Dissertation Submitted for the partial fulfillment of the **M.Sc. (Integrated) Five Years Program AIML** degree to the Department of AIML & Data Science.

M.Sc. Project Dissertation

YOGA POSE DETECTION AND POSE CLASSIFICATION

submitted to

By

ZEEL RATHI
Semester- VII

M.Sc. (Integrated) Five Years Program AIML

Department of AIML & Data Science
School of Emerging Science and Technology
Gujarat University

20 December, 2022

DECLARATION

This is to certify that the research work reported in this dissertation entitled "**Yoga Pose Detection and Pose Classification**" for the partial fulfilment of Master of Science in Artificial Intelligence and Machine Learning degree is the result of investigation done by myself.

Place: Ahmedabad

Date: 20-12-2022

Name of Student: Zeel Rathi

Index

Chapter 1
Abstract & Key Words

Yoga is a new way to look at the disciplines of our lives, such as physical, mental, and spiritual practices. Yoga started many years ago. Because of the many benefits of yoga, medical professionals and many celebrities suggest doing yoga for a healthy lifestyle. Taking care of your body, mind, and breath is a simple definition of yoga. However, depending on the COVID-19 situation, everything happening from home in this situation it is important to maintain your health. You need to do exercise daily. To do the exercise properly an instructor is needed at home that you cannot do in the COVID-19 situation. It is injurious to our health to do yoga poses without an instructor. So here we are going to represent a proposed model for yoga pose detection using a machine-learning algorithm to identify and detect the yoga pose form. Our system works on 3 yoga poses.

Keywords: Media Pipe, OpenCV, Machine Learning

Chapter 2

Introduction

Human pose recognition is an extremely troublesome and difficult task within the discipline of computer vision. It deals with the localization of human joints in a picture or video to make a skeletal illustration. To mechanically discover a user's activity in a picture may be a troublesome drawback because it depends on a variety of aspects like scale and determination of the image, illumination variation, background muddle, venture variations, and interaction of humans with the environment. The matter with yoga is that, rather like the other exercise, it's of utmost importance to apply it properly as any incorrect posture throughout a yoga session is often unproductive and probably damaging. This results in the requirement of getting an educator to supervise the session and proper posture. Since not all users have access or resources to an educator, an artificial intelligence-based application can be wont to determine yoga poses and supply customized feedback to assist people to improve their poses. In recent years, human pose estimation has benefited greatly from machine learning and large gains in performance are achieved. Machine learning approaches give an additional simple approach of mapping the structure rather than having to wear down the dependencies between structures manually. This project focuses on exploring the various approaches for yoga pose classification and seeks to realize insight into the following: what's cause estimation? What's Machine learning? However, will machine learning be applied to yoga pose detection in real-time? This project uses references from conference proceedings, revealed papers, technical reports, and journals. The second section talks about cause estimation and explains different kinds of cause estimation ways thoroughly and goes one level deeper to elucidate descriptive ways – learning primarily based (machine learning) and model. Totally different cause extraction ways are then mentioned in conjunction with machine learning primarily based models - logistical regression and python library like MediaPipe, OpenCV for cause estimation.

Chapter 3

Basic Terminology

What is OpenCV?

➔Read, Display and Write an Image using OpenCV

For that we have 3 basic operations:

1. `imread()` helps us read an image
2. `imshow()` displays an image in a window
3. `imwrite()` writes an image into the file directory

➔Reading and Writing Videos using OpenCV

These are the main functions in OpenCV video I/O

1. `cv2.VideoCapture` – Creates a video capture object, which would help stream or display the video.
2. `cv2.imshow()`, `cv2.waitKey()` and the `get()` method which is used to read the video metadata such as frame height, width, fps etc.

More Libraries Used

1. MATH
2. NUMPY
3. TIME
4. MATPLOTLIB

What is MediaPipe?

5. MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc.

6. It supports multimodal graphs.

7. To speed up the processing, different calculators run in separate threads.

8. Working with time series data must be in proper synchronization; otherwise, the system will break.

9. The graph ensures this so that flow is handled correctly according to the timestamps of packets.

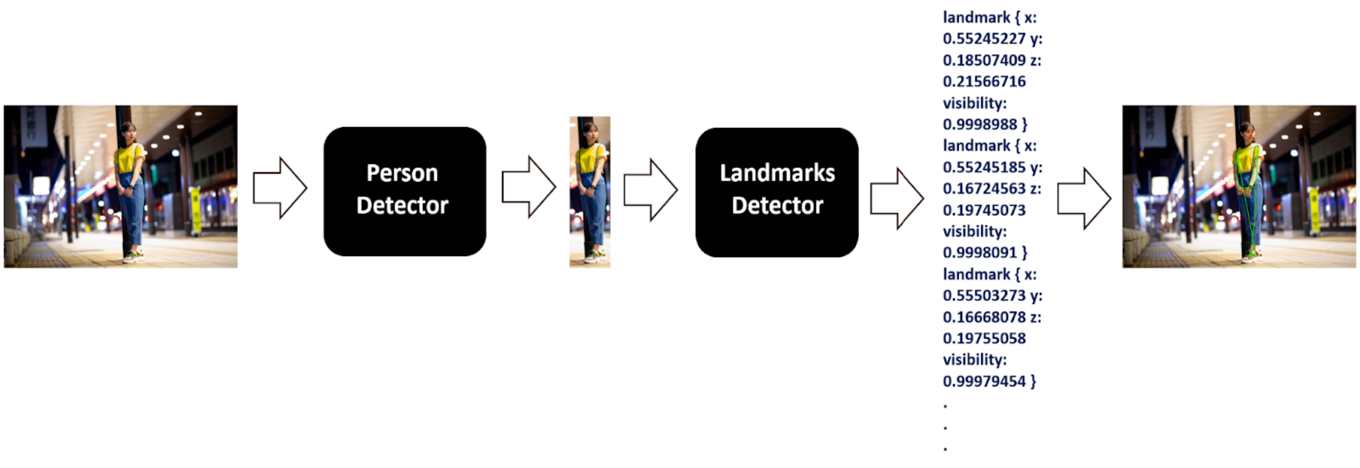
10. It is computationally inexpensive than PoseNet, OpenPose in pose detection.

Overall, it is a beautiful, fast-growing library that delivers promising results. Implementing MediaPipe in projects nullifies most of the hassles we usually face while working on an ML project.

Mediapipe employs a special detection scheme called “2 STEP DETECCTOR”.

2 STEP DETECTOR : Detection + Tracking

- ➔ Detection normally searches the object in the whole image.
- ➔ Tracking takes the image from the previous detected frame.
- ➔ The object detection is computationally expensive.
- ➔ The tracking approach is relatively inexpensive because we already have a previous bounding box.



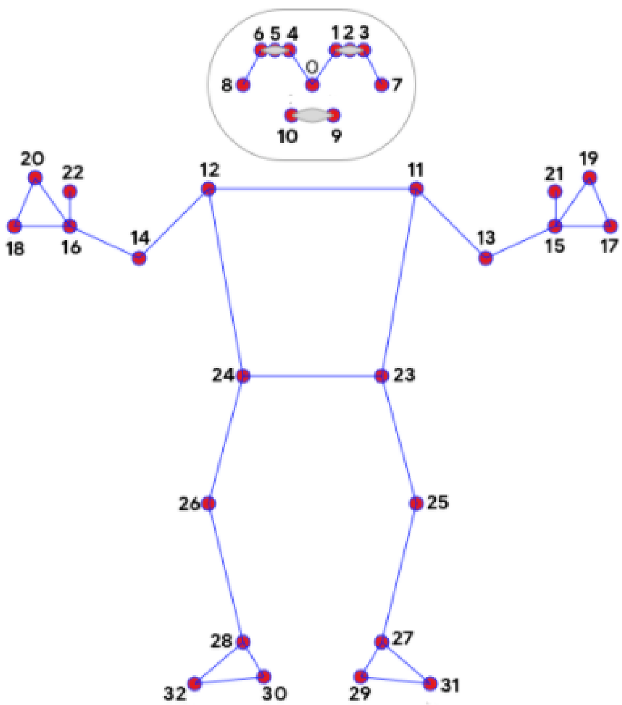
Here’s how this works:

You run the detector in the first frame of the video to localize the person and provide a bounding box around it, after that the tracker takes over and it predicts the landmark points inside that bounding box ROI, the tracker continues to run on any subsequent frames in the video using the previous frame’s ROI and only calls the detection model again when it fails to track the person with high confidence.

This model best works if the person is standing 2-4 m away from the camera

Limitation – Not applicable for multi-person detection.

Here are the 33 landmarks that this model detects:



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Terminologies Initialized in the Pose Detection Model

The first thing that we need to do is initialize the pose class using the mp.solutions.pose syntax and then we will call the setup function mp.solutions.pose.Pose() with the arguments:

- static_image_mode - It is a boolean value that is if set to False, the detector is only invoked as needed, that is in the very first frame or when the tracker loses track. If set to True, the person detector is invoked on every input image. So you should probably set this value to True when working with a bunch of unrelated images not videos. Its default value is False.
- min_detection_confidence - It is the minimum detection confidence with range (0.0 , 1.0) required to consider the person-detection model's prediction correct. Its default value is 0.5. This means if the detector has a prediction confidence of greater or equal to 50% then it will be considered as a positive detection.

- `min_tracking_confidence` - It is the minimum tracking confidence `([0.0, 1.0])` required to consider the landmark-tracking model's tracked pose landmarks valid. If the confidence is less than the set value then the detector is invoked again in the next frame/image, so increasing its value increases the robustness, but also increases the latency. Its default value is `0.5`.

- `model_complexity` - It is the complexity of the pose landmark model. As there are three different models to choose from so the possible values are `0`, `1`, or `2`. The higher the value, the more accurate the results are, but at the expense of higher latency. Its default value is `1`.

- `smooth_landmarks` - It is a boolean value that is if set to `True`, pose landmarks across different frames are filtered to reduce noise. But only works when `static_image_mode` is also set to `False`. Its default value is `True`.

Then we will also initialize mp.solutions.drawing_utils class that will allow us to visualize the landmarks after detection, instead of using this, you can also use OpenCV to visualize the landmarks.

Chapter 4

Literature Review

There have been multiple methods that have been used to perform human pose estimation. For our case we are going to focus on single person pose estimation.

- ➔ The traditional approach involves performing calculation over different combinations of local parameters of body parts and the relative dependencies between them. Two major methods are used, the tree structured model and non - tree based models. The tree-based model uses parametric encoding to find the spatial relation between related body parts using kinematic chain links. While the non-tree models use edge capture method, occlusion, symmetry and link relationship to obtain reliable observation related to body parts.
- ➔ Yuling Hsu et al in 2018 developed a inertial sensor net- work that worked on activity recognition algorithm that could accurately predict the daily human activity and sports activities.
- ➔ In 2018 Chen HT et al proposed a system of tracking human pose using Kinetic console sensors in "Computer-assisted yoga training system". In this method they have suggested the used of feature based approach for designing a yoga training system.
- ➔ In 2017 Ya Lui et al proposed using a interacting multiple model (IMM) based on unscented Kalman filters
- ➔ Other methods include contour-based pose estimation techniques. In "Yoga tutor: visualization and analysis using SURF algorithm" a SURF algorithm approach is taken to identify the pose of users in images. SURF (Speeded Up Robust Feature) algorithm is an advanced version of the SIFT algorithm. The approach includes Gray scaling i.e. converting the image into Gray scale, thresholding to eliminate the background and separate it from the user. Erosion and dilation to increase the smoothing the boundary noise of the image. Further the Canny algorithm is applied in the prefinal stage which does smoothing, gradient and edge tracking. And finally, the SURF algorithm is applied to compare the reference video with the practitioner video.
- ➔ Another method of pose estimation is through neutral network and deep learning. In "Human Pose Estimation Using Convolutional Neural Networks" the authors have created a simple model using convolutional neural network that estimates the poses and demonstrates the potential of CNN’s. The model predicts a set of locations and a set of confidence maps with body part location and degree of association between parts in the form of 2D vector fields
- ➔ In 2014 Alexander Toshev and Christian Szegedy presented the paper "DeepPose: Human Pose Estimation via Deep Neural Networks" in which they proposed a method for pose detection using Deep Neural Network (DNN). They considered pose identification as a joint regression problem and depicted how to successfully cast it in DNN model.

Chapter 5

Methodology

Here, a machine learning based yoga pose estimation methodology is proposed to detect correct yoga poses and provide feedback to improve the yoga posture. It consists of three main steps:

(1) Feature extraction: videos or images are given as input to the model, and frames are extracted at regular intervals from videos and sent to MediaPipe's function to extract key points. From these key points, 33 joint vectors are calculated. For all these 33 joints, angles between the x-axis and joints are found, respectively.

(2) Classification: these angles are sent to the classification model to classify the pose among 3 yoga poses. These angles are compared with an array of 33 angles of the classified pose. This array contains average angles of 33 joints from the dataset.

(3) Feedback generation: the differences are calculated, respectively, for every angle, and suggestions are revealed. The proposed approach is represented schematically in Figure 1, and further explanations of each step are provided in the following sections

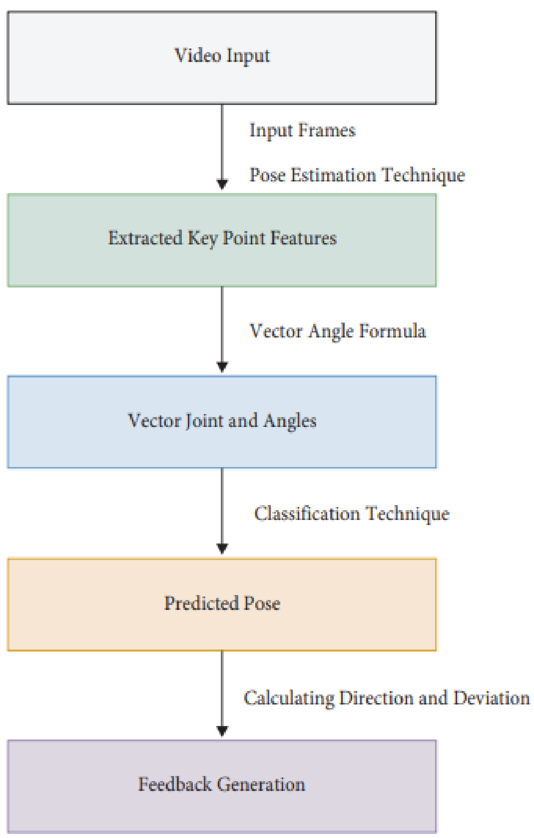


FIGURE 1: The figure illustrates the overview of the proposed method.

Chapter 6

Data Analysis

Our approach aims to automatically recognize the user's Yoga asanas from real time videos. The method can be written into four main steps. First, data collection is performed which is a real-time process running in parallel with detection. Second, Media pipe is used to identify the joint locations using BlazPose concept to detect the joints location which are visible.

➔ The dataset consists of jpg images downloaded from Kaggle.
➔ There are images of people posing in 3 different poses.

1. Warrior II Pose
2. Tree Pose
3. T Pose

DATA PREPROCESSING –

Step 1) Converting the image into RGB Format.

Step 2) Iterate through all landmarks and printing the first 2 landmarks which are going to be detected.

Step 3) After getting the extracted normalized landmarks values of x, y, z coordinates into their original scale by using the width and height of the image.

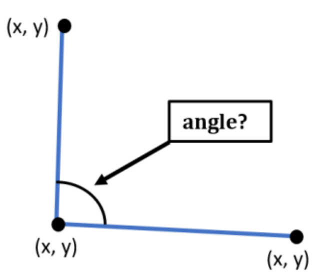
Step 4) Calculated the FPS for real time detection on webcam.

Pose Classification with Angle Heuristics

Now we will create a function that will be capable of classifying different yoga poses using the calculated angles of various joints. The function will be capable of identifying the following yoga poses:

- Warrior II Pose
- T Pose
- Tree Pose

➔ It is the simple function which takes three landmark points making an angle.
➔ Multiple no of angles are extracted.
➔ Detect what pose a person is making.



Chapter 7

Results

Our system successfully detects the yoga pose and make predictions accordingly. I have worked on total 3 asanas which are Warrior II pose, Tree pose, T pose.

Our model gives 100% accuracy so from that we say our system does an excellent job.

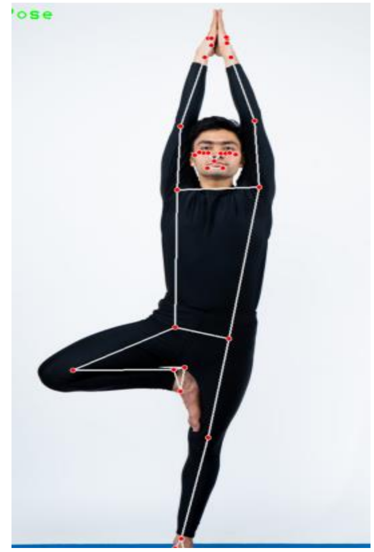
It detects the name of asana. This helps the user to do yoga asanas correctly.

Output Image: The image with the detected pose landmarks drawn and pose label written.

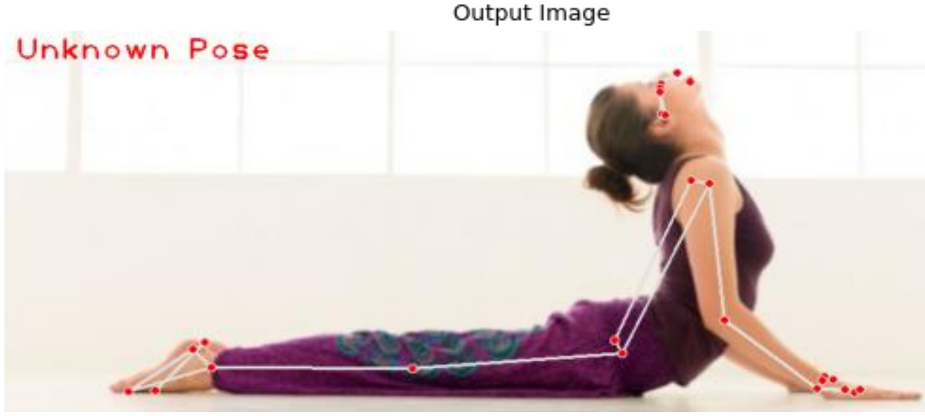
Label: The classified pose label of the person in the output_image.

Output Image

Output Image



Output Image



Chapter 8
Conclusion

In this project, we have developed a system which consists of a pipeline for pose identification and then point localization on human body and then followed by a error identification process. This system aims at helping people to perform yoga in a correct manner on their own and prevent injuries that can happen due to incorrect practices in yoga. Using deep learning techniques the system is able to analyse the pose of the user from the front view and give a feedback to them for improving their yoga pose. The system delivers satisfactory results as shown in the above experimental analysis, but there are some areas for improvement as well, for example the pose identification model provides less accuracy for similar poses. This inaccuracy can be overcome by improving the methods for feature point detection or by redesigning the methods for features extraction which is essential for pose identification. Besides, the working of this system be enhanced by adding more modules of other yoga poses. The future scope for this project is to enhance it by adding voice feedback and well an including use-cases in other field’s like sports, dance etc.

Chapter 9

Bibliography

[1] Y. Hsu, S. Yang, H. Chang and H. Lai, “Human Daily and Sport Activity Recognition Using a Wearable Inertial Sensor Network," IEEE Access, Vol. 6, pp. 31715-31728, 2018.

[2] Chen, H., He, Y. Hsu, C. "Computer-assisted yoga training system" Multimer Tools Appl 77, 23969–23991 (2018). https://doi.org/10.1007/s11042-018-5721-2

[3] N. Belling, "The Yoga Handbook: A Complete Step-by-Step Guide", New Holland Publishers Ltd, 2001

[4] Y. Liu, X. Tian and X. Xu, “Posture estimation system by IMMbased unscented Kalman filters," 2017 Chinese Automation Congress (CAC),Jinan, 2017, pp. 2363-2368.

[5] Sreeni.S, R.H.S, R.H, V.S. (2018)," Multi-Modal Posture Recognition System for Healthcare Applications". TENCON 2018 - 2018 IEEE Region 10 Conference

[6] Patil, Siddharth Pawar, Amey Peshave, Aditya Ansari, Aamir Nevada, Arundhati. (2011)." Yoga tutor visualization and analysis using SURF algorithm." 2011 IEEE Control and System Graduate Research Colloquium

[7] Soltow E., Rosenhan B.(2016)," Automatic Pose Estimation Using Contour Information from X-Ray Images," Huang F., Sugimoto A. (eds) Image and Video Technology – PSIVT 2015 Workshops. PSIVT 2015. Lecture Notes in Computer Science,vol 9555. Springer, Cham.

[8] Singh, A., Agarwal, S., Nagrath, P., Saxena, A., Thakur, N. (2019). "Human Pose Estimation Using Convolutional Neural Networks". 2019 Amity International Conference on Artificial Intelligence (AICAI)

[9] Babiker, Mohanad Khalifa, Othman Htike, Kyaw Hashim, Aisha Zaharadeen, Muhamed. (2017). "Automated daily human activity recognition for video surveillance using neural network" 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)

[10] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1653-1660, doi: 10.1109/CVPR.2014.214.

[11] Mohanty A., Ahmed A., Goswami T., Das A., Vaishnavi P., Sahay R.R. (2017) "Robust Pose Recognition Using Deep Learning". In: Raman B., Kumar S., Roy P., Sen D. (eds) Proceedings of International Conference on Computer Vision and Image Processing. Advances in Intelligent Systems and Computing, vol 460. Springer, Singapore. https://doi.org/10.1007/978-981-10-2107/7/9

[12] M. Verma, S. Kumawat, Y. Nakashima and S. Raman, "Yoga-82: A New Dataset for Fine-grained Classification of Human Poses," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 4472-4479, doi: 10.1109/CVPRW50498.2020.00527.