International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 10, October-2019

Match Pose - A System for Comparing Poses

Pradnya Krishnanath Borkar
Computer Department,
FRCRCE

Marilyn Mathew Pulinthitha
Computer Department,
FRCRCE

Mrs. Ashwini Pansare
Computer Department,
FRCRCE

***Abstract*:- The proposed system presents a simple method to compare users' real-time pose with any selected pose. The system uses PoseNet algorithm for real-time pose estimations. The pose comparison algorithm used in system allows a person to compare his/her real-time pose to check if it is successfully imitated.**

INTRODUCTION:

In this busy world, many people wish to go to yoga classes or gym but may not have time or money to do so. The proposed system can help such people to do yoga or dance steps at home without any need to spend a penny. Moreover, proposed system only needs a webcam and is accessible through any browser.

Proposed system allows the user to select an image which he/she wishes to imitate. The real-time poses of the user are then captured through webcam and processed using human pose-estimation algorithm. The selected image from database is also processed using the same algorithm. At the end, the system compares both these poses and prompts the user whether he/she has successfully imitated the pose.

POSE ESTIMATION:

Human pose estimation algorithm allows us to estimate human poses accurately. A detailed overview between some of the human pose estimation algorithms are mentioned below.

Many of the Machine Learning systems as mentioned in [1], [2], [3] needs a 2D human pose estimation as their input. To help the research community OpenPose was released publicly. OpenPose[8] is a multi-person algorithm which successfully detects 135 key-points on a single human image. It consists of three blocks namely a) body and foot detection b) hand detection c) face detection. It can run 22 FPS on a system with Nvidia GTX 1080Ti. A detailed explanation can be obtained in the [9].

PoseNet [10] is another example of human pose estimation algorithm which is widely used. PoseNet is able to detect 17 key-points in a single human image. PoseNet offers single pose algorithm which can detect key-points of one human at a time Or multi-pose algorithm which detects multiple person at a particular point of time. A detailed explanation can be seen in [11].

METHOD OVERVIEW:

The system is a web application which only requires a browser to run it. The system uses PoseNet algorithm to detect human poses. When the user selects a desired image that he wants to imitate, the selected image is processed using single pose algorithm and a set of 17 key points are stored in the system.

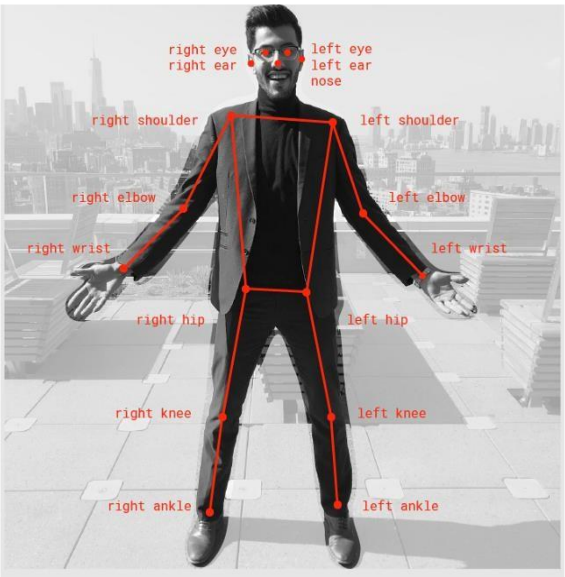


Figure 1: 17 key-points using PoseNet [12]

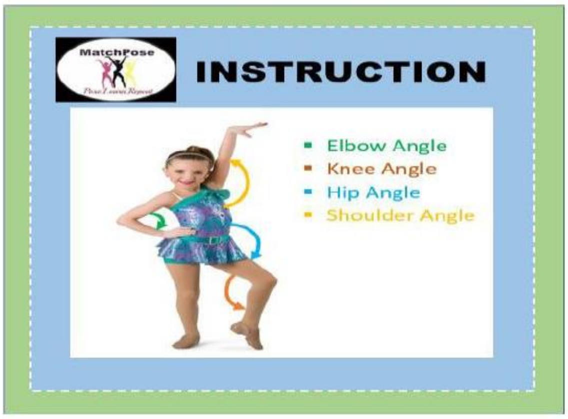
Now the webcam captures the real-time pose of the user. This feed is continuously being processed by the multi pose algorithm of PoseNet. A red colored skeleton formed by joining the 17 key points is displayed. The moment the red skeleton turns green, the user is said to achieve his desired pose.

Under the hood the 17 key points obtained from both the image selected and the webcam is compared using our comparison algorithm. The moment the pose matches each other , the red skeleton turns green.

COMPARISON ALGORITHM:

One of the possible ways you could think of comparing is by superimposing the two skeletons. But you need to transform one of the skeletons such that it overlaps the other. You need to keep in mind that every person has a unique physical structure. Hence length of the arm or legs of each person will differ, hence mapping the skeletons can be a difficult task. Hence the solution to this problem is that instead of mapping the skeletons length, the need is to compare the angles between their joints. The angles are independent of the length of the skeleton hence; they are constant for the maximum time.

Figure 1, refers the angles which are taken into consideration for comparing the pose. As shown in the figure, we consider the elbow and the shoulder angle for comparing the arm position mentioned in Figure 2.



INSTRUCTION

- Elbow Angle
- Knee Angle
- Hip Angle
- Shoulder Angle

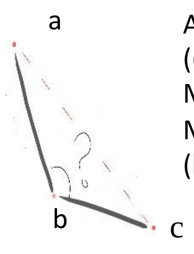
Figure 2: Angles considered in a pose

Along with these angles one more factor is needed.

There is a possibility that these angles are same for both arms which is to be calculated.

Figure 3 illustrates the calculation of the angle.

As shown above to distinguish between both the positions of the arm, the abdominal area becomes the base. The system compares the shoulder co-ordinates and the wrist co-ordinates of an arm to locate if the arm is inward, outward, upward or downward. Similarly, for the leg comparison the system uses Hip angle and Knee angle (Figure 2) and for the position of the leg it compare the hip co-ordinates with the ankle co-ordinates.



```
Angle_B=Math.acos
((Math.pow (len (b, c), 2) +
Math.pow (len (a, b),2) -
Math.pow (len (a, c),2))/ (2*len
(b, c) *len (a, b)))*180/Math.PI
```

Figure 3: Formula for calculating the Angle between joints

For calculating the angles, the system uses the cosine triangle rule as mentioned in Figure 3.

Suppose we are calculating the elbow angle; the co-ordinates of shoulder, elbow and wrist forms an imaginary triangle with the elbow point as the middle angle (refer triangle ABC in figure 3).

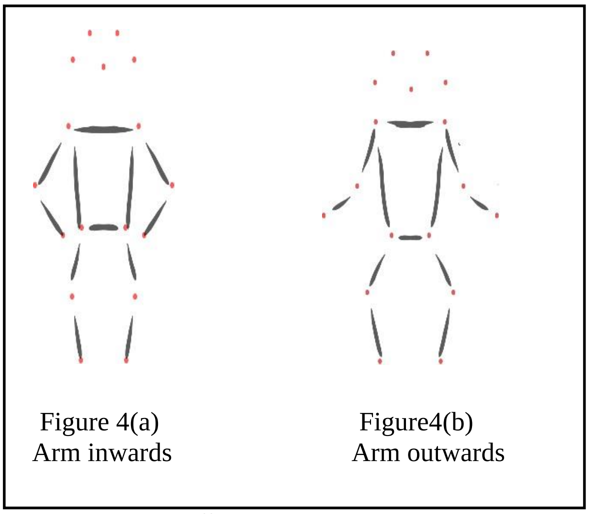


Figure 4(a)
Arm inwards

Figure4(b)
Arm outwards

Figure 4: Different Arm position with equal angles

After obtaining the right elbow angle of both the position in figure 4(a) and 4(b) a situation may be encountered wherein both the angles may be same yet the positions are so different. Hence the extra factor that was discussed earlier is needed, which means , a need to compare the right shoulder co-ordinates of posture in 4(a) and 4(b) with its right wrist position. If the hand position is inwards then the wrist x co- ordinate on a Cartesian plane would be less than the x co-ordinate of the shoulder. Hence, System is able to distinguish between different poses by calculating the angles of the joints and comparing the co-ordinates of the key-points.

CONCLUSION:

Matchpose is successfully able to compare poses. This system can be used for learning new yoga and dance poses. Although this system is working with PoseNet, it can be integrated with another pose estimation algorithm. The main component is the position that is obtained from the estimation algorithm. Currently this system is only comparing the 17 key points mentioned above.

Hence, the dance or yoga poses involving finger positions cannot be compared. That can be included in the system in future as it will follow similar concept as that for arms and legs.

REFERENCES:

[1] X. Qian, Y. Fu, T. Xiang, W. Wang, J.Qiu, Y. Wu, Y.-G. Jiang, and X. Xue,“Pose-normalized image generation for person reidentification,” in ECCV, 2018.

[2] A. Bansal, S. Ma, D. Ramanan, and Y.Sheikh, “Recycle-gan: Unsupervised video retargeting,” in ECCV, 2018.

[3] L. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. Moura, and M. M.Veloso, “Teaching robots to predict human motion,” in IROS, 2018.

[4] P. Panteleris, I. Oikonomidis, and A. Argyros, “Using a single rgb frame for real time 3d hand pose estimation in the wild,” in IEEE WACV. IEEE, 2018, pp. 436–445.

[5] H. Joo, T. Simon, and Y. Sheikh, “Total capture: A 3d deformation model for tracking faces, hands, and bodies,” in CVPR, 2018.

[6] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, “Vnect: Real-time 3d human pose estimation with a single rgb camera,” in ACM TOG, 2017.

[7] G. Hidalgo, Z. Cao, T. Simon, S.-E. Wei, H. Joo, and Y. Sheikh, “OpenPose library,” https://github.com/CMU-Perceptual-Computing-Lab/openpose

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh,” OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”

[10] “Posenet Library,” https://github.com/tensorflow/tfjs-models/tree/master/posenet

[11] Tensorflow Blog, “Real-time Human Pose Estimation in the Browser with TensorFlow.js,” https://medium.com/tensorflow/real-time- human-pose-estimation-in-the-browser-with- tensorflow-js-7dd0bc881cd5

[12] Description of 17 key-points detected by PoseNet, Tensorflow Blog