What's more, as a idea, we can train a multi-task neural network to get the keypoints and get the remark of each action at the same time instead of solely comparing with the standard template.

And the application range is limit, we can also add more actions and more judge modules to extend its field, update the video auto-clip function, calorie calculate function and even a user-friendly UI.

6. Contributions

We both do survey on related works of keypoint detection and run the network with various parameters. What's more, we both collect the 22+ videos about fitness. Since we will evaluate 5 kinds of exercise, Jiafei Song finished the first three exercises and Wang Xin finished the last two exercises and Dynamic time wrapping algorithm. We contributes to this project equally including the codes,presentation and report.

The whole code and project is on the:
https://github.com/clay001/CS280-Pose_Corrector.git

References

[1] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[2] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[3] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. *arXiv preprint arXiv:1702.07432*, 1(2), 2017.

[4] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018.

[5] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *arXiv preprint arXiv:1804.06208*, 2018.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310. IEEE, 2017.

Appendix A. Supplementary

Here shows the training detail of the network we reproduced in milestone. It is based on MSRAs network. It uses a resnet152 with input 256x192. Using Adam with learning rate 1e-3.

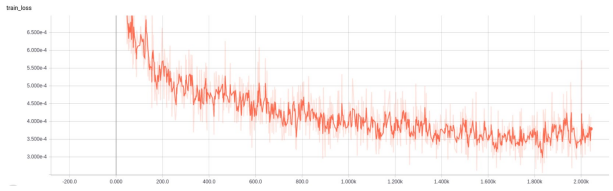

Figure 1. Train loss

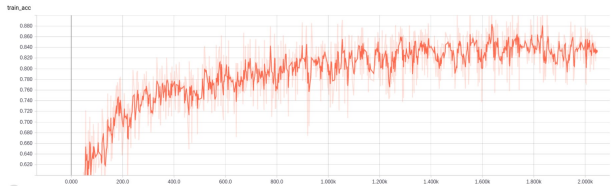

Figure 2. Train accuracy

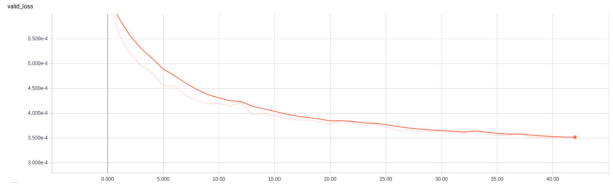

Figure 3. Val loss

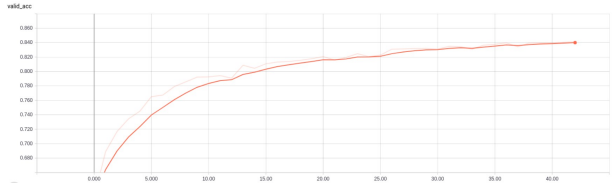

Figure 4. Val loss

What's more, we will also show the testing result on COCO keypoint dataset which is mentioned in milestone report.

Evaluation	AP	AP .5	AP.75	AR	AR .5	AR .75
Mine	0.681	0.904	0.760	0.714	0.911	0.781
Author	0.704	0.886	0.783	0.763	0.929	0.834

TABLE 1. RESULTS ON COCO VAL2017 DATASET