International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9, Issue-7, May 2020

Corrective Suggestions to Exercise Performed at Home by Physiotherapy Patients using Motion Analysis.

Atul Kumar, Rajesh Kumar Dhanaraj

Abstract: ***Physiotherapy patients pay a lot of money on their sessions where they do the exercises under an expert supervision. But not all can afford a physiotherapy appointment every now and then so few decide to redo the sessions at home by their own. These physiotherapy exercises are pose-sensitive, so in order to get maximum results, the workout pose should be as close to ideal as possible. Otherwise, all the effort put in the session would be for nothing and if done wrongly, it can cause injury also. So, there is the need of a benchmark system that can gauge the correctness of the workout pose performed by the patients remotely. This paper aims to address the issue and provide a system to solve it.***

The patient records the workout video and forwards it to the home workout benchmark system to get corrective suggestions. The system identifies the joint keypoints and the angles between all the involved keypoints. Further, with the application of time series data alignment algorithms, the system identifies the pose errors and generates the report to the patient. This system can be extended in future to the gyms and other health institutions for commercial use..

Keywords : corrective suggestions, pose estimation, pose extraction, workout analysis.

I. INTRODUCTION

With the fast-paced growth and easy reachability of internet, there is a huge library of online workout videos available on websites. Fitness enthusiasts and other people who are suffering from an illness can search online for suitable workout regime. After finding a suitable workout, they can perform the workout at the comfort of their home. But in reality, performing a workout require some sort of supervision and a guidance to get maximum benefit. This supervision can't be obtained just by analyzing on own. A third-party software or guide is required to tell the person that the exercise performed by them is correctly done or not. Since performing a workout incorrectly would only result in wastage of energy and an injury. We propose a workout benchmark system which can provide corrective suggestion to the person performing the workout.

Revised Manuscript Received on May 30, 2020.
***** Correspondence Author
Atul Kumar*, Masters of Technology, School of Computing Science and Technology, Galgotias University
Dr. Rajesh Kumar Dhanaraj , Assosiate Professor, School of Computing Science and Engineering, Galgotias University, India.

The system takes input in the form of video recorded using any video recording device. With that in mind, we propose a workout pose benchmark system in which patient records a video of himself doing the exercise and get the assessment by the system.

II. BACKGROUND

One of the earliest approaches to pose estimation were primarily focused on the graphical models. The fundamental principle is to represent human body as the collection of "parts" organized in a non-rigid configuration. The part can be called as an appearance template which matches the image. Two different parts are spatially connected by using springs. The parts are further placed at their respective pixel position and orientation, such that the system can perform structured prediction task by modelling articulation. But the major limitation of this approach is that the image data is not involved in creating pose model.

In recent years, the CNN model is the most adopted building block for pose estimation. DeepPose [8] applied deep learning technology for the pose estimation by formulating pose estimation as a convolutional neural network-based regression problem towards human body joints. It had 7 layers of AlexNet backend and a final layer which outputs 2k joint locations. The CNN model is trained on FLIC and LSP datasets using L2 loss for regression. It keeps refining the predictions using cascaded regressions. However, the regression to XY co-ordinates is difficult and complex. Hence, it performs poorly in certain regions.

Recent SOTA techniques transform the above problem to estimating heatmaps. Heatmaps perform better than direct join regression as it jointly uses CNN and graphical model. However, heatmap representation lacks structure modelling.

III. PROPOSED MODEL

3.1 Pose Extraction

Pose extraction is one of the primary research area of computer vision. It ranges from human pose to identifying different objects present inside an image. There exist various models to predict the 2D and 3D extraction and estimation of object. OpenPose, DeepCut, AlphaPose, Mask RCNN and many other implementation for human pose estimation exist.

In this paper, OpenPose [9] is used to identify the human body pose correctly.

It produces an output with identified keypoints superimposed on human body present in image, keypoints only in JSON format or as an array class. For a given image, the OpenPose model uses first 10 layers of VGG19 net to generate a vector representation of fixed length. Further, it has two multi-step branches of CNNs.

One branch predicts the confidence maps for human body joints which are represented by dots. The other branch predicts Vector Field maps to establish corelation between the human body joints.

The next stages are used to refine the pose predictions made by each branch. Bipartite graphs are generated using part confidence maps between pair of parts. The weaker links present in the bipartite graph are dropped based on PAF values. By repeating these steps, human pose skeleton is generated for each person present in the image.

A human pose bipartite graph consists of node and links. Node represent the body joints and links represent the joint that are connected to each other.

Fig. 1. Example of a bipartite graph for human pose.

Video, V = { set of Frames }

3.2 CNN

CNN stands for convolutional neural network or simply convNet. It is a deep learning algorithm. This algorithm takes an image as an input. The image can be from a video frame or just a photo clicked using camera. After getting the input, this algorithm assigns a learnable measure and biases to different objects present in the given image. Using this learnable measure and biases, the algorithm can differentiate one image from another image.

An image can be considered as a big matrix of pixel values. The CNN can effectively capture the temporal and spatial correlations present inside an image by using relevant filters. Furthermore, the CNN architecture provides the reusability of weights and the reduced number of parameters involved in processing an image.

The major building block of CNN model are convolution layers. A convolution means applying desired filter to the image input and getting results after its activation. If same filter is applied repeatedly on an input image then the input after activating through a series of layers generates the resultant output which is called as feature map. The feature map indicates the position and strength of the identified feature in the input image.

The filter for the 2-D input image used in convolution is obtained by performing multiplication between array of input image data and 2-D array of weights. Filter are sometimes also called as kernel.

Same filter is applied on multiple patches of the input in a raster scan manner. If the feature is capable of identifying a feature then applying same filter provides an opportunity to detect the same feature anywhere in input image. This is also called as translation invariance.

The weights for the features is obtained by training the convolutional network on specific dataset. For a given input image, the convolutional neural network can apply from 32 filters to 512 filters in parallel and learn from them.

CNN layers can not only applied to raw pixel values but also for output coming from other layers. A hierarchical decomposition of input image is obtained by the stacking of convolution layers.

Low level features such as lines are extracted from raw pixel values if the filer is directly operated on it. Further, these lines are passed through another convolution layer to extract multiple lines that may represent meaningful shapes. This process is continued until the feature is extracted. As the depth of the convNet increases, the abstraction of features improves to higher orders.

Further there are pooling layers which reduces the spatial size of features. These pooling layers decreases the total computational power required and improves the performance of the convolutional model.

3.3 Optical Flow Tracking

A video can contain a large number of frames. That's why optical flow tracking is used to

- compress the video by reducing the total number of irrelevant frames,
- stabilize the video if some frames are missing, and
- generate structure from motion.

It assumes that

- the pixel intensities do not change between two consecutive frames, and
- the neighboring image pixels are having similar pattern of motion.

Lucas-Kanade method is one of the popular methods used for optical flow tracking. It works based on above assumptions and for all the pixels present in the image, it solves the optical flow equations using the least square fit criterion. The method can remove the ambiguity present inherently in the optical equation. Furthermore, there are sparse and dense optical flow models.

Sparse optical flow provides flow vectors of edge pixels within the frame. Whereas dense optical flow provides flow vector of all pixels within the frame. The accuracy of dense optical flow is higher as compare to sparse optical flow but the computation power required to perform dense optical flow is quite high as a result it is slow as compared.

Sparse optical flow is done by selecting a feature like edge or corner from the input image and then tracking velocity vector(motion). The predicted track of feature is validated by checking neighbor frames.

Whereas dense optical flow is done same way like sparse optical flow but it is done for all pixels and its mostly used for video segmentation and obtaining structure from motion.

OpenCV library implements dense and sparse optical flow tracking.

3.4 Time Sequence Data Alignment Algorithm

DTW is one of the popular time sequence data alignment algorithm. These algorithms provide the similarity measure between two time sequences. The two sequences may perform action at different speeds. For example,

- comparison of walking style of person from 2 videos.
- comparing speech pattern of 2 different persons.

DTW stands for Dynamic Time Warping. It follows following rules and restrictions for comparing two videos:

- Every frame from first video must match with one or more frames from the second video or vice-verse.
- The first frame from the video must be mapped to first frame from the other video. (It can have more matches other than this frame)
- The last frame from the first video must match with the last frame from the other video. (It can have more matches than this frame)
- The mapping of frames from the first video to frames from other video must be monotonically increasing or vice-versa.
- Initially the two time sequences are divided into equal points.
- The Euclidean distance is calculated between first point of first sequence and every point in other sequence. The distances are stored for comparison. This stage is also known as time warp.
- Next, it moves to the second point in the first video and then calculates distance with respect to every point on the other time sequence.
- Above steps are repeated until it reaches end point of first time sequence.
- By taking second sequence as reference the above steps are repeated.
- All the minimum distances that were stored adds up and provide a similarity graph between two time sequences.

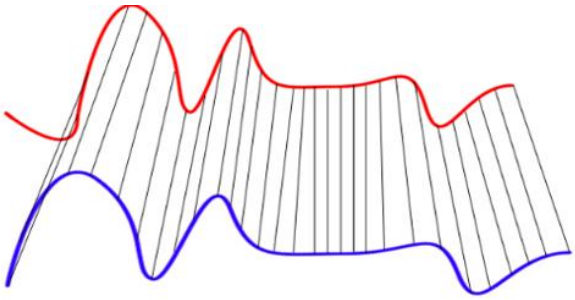

Fig. 2.[12] DTW mapping

If a match satisfies all the above conditions and has minimum cost, then it is said to be optimal match. Dynamic time warping algorithm provides a non-linear alignment between two time sequences i.e., it finds the best possible alignment between two sequences. Hence, it works as a good similarity measure even if the two sequences are out of phase.

3.5 System Overview

Initially, OpenPose [9] is run and it generates the bipartite graph for the human body present in the image frames. The image frames may involve a large number of pose changes. The output generated by OpenPose contains the bipartite graph on top of the given image.

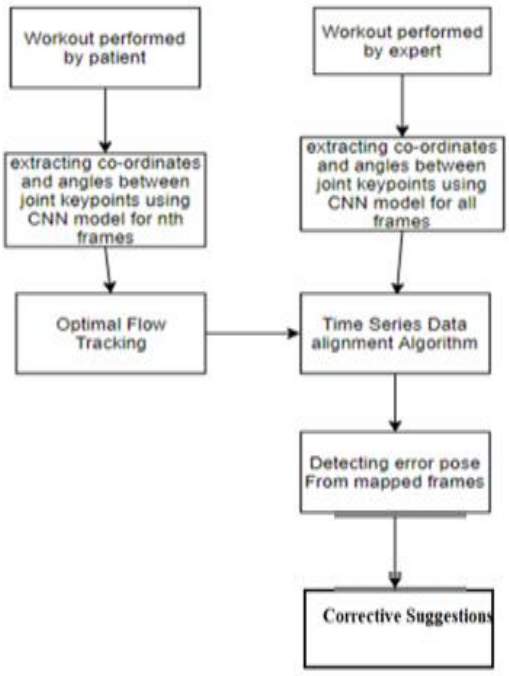

Fig. 3. Workout Benchmark System Flow Chart

The combination of angles between body parts is considered while analyzing the exercise frames. The trainer video is pre-processed first using the OpenPose for each frame in the video. In real-time scenario, it is not advised to perform CNN on each frame of patient's video since it would

Fig. 4. Pose Extraction

cause latency in results. That's why, optical flow tracking is performed in order to provide faster visual feedback. Body part keypoints are extracted from the patient's video at every n-th frame (for example, each 4-th or 8-th frame) and then sparse OpenCV is used to obtain human body keypoints for the intermediate frames. It provides the following advantages:

- Reduce errors in estimation by backward and forward tracking.
- Approximation of keypoints in a missing frame using the information from neighboring frames.

Next, DTW (Dynamic Time Warping) is used for comparing patient and expert's video on the basis of their frame similarity.

Two frame sequences aligned using the DTW algorithm. The similarity between patient and trainer video calculated using DTW as shown in the figure below. It is possible to have more than one frame from first sequence to be mapped to a single frame in second sequence (for example, patient staying in a steady pose).

After getting the mapping of frames, the next step is to perform affine transformations to remove the perception issues (camera orientation, person to camera distance, patient and expert's different body ratios) that might exist there. The expert reference frame is transformed to the patient points (vice-versa). It overlaps the keypoint graph on expert's frame. Furthermore, Least squares problem is solved in order to obtain affine transformation matrix.

The patient uploads an ideal workout video to the workout pose benchmark system. The system then extracts the co-ordinates and angles of joint keypoints using CNN model and stores it. Later, the patient uploads the workout video performed at home to the workout pose benchmark system. This video can be recorded using smartphone or laptop or any other device, keeping the device fixed at an angle. The co-ordinates and angles of joint keypoints from this video are extracted for every n-th frame. Next, optimal flow tracking is used to calculate intermediate frames and time series data alignment algorithm to provide frame-mapping. Now the system can compare the 2 videos using affine transformations. The error is shown in form of visual deviation of joint keypoints extracted from the patient and expert's video.

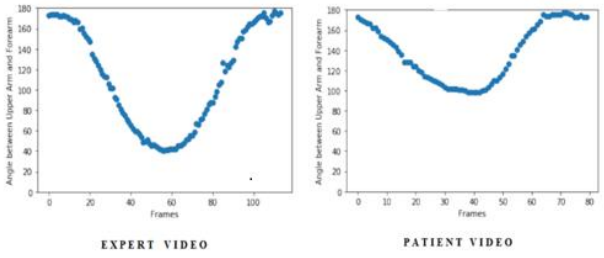

Fig. 5. Expert and Patient plot comparison

The plot shows how the angle between forearm and upper arm of expert and patient is changing frame by frame. The graph can be plotted based on which body part is majorly involved in exercise.

Based on above plot, the system is designed to display message like "Keep the arms a little closer" Or "don't extend the arms too much". In this way, the user can get feedback from the system and adjust the exercise pose accordingly.

IV. RESULTS

Out of all the pose extraction and detection tools, OpenPose[9] is used in our approach because of its ease of installation and open source nature. It is available for both Linux and Windows installation. The patient workout video of 1280x720 resolution is processed at 25fps. The output generated by OpenPose is in the form of JSON format which contains the information about the coordinates of body keypoints and their detection confidence. Depending on the exercise, keypoints to be considered are determined. The efficiency of this process is dependent on the computer system used. Hence, graphic card with at least 2GB RAM is recommended.

A parser extracts the JSON data and provides the plot using the coordinates. Furthurmore, The trainer video provides the minimum and maximum angle benchmark between body part joints. If the patient performs more deviation than a certain threshold , the corrective suggestion is given. Fig. 5 shows an exercise performed by trainer on left and patient on right. The graph is plotted with angle on Y-axis and frames on X-axis. A customized prompt is programmed when the patient exercise pose deviates from the threshold value. The workout is performed on 3 people with different body physique. The system shows that exercise performed by patient is correct with an accuracy of 95% when the angle deviation threshold is kept at 20º. For an ideal benchmark system, the maximum angle deviation allowed would be set to 0º which is not possible humanly. Qualitatively, the people involved in this process were quite satisfied with the results.

V. CONCLUSION

The workout benchmark system is capable of providing corrective suggestions to the patient. The corrective suggestions are provided based on the angle deviations found between expert and patient keypoints in the frames. Laptop and webcam set at 1280x720 resolution were used for recording patient's exercise. The exercise videos were recorded at 25 fps frame rate. Furthermore, The corrective suggestions are shown in the form of text. This can help the patients to correct themselves and gain maximum out of their home workout sessions.

The CNN model used above for human pose estimation was trained on COCO dataset. Currently the system can provide corrective suggestions in text form on display screen but in future it can be extended to an AI 3D avatar based app to provide corrective suggestions using speech.

REFERENCES

1. A. Newell., K. Yang, J. Deng, "Stacked Hourglass Networks for Human Pose Estimation.", Computer Vision – ECCV. Lecture Notes in Computer Science, Springer Cham, 2016, vol. 9912.
2. A. Jain, J. Tompson, Y. LeCun, C. Bregler, "MoDeep: A Deep Learning Framework Using Motion Features for Human Pose Estimation.", Computer Vision – ACCV. Lecture Notes in Computer Science, Springer Cham, 2014, vol. 9004.
3. Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, Richard Moore, "Real-time human pose recognition in parts from single depth images.", Communication of the ACM, vol. 56, 2013, no. 1.
4. A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, L. Fei-Fei, "Towards Viewpoint Invariant 3D Human Pose Estimation.", Computer Vision – ECCV. Lecture Notes in Computer Science, Springer Cham, 2016, vol. 9905.
5. Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh, "Convolutional Pose Machines", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4724-4732
6. Gabor szues and Bence Tamas, "Body part extraction and pose estimation method in rowing videos.", Journal of computing and information technology, vol. 26, 2018, no. 1
7. Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, Christoph Bregler, "Efficient Object Localization Using Convulational Networks.", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 648-656

8. Alexander Toshev, Christian Szegedy, “DeepPose: Human pose estimation via deep neural networks.”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1653-1660
9. Z cao, G Hidalgo, T Simon, SE Wei, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields.”, Computer Vision and pattern Recognition, 2018, arXiv:1812.08008.
10. Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, Jitendra Malik, “Human Pose Estimation With Iterative Error Feedback.”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4733-4742.
11. Bin Xiao, Haiping Wu, Yichen Wei, “Simple Baseliners for Human Pose Estimation and Tracking.”, The European Conference on Computer Vision (ECCV), 2018, pp. 466-481.
12. “DTWmapping”, https://commons.wikimedia.org/wiki/File:Euclidean_vs_DTW.jpg

AUTHORS PROFILE

Atul Kumar is currently pursuing Masters of Technology from School of Computing Science and Technology, Galgotias University. He has received B. Tech Computer Science degree from Ambedkar Institute of Advanced Technology & Research, New Delhi, India.

Dr. Rajesh Kumar Dhanaraj is currently working as Assosiate Professor at School of Computing Science and Engineering, Galgotias University, India. He has done his Ph.D in Information & Communication Engineering from Anna University, Chennai, India.