JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

POSE ESTIMATION FOR FITNESS TRACKER USING MACHINE LEARNING

[1]Surabi. S, [2] Tivekshaa. T, [3]Dr. S. Venkatesh
[1]Student, [2]Student, [3]Associate Professor
[1]Computer Science Department,
[1] Jeppiaar Engineering College, Chennai, India

Abstract : Human Pose Estimation becomes a popular project today in the field of Computer Vision. The human pose estimation can be developed using Artificial Intelligence or Machine Learning. In this work, we have developed a system that can get inputs through live web camera from which the it can localize joints in the human body, detects the pose and provides the repetition counts of the exercise he/she perform. This paper discusses the issues in human pose estimation and gives the overview of considerable research work in pose estimation, including Machine Learning approach. We have used OpenCV and MediaPipe. We conclude with a few promising bearings and directions that have to be explored for future research.

***IndexTerms* – Gym tracker, Machine Learning.**

I. INTRODUCTION

With the abundance of workout videos available on the internet and with popular fitness tracking apps like Samsung Health providing dedicated workout programs section, more people are following these workouts independently. Due to busy lifestyle and financial issues large number of people have adopted working out indoor independently. Although convenient, if the workout is not performed properly it may lead to severe injuries in long term. There is currently no system to tell a user how accurately was he/she able to follow a particular workout and to count the number of reps does he/she does.

Human posture estimation has always been a challenging problem that acquires great attention. It involves huge difficulty for the identification and localization of key points of the body that mainly includes various joints and body movement forecast and also shares difficulties in detection, for example in clustering, lighting, perspective, and scale, are the significant troubles interesting to human postures.

The detection of body key points has been a great problem due to little joints, impediments, and the need to catch content.
The model/network is provided with images as inputs. These images are labelled, and the labelling specifies the action/pose by a human and at the same time it has the positioning of different joints being marked. A typical pose estimation model makes use of 32 joints as the key points.

I. LITERATURE SURVEY

1. Human Pose Estimation Using Convolutional Neural Networks.

This system has a simple model using Convolutional neural network that estimates the poses and demonstrates the potential of CNN. They identify x-y coordinates for 14 body joints by taking a full raw image input. The dataset includes 40,000 images collected through a recognized catalogue of human activities. Pictures are analysed on the basis of annotations including body pose, body part occlusion, torso, and head viewpoint, and personal activity The architecture takes an input of w*h size and producing an output which includes of 2D locations of anatomically key points for every person in the image. The procedure involves feeding of images into feedforward network which predicting a set of locations with a set 2D vector fields having degree association between parts.

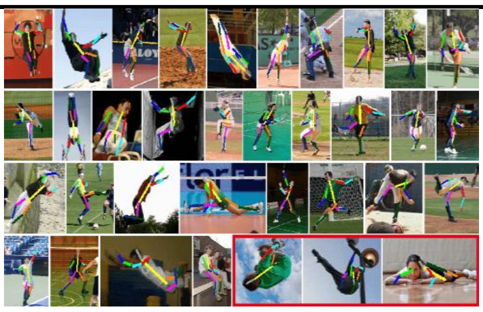


Figure. II.1.a) Pose estimation using CNN

2. Head and shoulder pose estimation using body mounted camera

This system contains a wearable camera-based pose estimation of a user's head and shoulders. The proposed system is mounted on a user's chest and the camera looks upward to observe a user's head and shoulders region. The proposed method is based on histograms of orientated gradient (HoG) features and support vector regression (SVR).

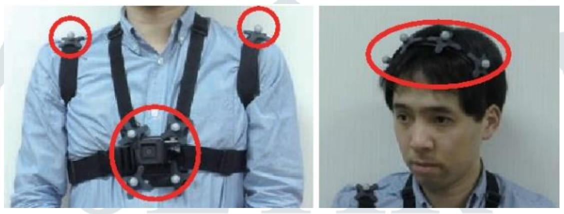


Figure. II.2.a) Pose estimation using body mounted camera

3. Pose estimation in real time using Tri-Line Method

The proposed Tri-Line method is designed to estimate the pose of the face using seven landmarks and its x and y coordinates. They use 7 landmarks and 3 Degree of Freedom namely Yaw, Pitch and Roll.

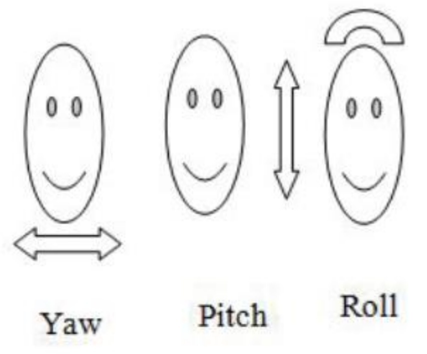


Figure. II.3.a) Degree of Freedom for Pose Estimation using Tri-Line Method.

Face is detected using Viola-Jones algorithm; facial landmarks are located using structured output.

III.IMPLEMENTATION

We have proposed system that comprises of 5 stages/ modules. It includes Making Detections, Determining Joints, Calculate angles, Counting curls. Our architecture work flow diagram is given below.

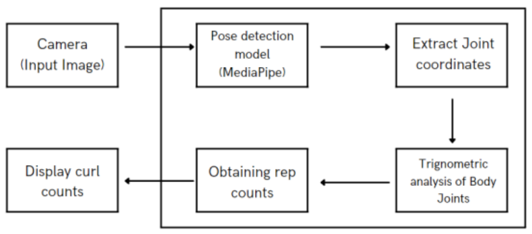

Camera (Input Image)
Pose detection model (MediaPipe)
Extract Joint coordinates
Trignometric analysis of Body Joints
Obtaining rep counts
Display curl counts

Figure. III. a). Architecture Diagram

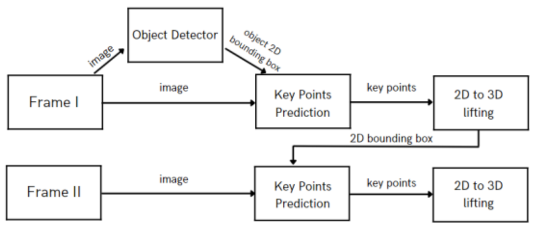


Figure. III. b). Architecture Diagram

IV. PROPOSED SYSTEM

Our proposed system has 4 main stages. The first step is to install MediaPipe. After installing, Recolour of image from BGR to RGB is done. RGB conversion is required because the MediaPipe can take only RGB sources. The detections are made using the webcam on the machine. After making the detections, the image is again converted from RGB to BGR as OpenCV takes BGR source images. After rendering the results, the landmark for each joint is found and the co-ordinate of each joint is also known.

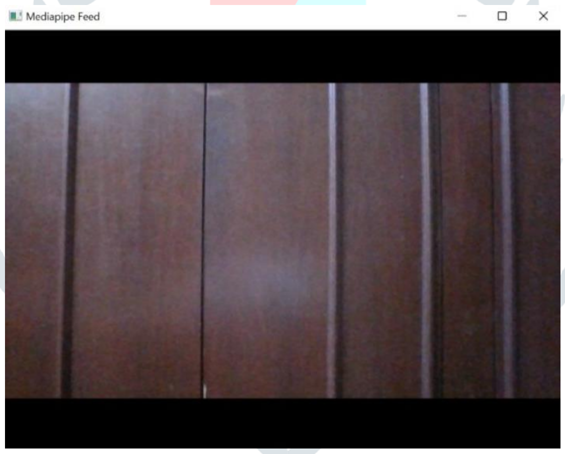


Figure. IV. a). MediaPipe popup

There are 33 landmarks in total, starting from index 0. These represent the different joint within the pose. We then extract the landmarks for our pose. The result will the x, y, z co-ordinates and the visibility.

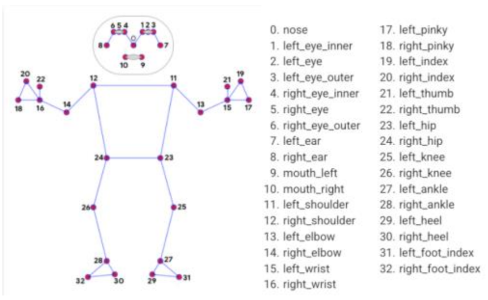


Figure. IV. b). MediaPipe Landmarks

Once the landmarks and co-ordinates are extracted, for example (the angle between the wrist landmark(15) and the shoulder landmark(13) is found). This is done by assuming three variables for each landmark and finding out the radians and the angles.

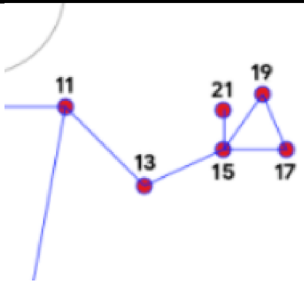
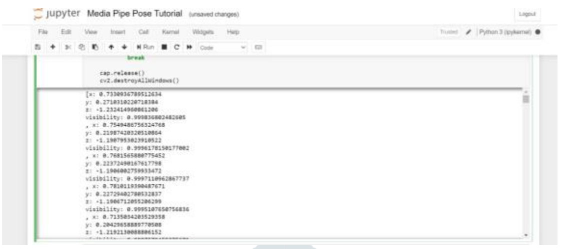


Figure. IV. c). Landmark 11, 13 and 15.

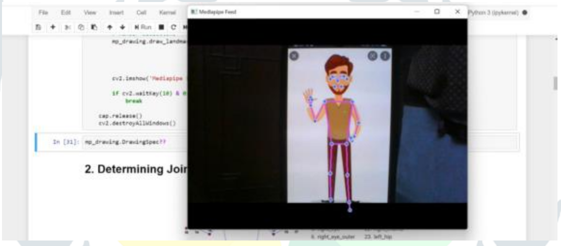

jupyter Media Pipe Pose Tutorial (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3 (ipykernel)

```
            break

    cap.release()
    cv2.destroyAllWindows()
```

```
[x: 0.7330936789512634
y: 0.2710310220718384
z: -1.232414960861206
visibility: 0.9998360802482605
, x: 0.7549486756324768
y: 0.2198742032051086
z: -1.1907953023910522
visibility: 0.9996178150177002
, x: 0.7681565880775452
y: 0.2237249016761779
z: -1.1906002759933472
visibility: 0.9997110962867737
, x: 0.7810119390487671
y: 0.2272940278053283
z: -1.1906712055206299
visibility: 0.9995107650756836
, x: 0.7135034203529358
y: 0.2042965888977050
z: -1.2192130088806152
```

Figure. IV. d). x, y, z co-ordinates of each landmarks

Our next stage is set a threshold. The 160-degree DOWN threshold gives you leeway in case you don't get a perfect 180-degree straight arm. The 30-degree UP threshold ensure a rep that doesn't go all the way up is counted.

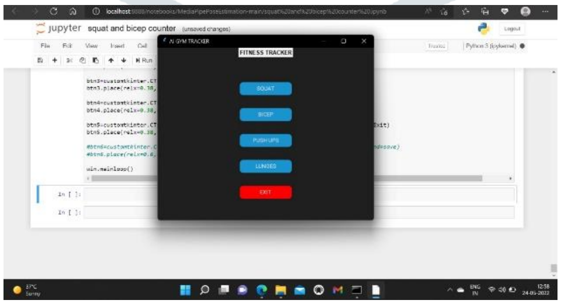


```
            mp_drawing.draw_landma

            cv2.imshow('Mediapipe

            if cv2.waitKey(10) & 0
                break

    cap.release()
    cv2.destroyAllWindows()
```

```
In [31]: mp_drawing.DrawingSpec??
```

Figure. IV. e). 33 landmarks for 33 joints

Now, we render the results on the screen itself by setting up a status box which displays the rep data and the stage data.The rep data hold the title and the counter whereas the stage data holds the title and stage state (UP or DOWN).This is the procedure to find the curl counts for biceps curls.



```
btn3=customtkinter.CT
btn3.place(relx=0.38,

btn4=customtkinter.CT
btn4.place(relx=0.38,

btn5=customtkinter.CT
btn5.place(relx=0.38,

#btn6=customtkinter.C
#btn6.place(relx=0.6,

win.mainloop()
```

Figure. IV. f). GUI for the system

Similarly, the other exercises like squats, lunges, push ups are done. We also constructed a GUI for the system in which the buttons for each exercise are given and then the needed one is chosen.

V. RESULTS

The screen displays the rep data and the stage data (UP or DOWN). It increments the count only if the angle between two landmarks reach the threshold angle value.

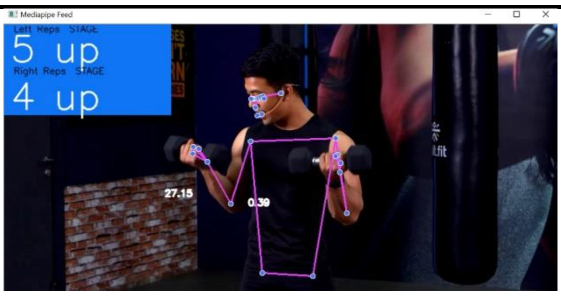


Figure. V. a). Displaying the rep and stage result on the screen for bicep curl.

The video inputs are taken from the webcam or usb microscope and it infers from 33 3D landmarks and background segmentation mask on the whole body from RGB video frames.
The results are displayed in the screen, ensuring the curl count is accordance with the actions.

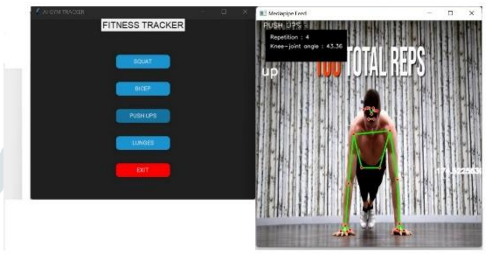


Figure. V. b). Displaying the repetition count for push ups.

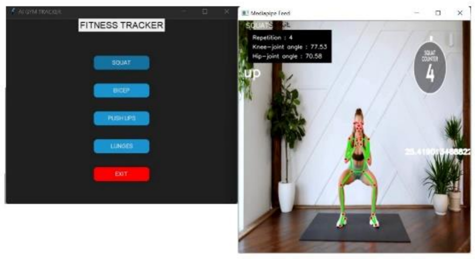


Figure. V. c). Displaying the output hip-joint angle and knee-joint angle.

VI. CONCLUSION AND FUTURE WORKS

In this paper we presented a system for obtaining the pose of the individual at the time of workout and provide the curl count, which makes the workouts effective. We can further enhance this my developing fitness apps that allows for the convenience of one-on-one training without high costs.

Based on human pose, we can build a variety of applications, like fitness or yoga trackers. As an example, we present bicep curl counters, which can automatically count user statistics, or verify the quality of performed exercises. Such use cases can be implemented either using an additional classifier network or even with a simple joint pairwise distance lookup algorithm, which matches the closest pose in normalized pose space.

REFERENCES

[1] 2017 IEEE 12th International Conference on Automatic Face Gesture Recognition Real-time Multiperson 2D Pose Estimation using Part Affinity Fields Zhe Cao Tomas Simon Shih-En Wei Yaser Sheikh, The Robotics Institute, Carnegie Mellon University.

[2] A survey on Model-based approaches for 2D and 3D Visual Human Pose Recovery Xavier Perez-Sala 1, Sergio Escalera 2, Cecilio Angulo3 and Jordi Gonz‘ alez SSN 1424-8220 www.mdpi.com/journal/sensor.

[3] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-tovoxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[4] Shanxin Yuan, Qi Ye, Bj¨orn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 2605–2613. IEEE, 2017.

[5] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 3476–3483. IEEE, 2013