Personal Exercise Assistant: Correcting Exercise Posture using Modified Open pose

Manisha C. Patel [#1], Nilesh B. Kalani [*2],

1Research Scholar, Electronics and Communication Department, R. K. University, Rajkot, India
1Assistant professor, Instrumentation and Control engineering department, L.D college of engineering, Ahmedabad, India
2 Research Guide and Professor, Electronics and Communication Department, R. K. University, Rajkot, India
mpatel708@rku.ac.in, nilesh.kalani@rku.ac.in

Article Info
Page Number: 10347-10358
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Received: 12 September 2022
Revised: 16 October 2022
Accepted: 20 November 2022
Publication: 25 December 2022

Abstract
Fitness exercises or yoga are very beneficial to personal health and fitness; however, they can also be ineffective and potentially dangerous if performed in correctly by the user. Exercise mistakes are made when the user does not use the proper form, or pose. In our work, we introduce Personal Exercise Assistant, an application that detects the user's exercise pose and provides personalized feedback. Modified Open Pose is used as a Pose estimator in this application. Open Pose is a pre trained model composed of a multi-stage CNN to detect a user's posture. This application then evaluates the vector geometry of the pose through an exercise to provide helpful feedback. Pose estimation is a method in which spatial locations of key body joints is calculated using image or video of the person. Modified open pose is a light weight (8bit Quantized version) Open pose. Personal Exercise Assistant works on yoga exercises and supports any Windows or Linux computer with a GPU.
Keywords: -Pose estimation, CNN, Open Pose, Quantization.

1. Introduction

Yoga exercises are favourable to human body fitness, but it can also be very harmful if the exercises are performed improperly. Due to the lack of training or knowledge many people do not follow the correct posture to be maintained while performing these exercises regularly. This may lead to muscle fatigue and muscle strain. In this research, using Modified Open Pose in pose estimation we help people in performing exercises with correct posture by developing a project that detects the users pose while exercising and provides feedback whether the exercise pose is correct or not[1]. The goal for this project is to prevent injuries and improve the form of human workout with just a computer and a camera.Main aim of this research is to aid people in performing the correct posture for exercises by building Personal Exercise Assistance, a software application that detects the user's exercise pose and provides useful feedback on the user's form, using a combination of the latest advances in Deep learning-based pose estimation.

Researchers use the Open Pose pretrained real-time system for the pose estimation component[1]l.The proposed Personal Exercise Assistance will help to prevent injuries and improve the quality of people's workouts with less hardware and flexibility.A Deep Neural Networks (DNN) based regression problem is used to develop the pose estimate with regard to body joints[2].Open Pose, the first open-source real time system for multi-person 2D pose detection with body, foot, hand, and facial key points, is the culmination of this work[3].The method's modular structure makes it simple to implement without the use of specialised optimization solvers and allows for quick inference[4].

Despite their good performance, it is frequently difficult to determine if their residual inaccuracy results from a limited knowledge of 2D poses or from a failure to translate 2D poses into 3D positions[5].The method produces a richer and more useful mesh representation that is parameterized by shape and 3D joint angles, in contrast to most current methods that compute 3D or 2D joint locations[6].

2. Problem statement:

For the pose estimation component, we utilize a pre trained real-time model, Open Pose[7], that can detect human body key points in videos. For the posture evaluation (pose training) component, we have recorded videos of certified yoga trainer performing various yoga. Our videos include correctly perform yoga, as well as intentionally incorrect examples.Posture identifier using machine learning algorithm was evaluated by splitting video dataset into train and test sets and report results on test data set.Yoga, Taijiquan, and other fitness activities are growing in popularity, yet improper exercise can injure the body as well as have no beneficial fitness results. Professional advice is necessary as a result.Moreover, personal trainers are costly and time-consuming, with a significant gap between the high demand for fitness and the lack of available expert advice.

3. Literature review:

Various Pose estimation models have been studied to evaluate human poses using machine learning approach.

A. Toshev and C. Szegedy[1] improved pose detection using regression on CNN for locating the locations of body joints for the first time. A. Newell, K. Yang, and J. Deng[8] presented a stacked hourglass neural network design that uses a bottom-up and top-down strategy to find position predictions. Single depth maps are used by J. Shotton, A. Fitzgibbon and others[9] to estimate the 3D positions of joints using object recognition. Based on the sequential prediction framework, S. Wei, V. Ramakrishna, and others[10] propose an alternative architecture that employs multiple convolutional networks to clarify joint estimates over successive passes and designs a cascaded CNN network to represent texture and spatial information with convolution layers while sequentially incorporating global context to improve part confidence maps from earlier iterations. Z. Cao, T. Simon, and others[7] used part affinity fields, which collected features from the first 10 layers of the VGG-19[11] without the requirement to recognise individual people, to identify many people in a real-time setting. It has a three-branch CNN architecture that forecasts joint position, limb direction, and orientation while maintaining the initial features in the part affinity field. Because it combines and strengthens the output of three branches, this method has increased the accuracy of regression. In order to give the user feedback for this application without requiring a complete physical simulation, a simpler approach is employed, which involves an analysis of the angles and distances between joint key points.A unique regional multi-person pose estimation (RMPE) framework is suggested that, in terms of precision and effectiveness, greatly beats the most recent techniques for multi-person human pose estimation[12].Providea fresh benchmark MPII Human Pose1 that represents a major improvement in terms of diversity and difficulty, a contribution that we feel is necessary for further advancements in human body modelling[13].a technique for estimating human posture using deep neural networks (DNNs). A DNN-based regression problem is used to develop the pose estimate regarding body joints. It presents a series of these DNN regressors that yields extremely accurate pose estimates[2]. Computer vision research on human position estimate is very popular. In contrast to typical algorithms, deep learning-based algorithms train the networks with a vast number of photos and learn features from the entire world. It has good robustness and doesn't require the local features of objects to be detected or recognised[14].

Santosh Kumar Yadav and Amitojdeep Singh et al. explained convolutional neural network (CNN) and long short-term memory (LSTM) are used in a hybrid deep learning model that is proposed for yoga recognition on real-time videos.When LSTM is used to provide temporal predictions and CNN layer is utilised to extract features from critical points of each frame collected from Open Pose.The algorithm obtains a test accuracy of 99.04% for single frames and 99.38% for 45 frames of the videos after polling predictions.Results from the experiments offer a qualitative evaluation of the approach and a comparison to current best practises[15].

Umer Rafi, Ilya Kostrikov and Juergen Gall et.al. presented Provide a suggestion for an effective deep network architecture that can be trained effectively on midrange GPUs without the requirement for any prior training.On common benchmarks for human pose estimation, our network performs on par with models that are substantially more complicated yet having lower processing requirements.It performs admirably on well-known benchmarks for estimating human poses, which is remarkable given that the model does not need to be pre-trained on massive datasets like other models and can be trained entirely from scratch on modest datasets like Frames Labelled in Cinema[16].

Keiron O’Shea and Ryan Nashhas outlinedthe development of artificial neural networks (ANN); the discipline of machine learning has recently undergone a significant change. These computer models with biological inspiration can execute common machine learning tasks much better than earlier artificial intelligence technologies.The Convolutional Neural Network(CNN) is one of the most amazing ANN architecture types.The fundamental ideas behind convolutional neural networks are presented, along with information on the layers needed to construct one and the ideal network configuration for most image analysis tasks.Recent years have seen a slight slowdown in the study of neural networks for image processing.The researchers hope that this study has helped to clear up some of the misunderstanding and has made the subject easier for newcomers to understand[17].

TianyiLiu,ShuangsangFang, and YuehuiZhao et al are presented that deep learning, which is based on learning levels of representations, is the deepest branch of machine learning. One type of deep neural network is the convolutional neural network (CNN).It provided a thorough analysis of the CNN algorithm's forward and backward propagation processes.the convolutional neural network was then used to implement the common Java face recognition challenge.Analysed the theoretical maximum speed up and parallel efficiency by measuring the actual time of forward and backward processing.Although the parallelism used is coarse-grained, the technique still has several modules that can be fine-grained. In order to keep improving the job, this will be our future priority[18].

Shih-EnWei,Varun Ramakrishna and Takeo Kanade et al are presented a methodical approach for integrating convolutional networks into the pose machine framework for the purpose of learning picture features and image-dependent spatial models for posture estimation.to implicitly represent long-range connections between variables in structured prediction tasks, like articulated pose estimation.this without the need of explicit graphical model-style inference by creating a sequential architecture made up of convolutional networks that directly operate on belief maps from earlier stages and produce increasingly accurate estimates for part positions.Convolutional network sequence architecture is capable of implicitly developing a spatial model for pose by exchanging ever-finer uncertainty-preserving beliefs between stages.Future work will address adapting our architecture to these issues because spatial dependencies between variables are a challenge in many computer vision applications, including semantic picture labelling, single image depth prediction, and object recognition.On all major benchmarks, our method achieves state-of-the-art accuracy, but we do occasionally see failures, usually when several people are nearby.Another difficult issue and promising area for future research is managing numerous persons in a single end-to-end architecture[10].

Leonid pishchulin, Eldarinsafutdinov and Siyu tang at el are describedthe process of estimating numerous people's poses in real-world photographs using articulated human motion.the suggest a method for concurrently resolving the detection and posture estimation problems: It determines how many people are present in a scene, recognises obscured body parts, and distinguishes between body parts of individuals that are near to one another.In contrast to earlier approaches, this joint formulation tackles the issue by first recognising persons and then assessing their body posture.the suggest a formulation for partitioning and labelling a set of body-part hypotheses produced by CNN-based part detectors.a sample integer linear programme that groups the set of part candidates into configurations of body parts while implicitly performing non-maximum suppression on the set of part candidates.Research on four separate datasets show cutting-edge outcomes for both single person and multiple person posture estimation.The subset partitioning and labeling problem (SPLP) model jointly infers the number of persons, their poses, spatial closeness, and part level occlusions, in contrast to earlier two-stage systems that separated the detection and pose estimating phases[19].

4. Technical Approach

4.1 Pipeline overview

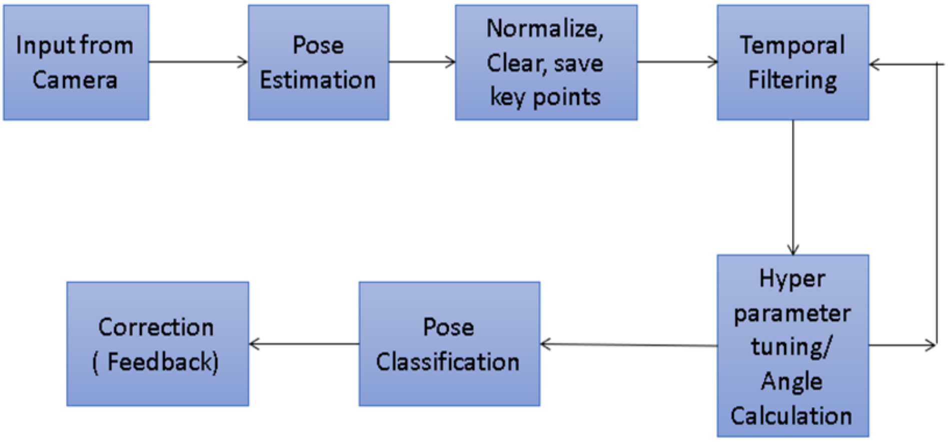


Figure 1. Personal Exercise Assistant pipeline

We now describe the Personal Exercise Assistant application from a technical perspective as a pipeline system, consisting of multiple system stages (see Figure 1). Input can be Image or video. The Resolution of camera – 720 Pixel or 1080 Pixel. Pose estimation consists of various stages like dataset, annotation, training, validation, testing and Model deployment. Once we get the coordinates of body landmarks, we need to normalise for rest of calculation. Cleaning is required for non-maxima suppression. It's like we get multiple face points and out of which point has strong confidence. Temporal filtering helps to stabilize the values. Let's say if there are wrong results or discontinuity or fluctuations then we can reduce, stabilize, and predict the actual possibilities of values. Angle calculation is a mathematical technique of line intersections which will be applied at different joints like armpit, waist, knee, and elbow. Using different angles, pose classification will be carried out by applying various conditions. Once pose is classified, then we can keep tracking various angles and if there is any change or variations then we can give feedback to user for Posture control.

4.2 Video Recording

The users must first record a video of themselves completing a specific activity from a specific point of view (front, rear, etc.) for the exercise to be effectively observed. There are no limitations on the user's distance from the camera or the type of camera; the only thing the user must ensure is that their posture is seen. Furthermore, the user must ensure that the recorded video only contains workout frames. Any video editing software can be used to complete the task. All common video formats supported by Open Pose are available on current PCs and devices. The essential tasks in the training of key point detection in computer vision models are classification and localization.

4.3 Pose Estimators

We deploy deep convolutional neural networks (CNNs) to identify RGB images for pose estimation. We selected Open pose, pre-trained model, for pose recognition after experimenting with numerous state-of-the-art pose estimators. Part affinity fields, or vectors that encode the position and orientation of limbs, are used in Open Pose to introduce a novel approach to posture estimation. The model is made up of a multi-stage CNN with two branches: one for learning the confidence mapping of a key point on an image and the other for learning part affinity fields.

Another major element in our decision to choose Open Pose is the ease with which our end-users can install and use it. Most posture estimators are currently available as Tensor flow or Caffe source code, which contains the model architecture, and requires difficult user installations and weight, downloads to be usable. For these estimators to work successfully, GPU libraries such as CUDA and CuDNN must also be installed. This is much beyond the capabilities of most computer users. Open Pose, on the other hand, is available in executable format for Windows and Linux, requiring no installation or programming experience to use. Furthermore, no GPU libraries must be installed externally: If the machine has an NVIDIA GPU, Open Pose will use it automatically.

The Open Pose output comprises of lists providing all key point location coordinate predictions as well as their accompanying prediction confidence. The nose, neck, shoulders, elbows, wrists, hips, knees, and ankles are among the 18 key points of the pose that we analyse for forecasts.

4.4 Key point Normalization

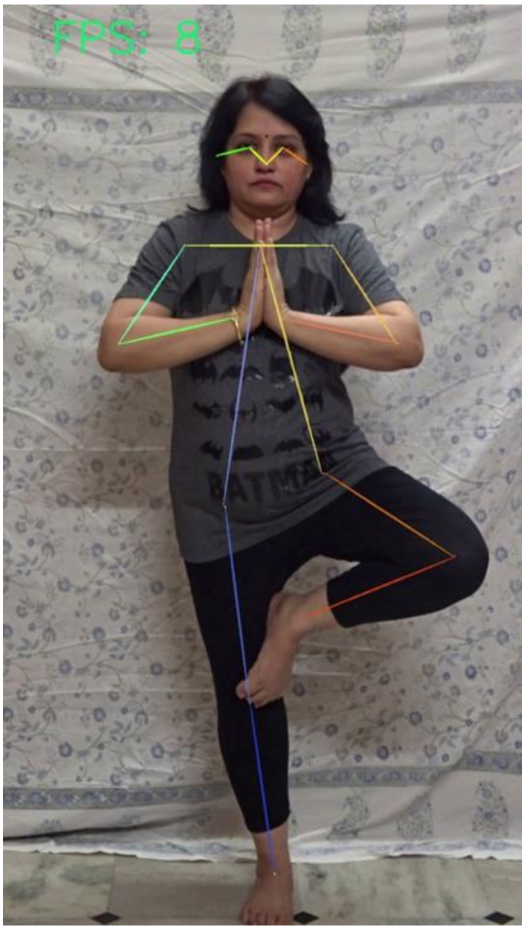
Key point normalisation contains a list of key point predictions for each video and part objects that record key point confidence. A joint key point is constructed to build a posture object for every frame in the video to depict comprehensive skeletal prediction of a human stance. Each video frame for the entire video is combined into a pose sequence object. The technique is generalised to account for variations in body length, camera quality, and distance from the camera, and other aspects.

A specific exercise must be captured from a specific camera perspective. When performing a front raise, the video should be taken from the side of the body and executed with either the left or right arm. Body vectors are calculated using key points in geometric evaluation.

4.5 Model Quantization

The process of converting deep learning models to employ parameters and computations with less precision is known as quantization[20]. The IEEE single-precision floating-point format, which uses 32 bits to express the floating-point model weights and activation tensors, has generally been used for DNN training and inference. Since most DNNs are trained in data centres or the cloud using NVIDIA V100 or A100 GPUs, which have a notably big compute capabilities and much larger power budgets, this computational budget might be acceptable for training[20]. However, during deployment, it is frequently necessary for these models to operate on edge devices with significantly lower compute power and power budgets. The edge's compute, memory, and power limits make it impractical for real-time analysis to run a DNN inference utilising the entire 32-bit representation. Running inference with less precision can significantly reduce the compute budget without sacrificing the model's structure or number of parameters. Tensors and weights were first represented as 16-bit floating-point numbers, and quantized inferences were performed with half-point precision. Even while this produced compute savings of roughly 1.2–1.5x, some compute budget and memory bandwidth might still be used.

However, models are now quantized to even lower precision, with weights and tensors represented as 8-bit integers. As a result, the model's memory footprint is reduced by 4 times, while its throughput is increased by 2-4 times.



Open Pose(Without Quantization) fps- 8 Open Pose (After Quantization) fps- 66

Figure 2. Comparison of Open pose and Modified Open pose

Figure 2 shows Pose estimation of a single person. If we compare both images of figure 1, from the pose estimation viewpoint, key-point location in both images are almost same while Modified Open pose is lighter (8 bit -Quantised) than Open pose, the FPS (Frames per seconds) is higher in Modified Open pose. For real time embedded application, Light weight model perform better.

4.6 Machine learning Evaluation

MobileNetV2 is a state of the art for mobile visual recognition including classification, object detection and semantic segmentation[21].Depthwise separable convolution is used as an effective building element in MobileNetV2.Two additional functionalities are added to the architecture by MobileNetV2: 1) bottlenecks that are linear between the layers, and 2) connections that are quick between bottlenecks.

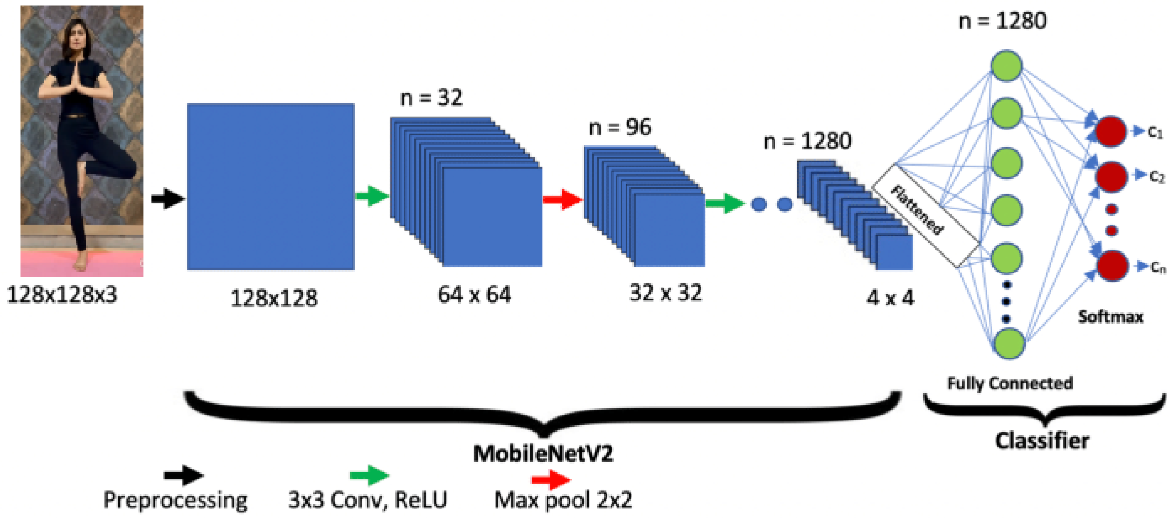


Figure-3 MobilenetV2 Network architecture

The inner layer contains the model's capacity to transition from lower-level ideas like pixels to higher-level descriptors like image categories, while the bottlenecks encode the model's intermediary inputs and outputs.Finally, shortcuts allow for quicker training and improved accuracy, much like with conventional residual connections.

5. Results:

The Modified Open pose is a 8bit quantized version of Open pose. Modified Open pose has been trained and tested on a system with ASUS ROG Strix G17 17.3" FHD 120Hz Intel Core i7-10750H 10th Gen, GTX 1660Ti 6GB Graphics (16GB RAM/512GB NVMe SSD/Windows 10.

We present quantitative and qualitative results of Personal Exercise Assistant on two different yoga exercises: Tree Pose and Worrier 2 Pose

5.1 Tree Pose

Tree Pose (Vrksasana) teaches you to simultaneously press down and feel rooted as you reach tall like the branches of a mighty tree. Tree Pose can improve your posture and alignment, which is especially helpful if you sit throughout the day.

For our ML algorithm, we split our 15-video Tree Pose dataset into 9 training examples and 6 test examples. The MobilenetV2 classifier achieves an F1 score of 0.85. The precision, recall, and F1 for all exercises are summarized in table 1. Figures 3 illustrate examples of good and bad form, as well as the pose angle statistics that we use when evaluating a user's exercise.

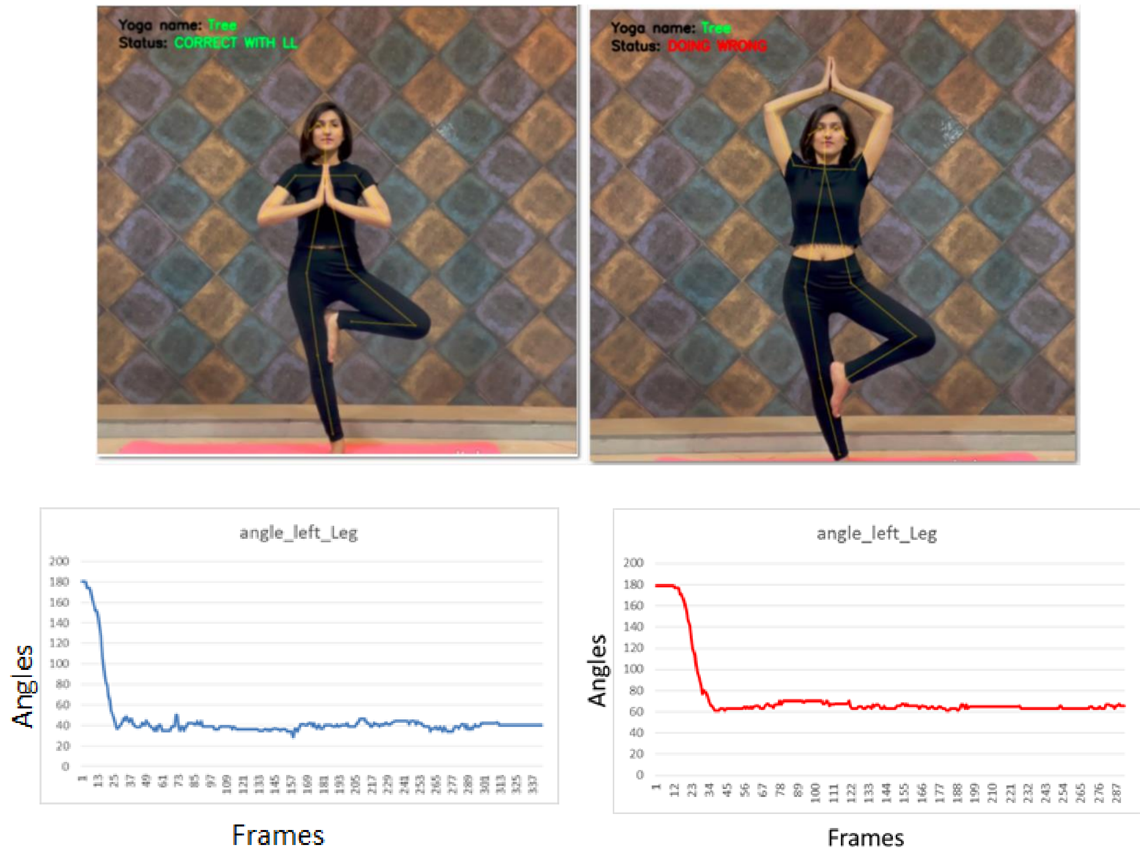


Figure 3. In a Tree Pose, bend your left knee and place the left foot high up on your right thigh. Make sure that your left right is straight. Find your balance and bring your palms together in 'Namaste' mudra. Leftshows proper angle of left knee form (40 degree): right shows improper form, wherethe left leg is not properly bent.

5.2 Worrier 2 Pose

In the pose, your front knee bends to create a stretch in your hips, your arms engage and extend straight out from your shoulders, and your gaze remains calm and steady toward your front hand.

For our ML algorithm, we split our 18-video Tree Pose dataset into 12 training examples and 6 test examples. The MobilenetV2 classifier achieves an F1 score of 0.88. The precision, recall, and F1 for all exercises are summarized in table 1. Figures 4 illustrate examples of good and bad form, as well as the pose angle statistics that we use when evaluating a user's exercise.

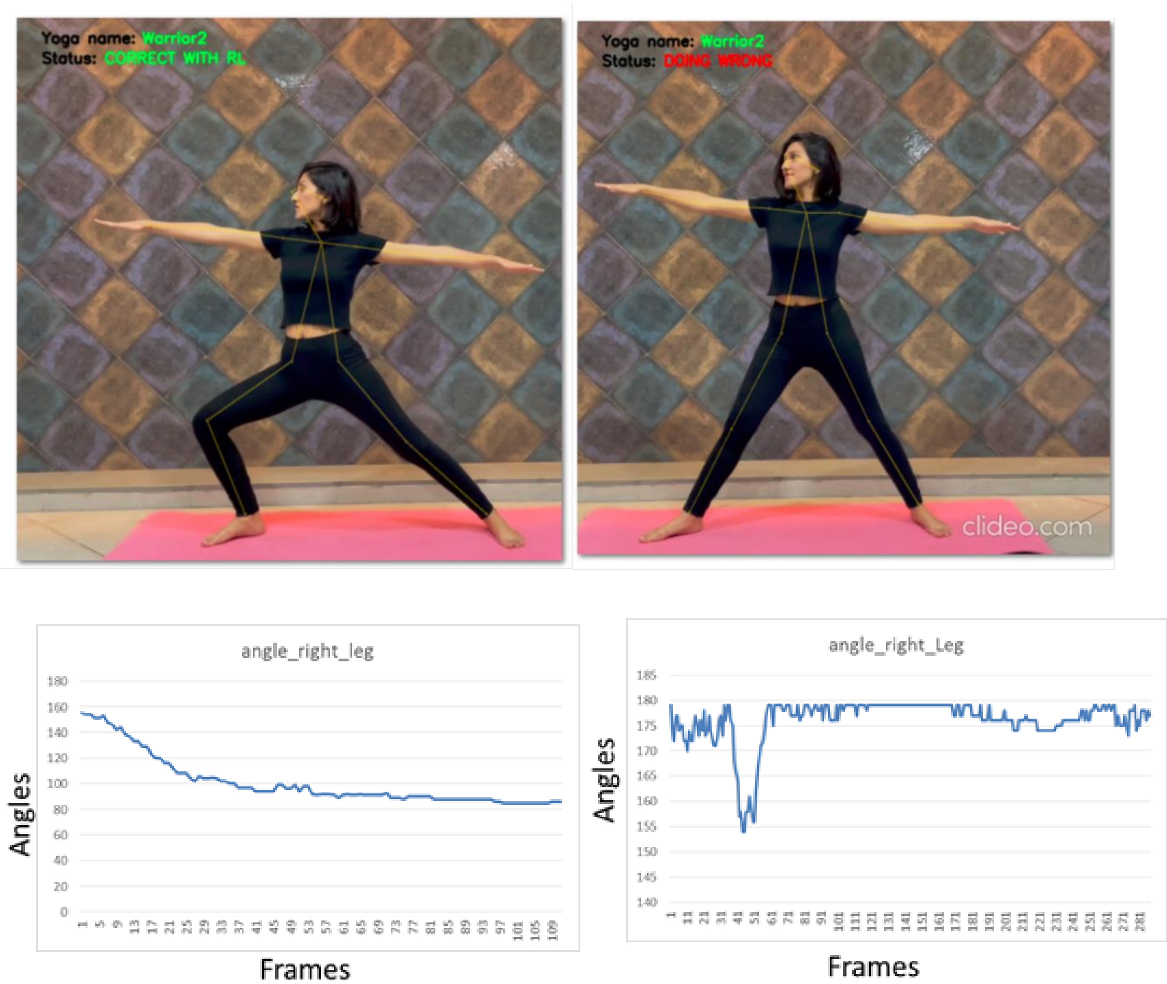


Figure 4. In a Worrier Pose, spread your legs around three to four feet wide. Place your hands on your waistline. Move your right toe in an angle of 90-degrees and the left one at an angle of 15-degrees. Turn your neck towards your right and fix your gaze ahead. Extend your hands in each direction. Your palms must face the ground. Left shows proper angle of right leg form (90 degree): right shows improper form, where the Right leg is not properly bent.

Table 1. Confusion matrix for our MobilenetV2 classification model

Tree Pose				
	Precision	Recall	F1 score	Examples
Correct	0.80	0.95	0.89	4
Incorrect	0.90	0.70	0.80	2
Avg/Total	0.89	0.84	0.85	6
Worrier 2 Pose				
	Precision	Recall	F1 score	Examples

Correct	0.85	1.00	0.95	4
Incorrect	0.90	0.88	0.85	2
Avg/Total	0.89	0.92	0.88	6

6. Conclusion and Future work

In this paper introduce Personal Exercise Assistant, an end-to-end computer vision application that uses pose estimation, machine learning to provide personalized feedback on yoga poses. We use the output of pose estimation to evaluate videos of exercises through human pose key points. Initially we work with two yoga, recording training videos for each and use machine learning algorithm to automatically determine posture correctness using only labelled input videos. Further we have lightened pose estimation model using 8-bit quantization techniques so lighten version can be implement on CPU or any embedded device.

In this paper highlightedseveral expansions as promising areas for additional work after this course project.One option would be to adapt Personal Exercise Assistant to cellphones and create an app that would enable users to record videos and receive feedback on their poses whenever and wherever they pleased.Another step would be to enhance pose feedback, including precise recommendations on the user's pose's problem areas (such as the back, neck, and shoulders), and suggesting targeted action.

References:

[1] S. Chen and R. R. Yang, "Pose Trainer: Correcting Exercise Posture using Pose Estimation," no. March, 2020, doi: 10.13140/RG.2.2.29224.47367.

[2] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1653–1660, 2014, doi: 10.1109/CVPR.2014.214.

[3] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, 2021, doi: 10.1109/TPAMI.2019.2929257.

[4] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8690 LNCS, no. PART 2, pp. 33–47, 2014, doi: 10.1007/978-3-319-10605-2_3.

[5] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A Simple Yet Effective Baseline for 3d Human Pose Estimation," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 2659–2668, 2017, doi: 10.1109/ICCV.2017.288.

[6] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-End Recovery of Human Shape and Pose," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7122–7131, 2018, doi: 10.1109/CVPR.2018.00744.

[7] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition,*

CVPR 2017, vol. 2017-Janua, pp. 1302–1310, 2017, doi: 10.1109/CVPR.2017.143.

[8] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9912 LNCS, pp. 483–499, 2016, doi: 10.1007/978-3-319-46484-8_29.

[9] J. Shotton *et al.*, "Real-Time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, 2013, doi: 10.1145/2398356.2398381.

[10] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 4724–4732, 2016, doi: 10.1109/CVPR.2016.511.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[12] H. S. Fang, S. Xie, Y. W. Tai, and C. Lu, "RMPE: Regional Multi-person Pose Estimation," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2353–2362, 2017, doi: 10.1109/ICCV.2017.256.

[13] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3686–3693, 2014, doi: 10.1109/CVPR.2014.471.

[14] J. Zou *et al.*, *Intelligent fitness trainer system based on human pose estimation*, vol. 550. Springer Singapore, 2019.

[15] S. K. Yadav, A. Singh, A. Gupta, and J. L. Raheja, "Real-time Yoga recognition using deep learning," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 9349–9361, 2019, doi: 10.1007/s00521-019-04232-7.

[16] U. Rafi, I. Kostrikov, J. Gall, and B. Leibe, "An efficient convolutional network for human pose estimation," *Br. Mach. Vis. Conf. 2016, BMVC 2016*, vol. 2016-Septe, pp. 109.1-109.11, 2016, doi: 10.5244/C.30.109.

[17] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," no. December, 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.

[18] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of Training Convolutional Neural Networks," 2015, [Online]. Available: http://arxiv.org/abs/1506.01195.

[19] L. Pishchulin *et al.*, "DeepCut: Joint subset partition and labeling for multi person pose estimation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 4929–4937, 2016, doi: 10.1109/CVPR.2016.533.

[20] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021, doi: 10.1016/j.neucom.2021.07.045.

[21] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, [Online]. Available: http://arxiv.org/abs/1704.04861.