Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

J Component report

Programme	**: B.Tech (CSE AI and ML)**
Course Title	**: VIDEO ANALYTICS**
Course Code	**: CSE3043**
Slot	**: B2**

Realtime Physiotherapy and Fitness Assistant

Aaditya Hemant	**| 19BAI1098**
Ritika Sarkar	**| 19BAI1031**
Subramanian Venkittanarayanan	**| 19BAI1089**

Faculty: Dr. G. Malathi **Sign:**

Date:

ABSTRACT

The pandemic has incentivized innovation that utilizes remote technology to achieve communication between people. Hospitals are often full, and as result, online consultations have seen a rise in preference. However, due to the limited availability of professionals, there is a difficulty in providing at-home services for patients that require physiotherapy. These individuals may be recovering from an operation or require such exercises to maintain their muscular and physical health.

We aim to create an automated system for tracking the physiotherapy poses of an individual which are then juxtaposed with the correct postures of the corresponding exercises. The system will offer suggestions on how to improve on the exercises as well as score them. We propose enhanced error identification by monitoring the body parts of the person and will perform these tasks on a live video feed in real-time. Thus our system will reduce the workload of professionals and provide easier solutions for patient-assisted physiotherapy and exercise.

1. Introduction

Physiotherapy is a form of medical treatment which helps restore the normal functioning of any part of the body if the person is injured due to some accident or traumatic incident or has some acquired disability. People of all age groups can receive physiotherapy to alleviate pain in their bodies and facilitate a speedy recovery. Various conditions like back pain, arthritis, patients with heart diseases, asthma, sports injuries can be treated using this method. Millions of people are affected by different kinds of incidents every day however most of them either don't have access to any medical facilities offering physiotherapy treatment nearby or are affected by the pandemic going on for more than two years now. Such a situation compels us to think of an innovative remote solution where the patients can self-learn the physiotherapy exercises and perform them without the physical presence of a trainer or expert. Some existing systems that most patients have resorted to these days consist of an android application playing pre-recorded videos of the trainers performing the exercises which are recommended by the patients' doctors. However, this is inefficient, as they hardly have a way of knowing whether they are able to follow along properly without anyone correcting them.

A more intelligent and automated system is required for overcoming these challenges. We propose an application for real-time physiotherapy exercise monitoring, through which the patients can follow along with the instructor and also get feedback about their performance in real-time using a scoring system and whether the correct body parts are being used. We use the popular OpenPose framework for the Pose Estimation and construct an interface for users to monitor their real-time video feed getting captured and compared against the expert's video so that they are able to know how to correct themselves. The paper first presents literature on existing works in this domain, then introduces the proposed model, and finally the implementation and results.

2. Literature Survey

Hassan et al. [1] utilize the PoseNet model to assess a video input of a person performing an exercise and analyze it to provide corrective feedback. The Home Based Physiotherapy Exercise (HPTE) dataset consists of 240 exercises, with 480 videos available in both RGB and grayscale. L2 normalization is applied on the key points estimated by PoseNet, to accommodate users of all shapes and sizes. In order to perform the comparison, two methods are implemented. In the first, a pickled version of keypoints in the HPTE dataset was created and used. This method is more accurate but slower. The second methodology involves Dynamic Time Warping (DTW) which is used to measure nonlinear similarity between two-time series. This technique is faster but slightly less accurate.

Human gestures in RGB videos are classified using a deep learning pipeline consisting of a real-time human pose extraction and processing system called OpenPose, and time series classification using Dynamic Time Warping and One-Nearest Neighbor by Schneider et al. [2]. The approach is flexible and hardware-independent, and new gestures can be taught to the classifier using only a small amount of training data. As compared to the existing literature, they did not achieve higher accuracy. However, the ability of their model to work on less training data is a noteworthy result.

Godse et al. [3] propose an end-to-end computer vision-based model for rating postures. Each frame of video is resized to a fixed form factor and converted to a minimum data loss format. Open Source posture detection model OpenPose is used to detect various joints and important points in the body and estimate measurement between them. A normalized error value is created based on the difference in distance of these points against previously fed values. The person's heights, weight, and size are included in the calculations to improve performance and efficiency.

Saree et al. [6] have developed a remote platform for therapists to monitor the exercises performed by their patients. The platform, named ExerciseCheck enables the patient to receive supervision and instant feedback from their therapists. The platform is used with the Kinect interface and the authors have demonstrated through the results of testing it with the arm raise and the squat exercises the potential benefits this system possesses, enabling two-way remote communication between the patients and their therapists.

Rao et al. [7] propose a sensor-based framework to monitor the progress of physiotherapy exercises. They use the features of the skeleton generated using time-frequency through Discrete Wavelet Transforms and feed into an RNN and LSTM architecture to identify the exercise being performed. They also study whether muscle fatigue is caused by using a reference exercise generated from the best instance out of the segmented exercises. The progress of the exercise is generated throughout the course of the exercise and evaluated at the end using certain metrics.

3. Requirements Specification

3.1 Hardware Requirements

The following specifications are only representative of the system used for the development of this project:

Processor - Intel Core i5 9th generation above or AMD Ryzen 5 3000U/H and above, or a CPU with at least 8 cores

Installed RAM - 8 GB minimum, 16 GB recommended

GPU - Nvidia GPU with minimum of 1.5 GB available memory, 4 GB memory recommended

3.2 Software Requirements

The software and packages used were as follows:

i. *Google Colab* - It is a platform where IPython notebooks are hosted in the Google cloud and have access to GPU and TPU resources by Google. We use this platform to collaborate in real-time and implement the project.

ii. *Python* - This is the main programming language used throughout the project.

iii. *OpenCV* - The video processing tasks and extraction of keyframes have been done using this Computer Vision library.

iv. *OpenPose* - Openpose is a real-time human pose detection library that is capable of identifying various key points in the human body. Each body part is associated with its own keypoint which is used to identify and track parts from the reference video and the data extracted from the real-time video feed.

v. *Tkinter* - Tkinter is a python GUI library with vast documentation and various features. We make use of Tkinter to create an application that can show the videos in the directory and begin OpenPose inference on two videos plus show them in the same video for comparison and user guidance.

vi. *Operating System* - Ubuntu 18.04 or Ubuntu 20.04+ required. Currently does not support Windows, in its application format. The standalone code to perform video comparison can be run on Windows 10 if needed.

4. System Design

The basic hardware requirements include a laptop with a camera, and a CPU powerful enough to perform pose estimation and real-time video processing. The models will be created and trained in the Python programming language. OpenPose is a pre-existing real-time pose estimation library complete with limb identification and tracking. This will be used to track the user's body movements. A reference video containing the correct method of performing the exercise will be required. This will allow us to perform a similarity comparison between the user and the professional when they perform the exercise. Evaluation will be done based on the similarity between the videos, ensuring that certain differences such as length differences between the limbs and speed of motion are accounted for. Corrective feedback will be provided in real-time as well to allow users to adjust accordingly.

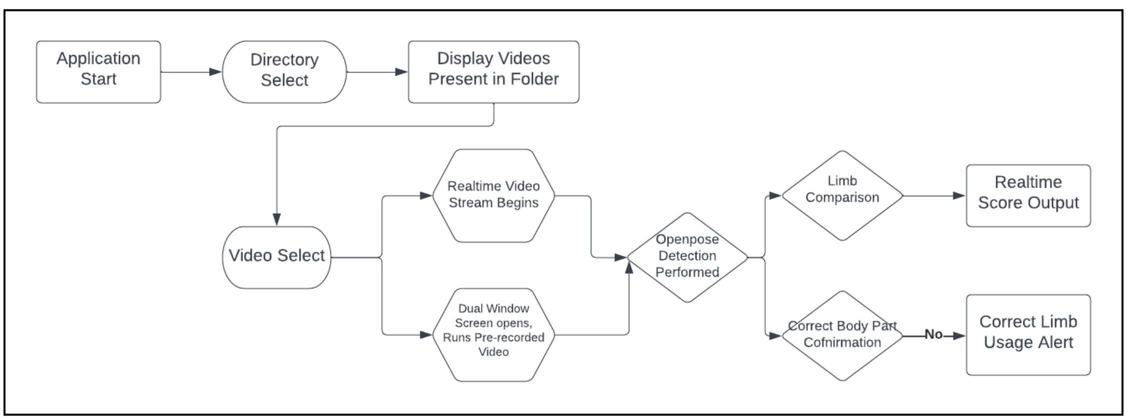

Figure: Architecture Diagram of the Physiotherapy Assistant

5. Implementation of System

5.1 Dataset Generation

Dataset for the physiotherapy exercises is taken from the Home-Based Physical Therapy Exercises data set which has various exercises and physiotherapy movements. The dataset contains 240 exercises recorded in a standard video .avi format. We utilize this dataset to collect reference videos for multiple preset configurations. Following this, we use our Openpose inference to extract pose information and motion ranges for each exercise and store them in a database for comparison.

5.2 **Methodology**

Our proposed method starts by obtaining the video feed live from the user as well as the pre-recorded reference video of the expert that contains the correct performance of the exercise. Both the videos are then compared side by side in real-time using Fast Dynamic Time Warping with the cosine similarity distance metric to measure and quantify how much the videos are differing from each other and return the score. The user performs the exercise as done in the video and the model also tracks their limbs to evaluate the technique, as to whether the correct limb or body part is being used.
Based on the similarity comparison, the model evaluates the exercise done by the user. Corrective feedback is provided, and the user may continue to perform the exercise as many times as required.

The various modules of the project are as follows:

5.2.1 Real-time Video Processing

Ensuring that the user can use our application in real-time to help better understand the correct technique to perform the exercise. Takes in video input from a camera source such as the laptop camera, and outputs the processed frames as a video stream onto the display. OpenCV was used to read the video feed and process it as frames for real-time processing. Two subprocesses are used to simultaneously process the user feed and the reference video.

5.2.2 Pose Estimation

From the processed video we make use of pose estimation libraries such as OpenPose to get the positions of different points across the body. Take in individual video frames and return an associated JSON object containing the coordinates of each of the points.

5.2.3 Limb Tracking

This module will help identify the individual body parts of the user. It is necessary to monitor them as it is possible that the user may be performing the exercise using an incorrect body part. That may deter the very purpose of the exercise and increase complications. It is done using the knowledge of the keypoints values returned by OpenPose and their position in the human anatomy. After getting the keypoints we process the frames to identify the points with the most displacement and ensure that these are the same points being utilized by the user.

5.2.4 Pose Evaluation and Scoring

It is done based on the video feed containing the correct performance of the exercise. Users will be encouraged to follow along with the sample video, to ensure the exercise is performed with the appropriate technique and speed. Fast DTW is used for calculating the measure of similarity between the real-time user video and the reference video using cosine-similarity as the cost function for comparing the keypoints. The score shows how well the user is doing the exercise, where a high score indicates correctness and vice versa. The presence of the correct body part is also taken into consideration in the score calculation.

5.2.5 Graphical User Interface

In order to make a user-friendly application that is both functional and easy to use. GUI was developed using Tkinter. The interface consists of a start screen that loads the previously selected directory. The user has the option to change the directory, which will reload the window and update the buttons present to contain thumbnails of the videos present in the chosen folder. When the user clicks on one of the video buttons listed, the OpenPose code is executed, and a window with both a real-time camera stream and the prerecorded video will be displayed. The GUI elements were primarily developed using the Tkinter library, and provide responsive UI features that can be coded in Python.

6. Results and Discussion

The application was successfully built-in Python using the Tkinter library, providing the user with an interface that is easy to use, functional and responsive. The user is able to select a video of their choice to use as a reference to perform the exercise. The user can also select an alternative directory, and the application successfully reloads to display the correct videos and their thumbnails.

Image 1: Physiotherapy Assistant App Home Screen

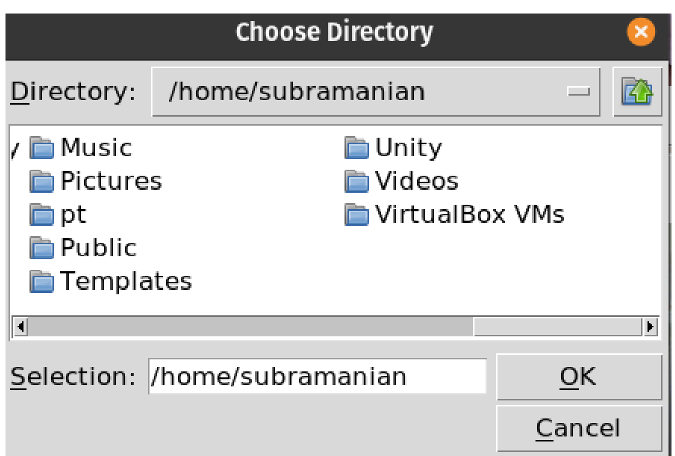

Image 2: Directory Selection Option

Image 3: Successful Reload After New Directory Selected

Selecting a video loads the OpenPose comparison window frame. The user can then visualize the performance of the exercise, through the limb identification overlay. The user then follows along with the exercise. If the correct body part is not being used, the alert is displayed indicating the user's mistake. The correctness to which the user performs the exercise is also displayed as a percentage. The percentage score is continually updated every frame, ensuring that the user has to perform the exercise correctly throughout the video, in order to achieve a high score.

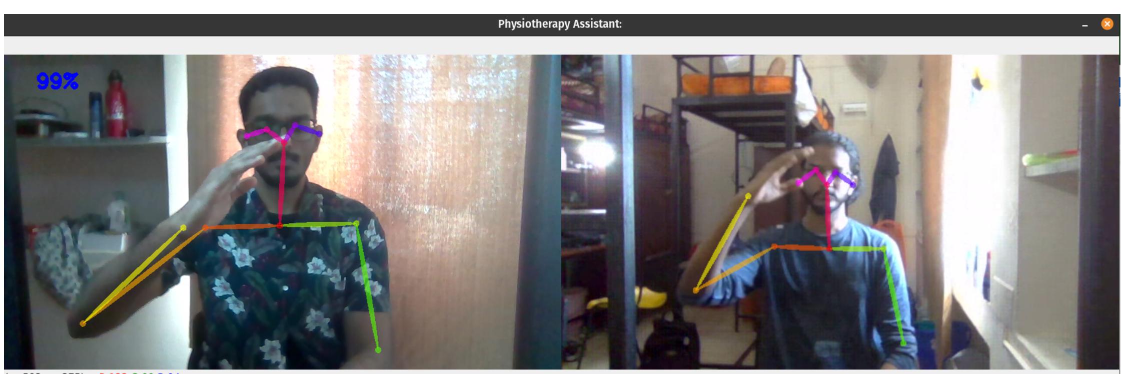

Image 4: Openpose Model Pose Comparison- High Similarity

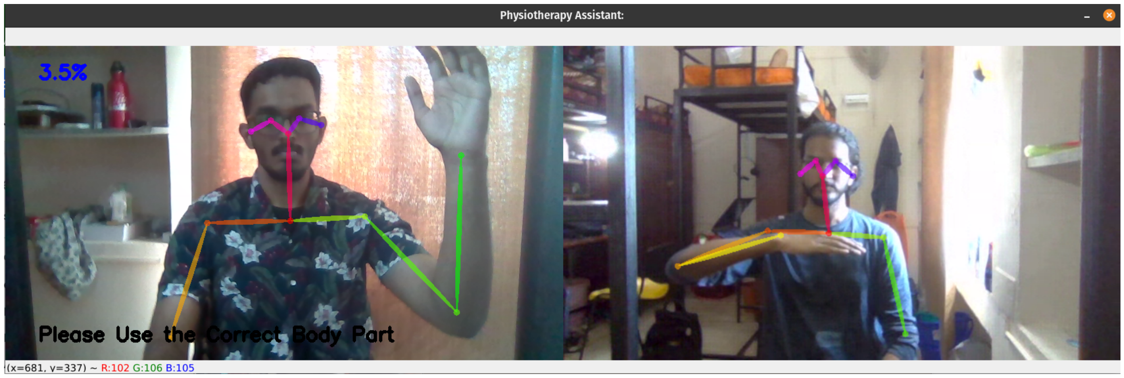

Image 5: Openpose Model Pose Comparison - Wrong Body Part

In some cases, however, when the motion is simple and restricted to one or two limbs, the score output may not be completely indicative of how well the user is performing the exercise. This score is a result of the other limbs detected, aligning with the limbs of the individual present in the reference video. This can be fine-tuned by identifying the limbs that partake in the movement, via optical or flow or frame difference, and only using them to calculate the similarity score.

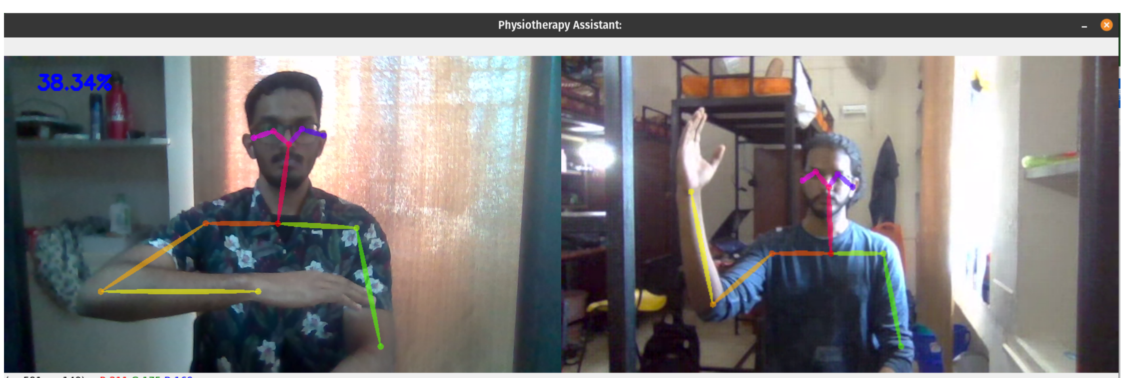

Image 6: Openpose Model Similarity Comparison - Low Similarity

A performance summary graph of the user's scores throughout the exercise is also generated, by taking the scores at every frame and plotting it. This provides a visualization of how well the user performed the exercise, and at which parts they struggled. Repeatedly performing the action using the software allows the user to improve their form and eventually complete the exercise without requiring the reference footage.

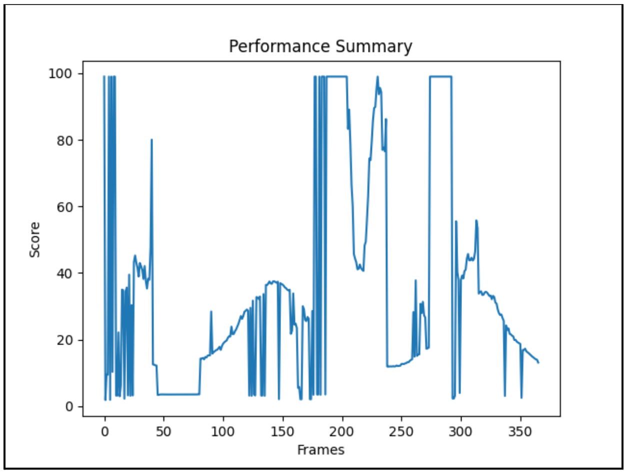

Image 7: Performance Summary Graph Representing Correctness of Exercise Action

7. Conclusion and Future Work

Our project aims to improve the process of postoperative recovery and general physiotherapy by ensuring that patients everywhere have an easy, simple, and accessible way of performing self-evaluation. With the pandemic still affecting our lives, and changing the future as well, it is important to innovate with the idea of remote/at-home services in mind. We utilized OpenPose library to extract the key points of the patient and the reference and process this data in a comparative manner to identify the appropriate limbs being used for the exercise using cosine-based comparison we determine whether the user is performing the exercise correctly. Both the reference video and the user live feed are synced and displayed side by side in the GUI where the user can easily self evaluate. We also used Tkinter and Python to generate a graphical user interface that makes navigating our application simple and straightforward.

8. REFERENCES

[1] Hassan, H.A.; Abdallah, B.H.; Abdallah, A.A.; Abdel-Aal, R.O.; Numan, R.R.; Darwish, A.K.; El-Behaidy, W.H. Automatic Feedback For Physiotherapy Exercises Based On PoseNet. FCAI-Inform. Bull. 2020, 2, 10–14.

[2] Schneider, Pascal, Raphael Memmesheimer, Ivanna Kramer, and Dietrich Paulus. "Gesture recognition in RGB videos using human body keypoints and dynamic time warping." In *Robot World Cup*, pp. 281-293. Springer, Cham, 2019.

[3] Godse, Sachin P., Shalini Singh, Sonal Khule, Vedant Yadav, and Shubham Wakhare. "Musculoskeletal physiotherapy using artificial intelligence and machine learning." *Int. J. Innov. Sci. Res. Technol.* 4, no. 11 (2019): 592-598.

[4] Ar, I., & Akgul, Y. S. (2013). A monitoring system for home-based physiotherapy exercises. In *Computer and Information Sciences III* (pp. 487-494). Springer, London.

[5] Vaish, A., Ahmed, S., & Shetty, A. (2017). Remote physiotherapy monitoring using the novel D+ R Therapy iPhone application. *Journal of Clinical Orthopaedics and Trauma*, *8*(1), 21-24.

[6] Saraee, E., Singh, S., Hendron, K., Zheng, M., Joshi, A., Ellis, T., & Betke, M. (2017, June). ExerciseCheck: remote monitoring and evaluation platform for home based physical therapy. In *Proceedings of the 10th international conference on PErvasive technologies related to assistive environments* (pp. 87-90).

[7] Rao, N., Surana, P. M., Ragesh, R., & Srinivasa, G. (2019, September). Analysis of Joints for Tracking Fitness and Monitoring Progress in Physiotherapy. In *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)* (pp. 181-185). IEEE.