Workout Analysis Using Deep Learning

Neelay Goswami[1] Jainesh Desai[2] Keyur Kalavadiya[3] Shruti Raval[4]
[1,2,3]Student [4]Assistant Professor
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]L. J. Institute of Engineering & Technology, Ahmedabad, Gujarat, India

Abstract— Doing exercise is really crucial for maintaining healthy lifestyle. Presently due to influence of various social media platforms a lot of people wish to achieve a better physique, but a lot of people tend to perform exercises incorrectly which results into damaging muscles where sometime damages caused becomes irreversible to overcome. In the present research, we used the deep learning's CNN algorithm to estimate the human pose and with the use of simple mathematical equation we can check whether the person practicing the exercise correctly or not.

Keywords: Pose Estimation, Convolution Neural Network, Machine Learning, Deep Learning

I. INTRODUCTION

In past, as people were not much in to body building in India but now the number has increased drastically, so people suffering from muscle injury has also increased due to lack of knowledge regarding the posture. But now with algorithms like CNN along with some traditional geometric equation we can do pose estimation and pose correction which can help to reduce the chances of getting muscle injury along with a good body.

II. LIMITATION OF AN EXISTING SYSTEM

There are several systems that can work in this field but they have done the work in mainly improving the pose detection while we have combined pose detection along with pose correction which shows the angle range in between the joints to perfume exercise more accurately.

III. OBJECTIVE

Our main objective is to provide better accuracy to reduce the chances of getting muscle injury while performing a exercise along with that we would like to provide this system in the most economic way compared to regular gym trainer cost as to provide help to humble people as well.

IV. FEASIBILITY

The feasibility refers to the chances of a doable project.

- Technical Feasibility: With the help of advanced technologies like python and Machine learning and useful frameworks like Flask, it becomes possible to implement such a concept.
- Economic Feasibility: The major cost would be related to only model training in cloud other than that there are no major finances required.
- Scheduling Feasibility: It is necessary to understand the timeline of the project. Whether if not completed in time, would it fail?
- Operational Feasibility: Analyses if the project plan satisfies the requirements mentioned in requirement analysis.

V. METHODOLOGY

In this section, we explain our approach in detail. We used Convolution Neural Network (CNN) for checking where the particular pose is properly performed or not. We got the Dataset from the Microsoft COCO Dataset used it for object detection in order to detect human body. The pipeline of open pose to achieve the pose estimation is in the following in ascending order:- (a)Image input , (b)part confidence map, (c)Part Affinity Fields, (d)Bipartite matching , and(e) parsing result .

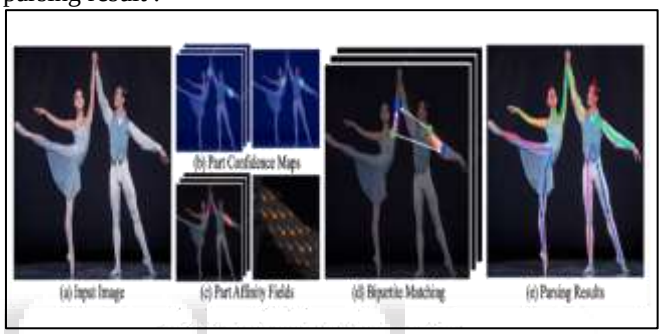

Fig. 1: Pipeline of Open pose

We developed our model by training the dataset using CNN algorithm.

The Architecture of Multi-stage CNN is following:

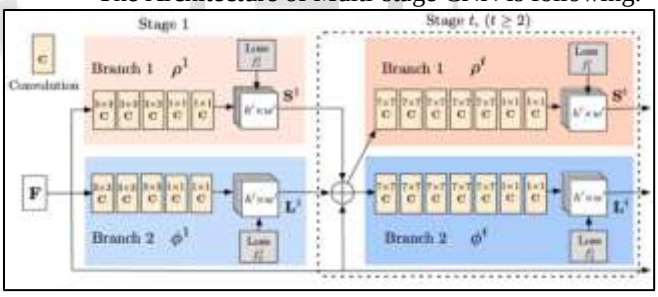

Fig. 2: Architecture of the two-branch multi-stage CNN

Here the Branch 1 (S in red) does the work of predicting confidence maps the right ankle, left ankle, right elbow and others. The branch 2, shown in blue, predicts the affinity fields, which represents a degree of association between different body parts to connect the body parts. As you can see we have here multiple stages so here with each stage until the last stage which is 6th in our project, will increase the affinity fields which results in the detection of various body parts correctly and accurately, which can be understood by diagram below.

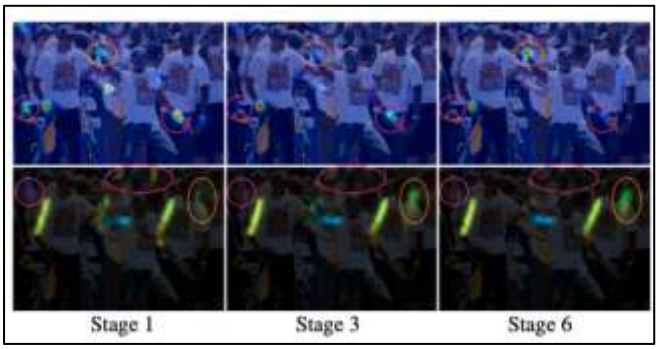

Fig. 3

A. Confidence Maps

Referring back to fig.2, the top branch of the neural network produces a set of detection confidence maps S, this mathematically defined as follows.

$$S = (s1, s2, s3...sj)$$
$$sj \in R^{w*h}$$
$$j \in \{1..j\}$$

where j is the total number of body parts

Part Affinity Fields (PAF) Maps

Referring back to Fig 2, the bottom branch of the neural network produces a set of part affinity field maps L. This is mathematically defined as follows. Mathematically defined as follows.

B. Part Affinity Fields (PAF) Maps

$$L = (L1, L2, L3...Lc)$$
$$Lc \in R^{w*h*2}$$
$$c \in \{1..C\}$$

where C is the total number of limbs

C, the total number of limbs, depends on the dataset that OpenPose is trained with. The paper refers to part pairs as limbs for clarity, despite the fact that some body part pairs are not human limbs. For COCO dataset, C = 19. The figure below shows the different part pairs.

Coco pairs = [(1,2), (1,5), (2,3), (3,4), (5,6), (6,7), (1,8), (8,9), (9,10), (1,11), (11,12), (12,13), (1,0), (0,14), (14,16), (0,15), (15,17), (2,16), (5,17),] (total 18 pairs for body parts and one for blank).

1) Stage 1 in the figure 2 equations:

The network produces a set of detection confidence maps S and a set of part affinity fields L. The symbol **ρ** is used as a function variable which represents the CNN with input F to produce the output map S. The symbol **ϕ** is used as a function variable which represents the CNN with input F to produce the output map L. The annotation "1" at the top of each symbol means inference at the first stage.

$$S1 = \boldsymbol{\rho}1(F)$$
$$L1 = \boldsymbol{\phi}(F)$$

2) Stage t:

The predictions from both branches in the previous stage, along with the original image features F , are concatenated and used to produce more refined predictions.

$$S^t = \rho^t(F, S^{t-1}, L^{t-1}), \forall t \leq 2$$
$$L^t = \phi^t(F, S^{t-1}, L^{t-1}), \forall t \leq 2)$$

3) Loss Functions:

In order for the network to learn how to generate the best sets of S and L, two loss functions at the end of each stage are being added, one at each branch respectively. The paper uses a standard L2 loss between the estimated predictions and ground truth maps and fields. Moreover, some weight is added to the loss functions to address a practical issue that some datasets do not completely label all people. The loss functions at a particular stage t are given as follows.

$$f_S^t = \sum_{j=1}^{j} \sum_{P} W(p). ||S_j^t(p) - S_j^*(p)||_2^2$$

$$f_L^t = \sum_{C=1}^{C} \sum_{P} W(p). ||L_c^t(p) - L_C^*(p)||_2^2$$

The notation p represents a single pixel location in a w x h image. The * notation next to the set S and L means that it is the ground truth The output of S(p) is a 1 dimensional vector which consists of the confidence score for that particular body part j at image location p. The output of L (p) is a 2 dimensional vector which consists of the directional vector for that particular limb c at image location p. In the Open Pose paper, J , the total number of body part is 19. Also, C , the total number of "limbs" or body to body connections is 19.W(p) represents the weighing function as previously mentioned. W(p) = 0 when the annotation is missing at an image location p. The mask is used to avoid penalizing the true positive predictions during training.

4) Final common Loss function:-

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t)$$

Until now the above given info is only about pose detection and we will look into pose correction using geometry equation.

Python code snippet which we have used to find the angle between any three key-points assigned to body parts obtained using above given process:-

```
def get Angle(a, b, c):
ang = math. degrees(math.atan2(c[1]-b[1], c[0]-b[0]) - math.atan2(a[1]-b[1], a[0]-b[0]))
print (ang)
return ang + 360 if ang < 0 else ang
```

Here a, b, c are the co-ordinates of any given 3-body part key points. And we have here applied tan θ inverse formula and then converted the given values from radian to Degree in order to show the values in degree. Here we have set the angle range for a single exercise(Bicep curl)but we can do that for any number of exercises but one has to assign correct angle range for each exercise(No need to train model again for that).

C. Outcome:-

In order to show outcome we have use python Computer Vision Library Open-Cv enabled with Cuda to obtain better frame-rates in output.

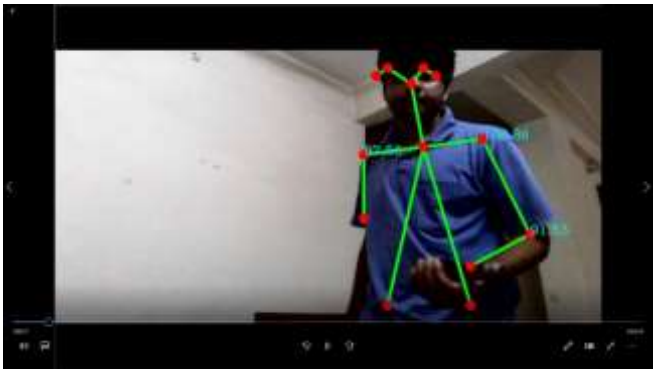

Fig. 4

Correct posture for bicep curl exercise which can be seen in Fig (4)

Fig. 5

Incorrect posture for bicep curl exercise which can be seen in Fig (5)

VI. SUMMARY

The working of workout Analysis using deep learning with use of CNN algorithm is in the following steps:

1) Provide an Image or a Image frame of video
2) Perform convolution on the image and follow through the pipeline of Open pose
3) Here there are two branches in each Convolution one deals with the confidence maps of the various body parts and other deals with the affinity fields between them.
4) The output of each stage becomes the input of the next stage and this output is passed through a loss function which improves the accuracy after each stage. There can be any number of stages in open pose but here we have kept 6 stages.
5) After this we have detected human body with all the body parts but in order to check for the correct posture for any given exercise we have to set pre-determined correct range of angle for any given exercise in order to work correctly.
6) Finally the formula to find the angle between any given three points is applied and using python computer vision library Open-Cv we can show the output.

VII. CONCLUSION

The task of detection and correction using various deep learning algorithms is still in developing phase and far from as a viable option, As with our current trained model, we are achieving 85% of accuracy in predicting the result of whether the user is performing correct exercise or not. Major issue we face is the speed of execution, as now a days people are having camera of very high qualities so in order to process it we need much more powerful computers to make our model flawless and easy to work with.

VIII. FUTURE ENHANCEMENTS

Increasing the accuracy of our project and along with that providing better frame rates is the main focus. We are also planning to make this project available in the Application form on Android Play Store and Apple app store and also as a Web-App so we can reach out and help more people and the ones who don't have access to excellent healthcare.

ACKNOWLEDGMENTS

Authors acknowledge the support provided by the Department of Computer Engineering, L.J Institute of Engineering & Technology, Ahmedabad, Gujarat, India. And Gujarat Technological University, Ahmedabad, Gujarat.

REFERENCES

[1] Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields (2018), arXiv
[2] OpenPose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation
[3] Simple Baselines for Human Pose Estimation and Tracking (ECCV'18)
[4] Deep High-Resolution Representation Learning for Human Pose Estimation (CVPR'19)
[5] Appearance based pedestrians' head pose and body orientation estimation using deep learning Mudassar Raza, ZonghaiChen, Saeed-UrRehman, PengWang , Peng Bao
[6] Real-time Model-based Articulated Object Pose Detection and Tracking with Variable Rigidity Constraints Karl Pauwels, Leonardo Rubio, Eduardo Ros
[7] K. Pauwels, L. Rubio and E. Ros, "Real-Time Pose Detection and Tracking of Hundreds of Objects," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 12, pp. 2200-2214, Dec. 2016, doi: 10.1109/TCSVT.2015.2430652
[8] Application of Machine Vision in Pose Detection of Objects Yuxi Ying[1], Qingqing Shao[1], Xingting Pei[1] and Hongbing Yang[1]

