2023

A Real-time Machine Learning Framework for Smart Home-based Yoga Teaching System

Jothika Sunney
National College of Ireland, Ireland

Musfira Jilani
National College of Ireland, Ireland

Pramod Pathak
Technological University Dublin, Ireland, pramod.pathak@tudublin.ie

See next page for additional authors

Funder: This research received no external funding

Authors
Jothika Sunney, Musfira Jilani, Pramod Pathak, and Paul Stynes

2023 7th International Conference on Machine Vision and Information Technology (CMVIT) | 978-1-6654-6485-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/CMVIT57620.2023.00029

A Real-time Machine Learning Framework for Smart Home-based Yoga Teaching System

1st Jothika Sunney
School of Computing
National College of Ireland
Dublin, Ireland
jothikasunney@gmail.com

2nd Musfira Jilani
School of Computing
National College of Ireland
Dublin, Ireland
musfira.jilani@ncirl.ie

3rd Pramod Pathak
Faculty of Digital&Data
Technological University Dublin
Dublin, Ireland
pramod.pathak@TUDublin.ie

4th Paul Stynes
School of Computing
National College of Ireland
Dublin, Ireland
paul.stynes@ncirl.ie

***Abstract*—Practicing yoga poses in a home-based environment has increased due to Covid19. Yoga poses without a trainer can be challenging, and incorrect yoga poses can cause muscle damage. Smart home-based yoga teaching systems may aid in performing accurate yoga poses. However, the challenge with such systems is the computational time required to detect yoga poses. This research proposes a real-time machine learning framework for teaching accurate yoga poses. It combines a pose estimation model, a pose classification model, and a real-time feedback mechanism. The dataset consists of five popular yoga poses namely the downdog pose, the tree pose, the goddess pose, the plank pose, and the warrior pose. The BlazePose model was used for yoga pose estimation which transforms the image data into 3D landmark points. The output of the pose estimation model was then passed to the pose classification model for yoga pose detection. Four machine learning classifiers namely, Random Forest, Support Vector Machine, XGBoost, Decision Tree, and two neural network classifiers LSTM and CNN were evaluated based on accuracy, latency and size. Results demonstrate that XGBoost outperforms other models with an accuracy of 95.14 percentage, latency of 8 ms, and size of 513 KB. The output of the XGBoost Classifier was then used to correct yoga poses by displaying real-time feedback to the user. This novel framework has the potential to be integrated into mobile applications which can be used by people for the unsupervised practice of yoga at home.**

***Index Terms*—Yoga Pose Detection; Machine Learning Framework; BlazePose; XGBoost.**

I. Introduction

There has been an increase in home-based yoga practitioners after covid19 [1], [2]. However, performing incorrect yoga poses can cause ligament pains and muscle damage [3]. Aritifical Intelligence (AI) systems with human pose estimation techniques can be used for teaching yoga poses accurately. But the major challenge with existing systems is that they require a lot of computing resources and are not suitable for real-time applications. Also, real-time yoga pose detection is difficult to estimate due to its variety, degrees of freedom, occlusions, and variations in appearance [4]. A computationally inexpensive machine learning framework with good accuracy is required for smart home-based yoga pose teaching systems. Yoga pose detection falls under human pose estimation which is a computer vision technique that predicts and tracks the location of a person.

BlazePose [4] is a popular human pose estimation model under Mediapipe open-source platform. BlazePose is uniquely suitable for fitness applications as it can accurately locate more landmark points than other pose estimation models. Additionally, it gives real-time performance.[1]

The aim of this research is to investigate to what extent a real-time machine learning framework can be used for teaching an accurate yoga pose. The major contribution of this research is a novel framework that combines a pose estimation model, pose classification model and real-time feedback mechanism to detect five different yoga poses for unsupervised practice at home. The pose estimation model is based on the BlazePose model. The pose classification model is based on XGBoost and the real-time feedback mechanism is based on a comparative algorithm and OpenCV. For the pose classification, this research compares four machine learning classifiers namely, Random Forest, Support Vector Machine, XGBoost, Decision Tree, and two neural network classifiers LSTM and CNN.

II. Related Work

Smart home-based yoga teaching systems are AI-based systems that help users to perform yoga poses accurately without any trainer. Yoga pose detection has emerged as an important field of research under human pose estimation. Even before pose estimation models there were efforts to create automated and semi-automated systems to analyze exercise and sports activities [5]–[8]. Patil et al. [5] used the surf algorithm to detect the yoga pose based on the video provided. SURF is a robust image detector and descriptor which can be used for various image transformations. The method is inaccurate as only the contour information was considered for pose prediction. Researchers [6]–[8] have used Kinect (depth sensors) based yoga pose detection approaches. Although this method gave good accuracy of 94%, depth sensor-based cameras are expensive and are not accessible to ordinary users. Additionally depth sensors are sensitive to light and difficult to use outdoors.

Deep neural networks have revolutionized human pose estimation from traditional systems. DeepPose was the first

[1] https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html

978-1-6654-6485-7/23/$31.00 ©2023 IEEE
DOI 10.1109/CMVIT57620.2023.00029

implementation of a deep neural network for human pose estimation. Toshev and Szegedy [9] proposed the DeepPose pose estimation model. Compared to the traditional methods, DeepPose provides better results on challenging datasets with improved performance. Other human pose estimation models PoseNet, AlphaPose [10], [11] were also used for yoga pose detection. These pose estimation models do not support 3D pose estimation. Also, it does not deal with articulations properly. Additionally, these models did not perform well when applied to video frames, so it's not efficient for real-time pose tracking.

OpenPose is a widely used pose estimation model. Chaudari et al. [12] proposes a real-time yoga pose correction system combining OpenPose and deep learning methods. OpenPose pose estimation model was used to identify 15 landmark points from the image frame which was then passed to a CNN model for pose classification and a real-time feedback was provided back to the user for pose correction. Eventhough the model achieves an overall accuracy of 95%, It failed to distinguish similar poses and required GPU for implementation. OpenPose pose estimation models with deep learning classifiers were investigated by researches [13]–[16] for yoga pose prediction. These models were computationally expensive as OpenPose requires GPU for execution and also the complexity of deep learning classifiers may lead to higher latency and model size. Thar et.al [17] used OpenPose algorithm to detect landmark points and calculated the angles between landmark points to detect yoga poses. In this method, each yoga pose's angles must be identified beforehand, which seems difficult to accomplish for a wide variety of poses.

A pose estimation solution from Mediapipe framework is known as BlazePose. Bazarevsky et al. [4] proposes BlazePose architecture for human pose estimation. it has a lightweight convolutional neural network based architecture. BlazePose is suitable for real-time inference on mobile devices. It can produces 33 body landmark points for a single person, and runs at over 30 frames per second. Real-time capabilities make it particularly suitable for applications such as fitness tracking and sign language recognition. BlazePose model and angular metrics calculations were used by Anilkumar et al. [18] to develop a yoga posture correction system. By using BlazePose, 33 landmark points from the yoga posture were extracted. Then angles between the landmark points were calculated to identify the yoga pose. Although this method is effective for yoga pose detection and correction, it has the disadvantage of requiring separate angular calculations for each pose. An interactive yoga recognition system based on rich skeletal joints was described by Lo et al. [19], An LSTM neural network combined with Mediapipe's holistic model was used to classify five yoga poses with an accuracy of 85%. The holistic model detects 543 landmarks, but it includes unnecessary face and palm landmarks which increases the model complexity. Garg et al. [20] proposed a model called yogaConvo2d by combining BlazePose and CNN algorithms. Instead of obtaining landmark points from the skeletal images, the skeleton image from BlazePose was directly passed to a 2D CNN for pose classification. Using skeletal images instead of landmark points may add complexity to a model and may require more time for training. While this method achieved 99.62% accuracy on test data, the research did not analyze its performance on real-time prediction.

In conclusion, one of the main challenges of existing yoga pose estimation algorithms is the computational time and cost. Although OpenPose has been well established for pose estimation, it is computationally expensive and requires GPU for high performance. Depth sensors were used by existing systems to calculate distances between the user and camera, but this method is not reliable as it is not available to all users. The research also found that an angular heuristic is commonly used for pose classification. However, it is not a suitable method as each pose requires separate angular calculations. The BlazePose model outperforms existing pose estimation frameworks for real-time yoga/fitness applications. Also it provides 3D feature extraction without requiring depth sensors. There has not been much research combining BlazePose and machine learning algorithms. A novel approach is proposed in this study by combining BlazePose model with XGBoost classifier for teaching yoga poses accurately in real-time.

III. Methodology

The research methodology consists of five steps namely Data gathering, Data preprocessing, Data transformation, Data modelling, Evaluation and Results as shown in Fig.1.

The first step, *Data Gathering* involves identifying the yoga pose dataset. A publicly available yoga dataset from kaggle was used in this research to classify five commonly used yoga poses [2]. The yoga poses examined in this study were down-dog(320 images), goddess(260 images), plank(381 images), tree(229 images), and warrior(361 images). A total of 1551 yoga pose images were included in this collection, consisting of different individuals with diverse background. Yoga poses were organized into folders with names corresponding to the respective yoga pose.

The second step, *Data preprocessing* involves converting the yoga pose image dataset into skelton images and then feature extracting 33 3D landmark points in x,y,z directions using BlazePose pose estimation model. With BlazePose, a single image frame provides 33 3D landmarks. The 33 landmark points detected by the BlazePose model are shown in Fig 2.

Each yoga pose was determined by the x, y, and z co-ordinates of the 33 landmark points. Firstly the image was read using the OpenCV library. OpenCV reads data in BGR format, but BlazePose requires RGB input. In the first step of preprocessing, the image was converted from BGR to RGB. To detect the skeleton image and extract 33 landmark points, the image was passed to the BlazePose model. The extracted 33 landmark data in x,y,z directions for each image were appended to a CSV file. Additionally, the folder names indicating the yoga pose class name were appended to the

[2] Yoga Pose Dataset :https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset

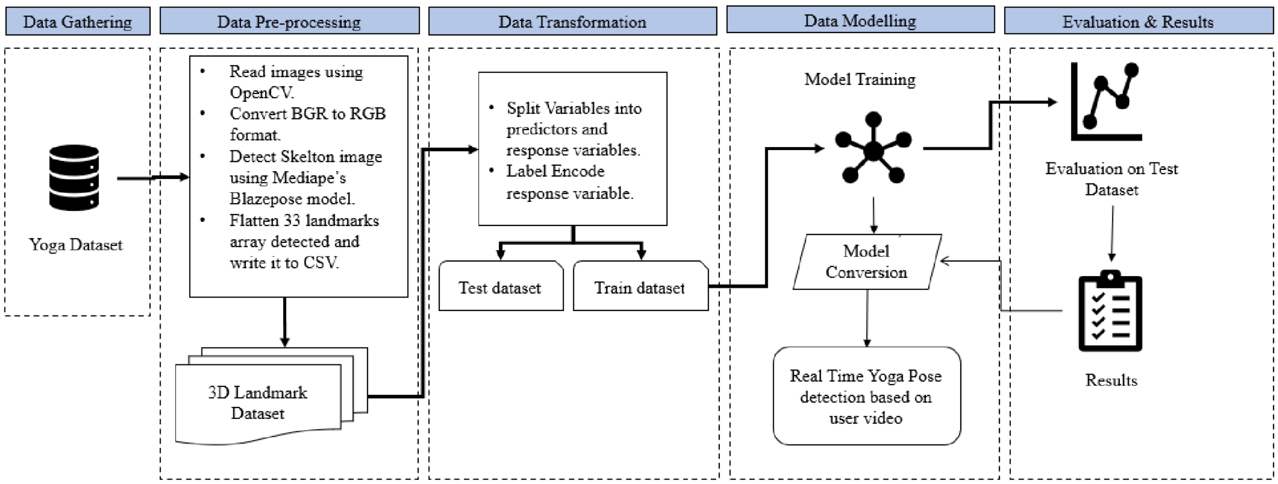


Fig. 1: Research Methodology

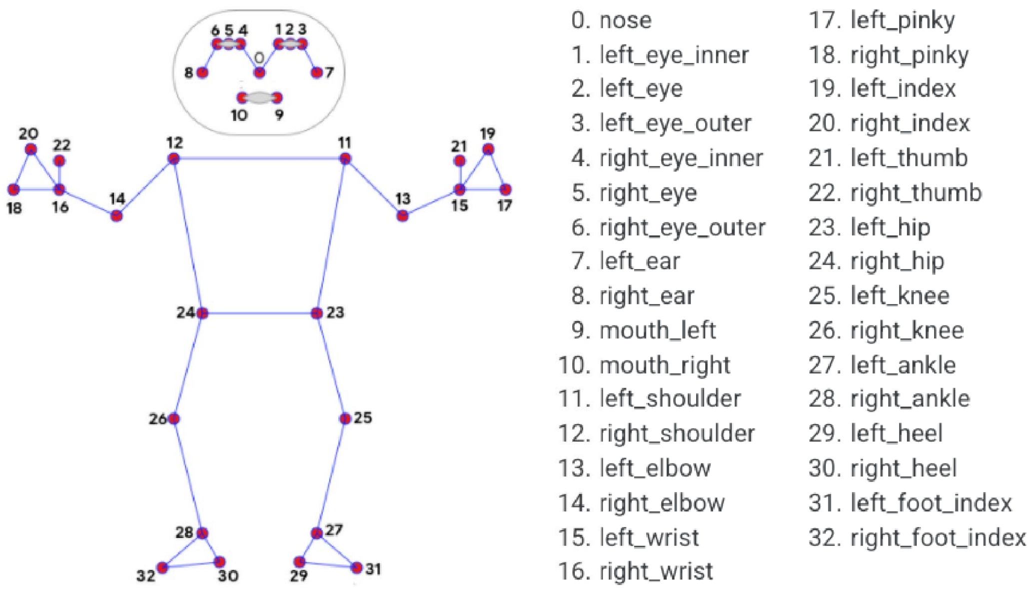


Fig. 2: BlazePose Model - 33 landmark points [4]

CSV file as the response variable. In the generated dataset, the independent variables were standardized 33 landmarks in x, y, and z directions. The variables were defined as x1, y1, z1...x33, y33, z33, and added as headers to the CSV file. Fig.3 shows the 33 3D landmarks data generation on warrior pose using BlazePose model.

The third step, *Data transformation* stage separates the data into response and predictor variables. The response variable, which indicates five different yoga poses, was label encoded before applying to the classification model. The dataset was then divided into train and test datasets according to the 70:30 ratio.

The fourth step, *Data Modelling* involves training and evaluating four machine learning models (Random Forest, XGBoost, SVM classifier, Decision Tree) and two deep learning models (LSTM, 1D CNN). Hyperparameter optimization coupled with cross-validation was implemented to select the best-fit parameters for the machine learning models. Before applying deep learning models to the landmark dataset, it needs to be reshaped. For LSTM networks, CSV input was reshaped into an array of shape (sample size,1,99), where 99 is the total number of features(33*3 dimensions). For the 1D CNN network, the input was reshaped into an array of shape (sample size, 99,1). Both LSTMs and 1D CNNs were trained

Fig. 3: 3D Landmarks data on warrior pose using BlazePose model

using 200 epochs with cross-entropy loss functions, ReLU activation functions, and SoftMax activation functions. The models were optimized using Adam optimizer and evaluated based on categorical accuracy and loss.

In the final step, *Evaluation and Results*, all models were evaluated based on accuracy, latency and size. As per the experimentation, XGBoost Classifier was determined to be the most suitable model for real-time yoga pose prediction. Based on the predicted yoga pose, a real-time feedback was then provided to the user using a comparative algorithm and OpenCV.

IV. Design

The Real-time machine learning framework architecture combines pose estimation model, pose classification model, and real-time feedback mechanism. Fig.4 shows the System Architecture.

A. Pose Estimation Model

Pose estimation model is initiated when a user performs yoga infront of a webcam. Live video from the user's webcam is captured using OpenCV's VideoCapture function. This function allows the creation of a video capture object for the webcam. Each image frame is converted from BGR to RGB format and then given as an input to BlazePose model. A two-step detection model is used by BlazePose, consisting of a detection phase and a tracking phase. By using a detector, the pose region-of-interest (ROI) is located. The tracker predicts all 33 pose landmark points from this ROI. It uses only the first frame of the video to detect landmark points. In subsequent frames, ROIs are extracted from landmark points in the previous frame. The detector is triggered only if the ROI cannot be detected [4]. This method improves the performance of BlazePose, making it suitable for real-time applications. BlazePose extracts 33 3D landmark points in x,y,z directions for each image.

B. Pose Classification Model

Pose classification model consists of pretrained XGBoost classifier. The 33 3D landmark points detected by the BlazePose model is flattened, converted into a dataframe, and inputted into the XGBoost classifier for prediction. Using the XGBoost classifier, the correct yoga pose is detected for each image frame.

C. Real-time Feedback Mechanism

A real-time display of the detected yoga pose, the probability of pose classification, and grades were provided as feedback. Based on a comparative algorithm, the grades are categorized as *'Very Good'*, *'Good'*, and *'Needs Improvement'*. A score of *'Very Good'* will be awarded if the prediction probability exceeds 0.95. The prediction probability less than 0.95 and more than or equal to 0.90 will receive a *'Good'* score, and the prediction probability greater than or equal to 0.85 will receive a *'Needs Improvement'* score. The system will display *"No Pose Detected"* for all probabilities below 0.85. A dialogue box with feedback, image frames, and detected landmarks was displayed using the OpenCV library. When the user presses "q" key, then the video capture object will be released and all opened windows will be closed.

V. Implementation

The real-time machine learning framework for smart home-based yoga pose teaching system was implemented by combining the BlazePose model, XGBoost Classifier, and computer vision methodologies. Jupyter Notebook with Python 3.8.8 was used throughout the research. The main libraries used for the research were Mediapipe, OpenCV, os, Pandas, NumPy, Sklearn, Tensorflow, and Keras. The model was trained using a publically available kaggle dataset. There were 1551 yoga pose images in the dataset, divided into 5 different yoga classes . BlazePose pose estimation model extracts x,y, and z 3D components for each of the 33 landmark points. A CSV file was created by flattening the 3D landmark array. Generated dataset was then divided into 70:30 ratios for training and

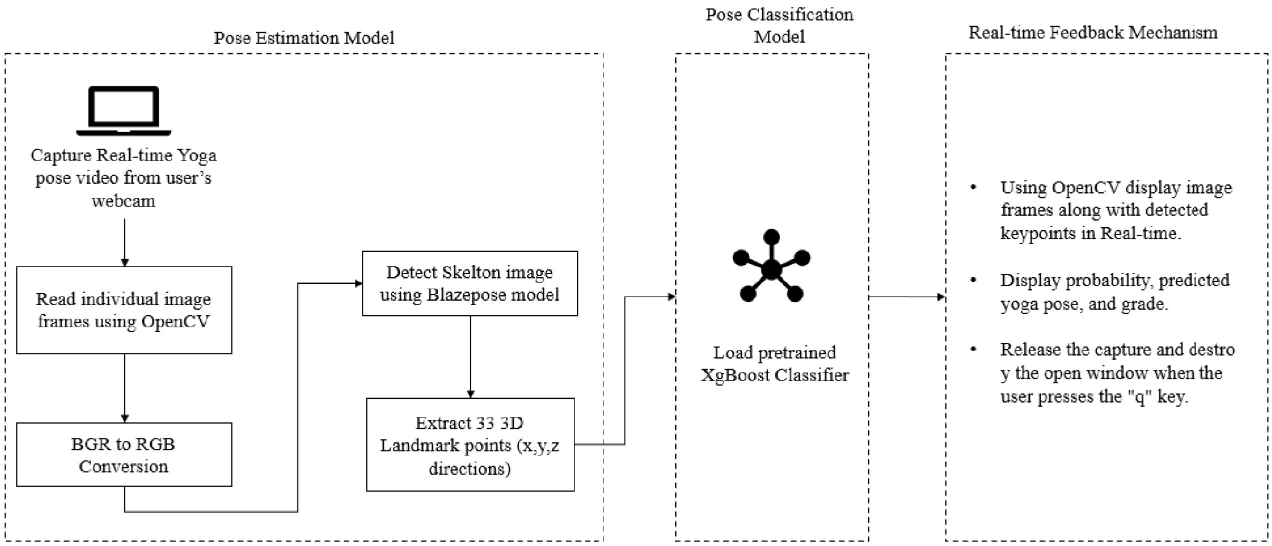


Fig. 4: System Architecture

testing, respectively. The Sklearn library was used to evaluate machine learning models and Keras with Tensorflow backend was used for implementing deep learning models. In order to select the best-fit parameters for the machine learning models, hyperparameter optimization along with cross-validation was applied. Based on the accuracy, latency, and size, the XGBoost classifier was selected as pose classification model for the real-time yoga pose prediction framework. In order to provide real-time feedback to the user, OpenCV library functions were used. All the experiments were carried out on a single CPU with 8GB RAM.

VI. Results and Discussion

This section discusses the results obtained from the experiments conducted in the research. Section A evaluates performance of pose classification models and section B evaluates performance of real-time machine learning framework.

A. *Performance of pose classification models*

The aim of the first experiment was to compare the performance of different machine learning algorithms and deep learning algorithms on 3D landmark dataset generated by the BlazePose model. Four machine learning models XGBoost Classifier, SVM, Random Forest, Decision Tree, and two deep learning models LSTM and CNN were analyzed based on accuracy, latency, and Size. Fig.5 shows the comparison of different classifier models based on accuracy. This result indicates that XGBoost has the highest accuracy of 95.14%.

As the research focuses on real-time yoga pose detection, latency plays an important role. Latency was measured by calculating the time required for predicting the test data. Lower latency means the model is able to make the prediction faster.

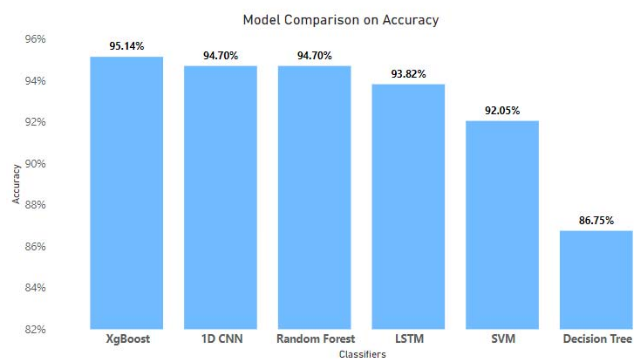


Fig. 5: Model comparison based on accuracy

Latency was measured in milliseconds. Fig.6 shows that all the models have latency within 100 ms. Also, as expected, the traditional machine learning models have low latency as compared to deep learning models. XGBoost Classifier and Decision Tree have the lowest latency of 8 ms, whereas LSTM and 1D CNN have the highest latency of 100 ms. Random Forest and Support Vector have a latency of 34 ms and 24 ms, respectively. As the Decision Tree have the least accuracy, it can not be considered as the final model. With the highest accuracy of 95.14% and the smallest latency of 8ms, XGBoost is suitable for real-time yoga pose detection without any delay. Models were then compared based on size.

As the complexity and parameters of the model increase, the size of the model also increases. The real-time yoga pose detection framework developed in the research can be

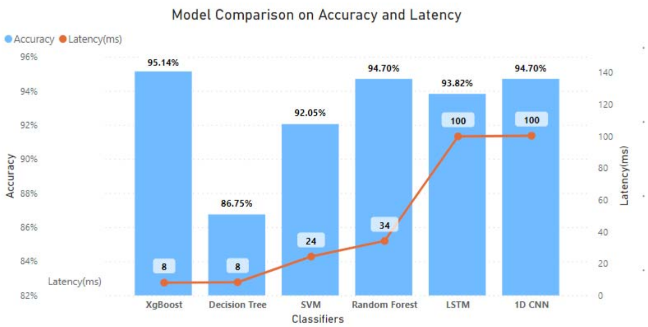


Fig. 6: Model Comparison - Accuracy vs Latency

integrated into mobile applications in the future. To integrate with mobile applications with less overhead, the model size should be smaller. From Fig.7, it can be seen that deep learning models have a larger size compared to machine learning models. It is because deep learning algorithms have more trainable parameters. Among the machine learning models, Decision Tree have the smallest size of 30KB but the least accuracy. The XGBoost classifier have a model size of 513 KB which is smaller when compared to LSTM (2374 KB), 1D CNN (2374 KB) and random Forest classifier (1726 KB). In terms of latency, accuracy, and size, the XGBoost classifier gave the optimal performance. Hence, this is an accurate, computationally efficient model suitable for detecting yoga poses in real-time.

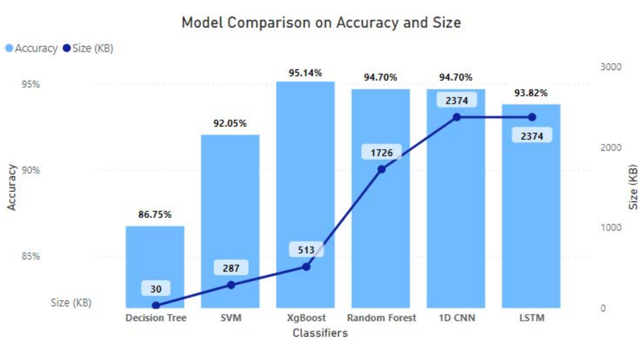


Fig. 7: Model Comparison - Accuracy vs Size

B. Performance of Real-time Machine Learning Framework

The aim of the final experiment was to detect how efficiently the yoga poses can be detected in real-time using the proposed framework. The user video was retrieved from the webcam using OpenCV. The landmark points were extracted using BlazePose model and then fed to the pretrained XGBoost Classifier. Real-time pose detection requires the user to be within 2 to 3 meters from the webcam and all landmark points from head to toe should be visible. A real-time display of the prediction results, along with probability and grade, was provided back to the user as feedback. Fig.8 shows a snapshot of real-time video detection when five different poses namely Tree Pose (see Fig. 8(a)),Warrior Pose (see Fig.8(b)), Plank Pose (see Fig.8(c)) ,Downdog Pose(see Fig 8(d)) ,Goddess Pose(see Fig.8(e)) were performed accurately. it is observed the framework was able to accurately detect all yoga poses in real-time with a probablity between 0.98 and 1. Also it could provide real-time feedback to the user with pose detection, score, and grade without any latency issues.

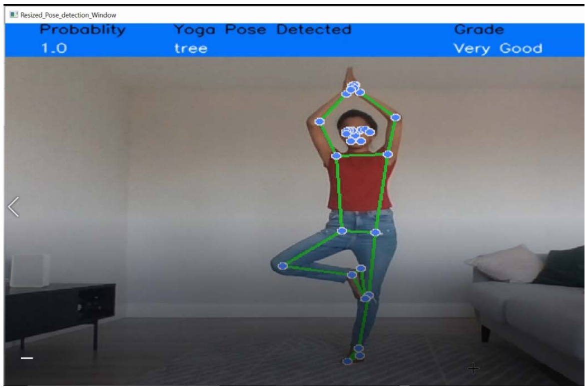
Researchers have previously reported better accuracy on test data with OpenPose than BlazePose model for detecting yoga poses [14], [15]. However, BlazePose have lower latency and lower computational cost. All experiments were conducted on a Jupyter notebook with an 8GB RAM CPU. Even with relatively smaller datasets, the proposed framework provides reasonable accuracy of 95.14% on test data. Generated skeletal images from BlazePose were converted to 3D landmark dataset in CSV format, making them suitable for machine learning. The skeletonisation of the image ensures that the model uses only the necessary features for detecting yoga poses.

One major challenge faced during the project was identifying the right dataset. Most of the publically available dataset was collected in a closed room with very few individuals. In real-time detection, these datasets might not perform well due to their lack of variety. Thus, a kaggle image dataset with five poses with different backgrounds was selected for this research .

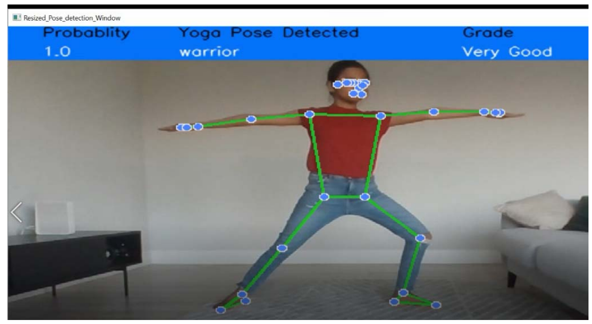
VII. Conclusion and Future work

The aim of this research was to investigate to what extent a real-time machine learning framework can be used for teaching an accurate yoga pose. The real-time framework proposed in this research combines a pose estimation model, pose classification model and a real-time feedback mechanism to detect five different yoga poses for unsupervised practice at home. The pose estimation model was based on the BlazePose model. The pose classification model was based on XGBoost. The real-time feedback mechanism was based on a comparative algorithm and OpenCV. As compared to OpenPose pose estimation models, BlazePose provides super real-time performance, low latency, and comparable accuracy in 3D pose estimation. Results demonstrate that BlazePose pose estimation model with XGBoost Classifier gives optimum results in terms of accuracy 95.14%, latency 8 ms, and size 513KB. On comparison with state of art approach, The proposed method gives high computational effectiveness with good accuracy. A limitation of this study is that it is not suitable for multi-person detections.

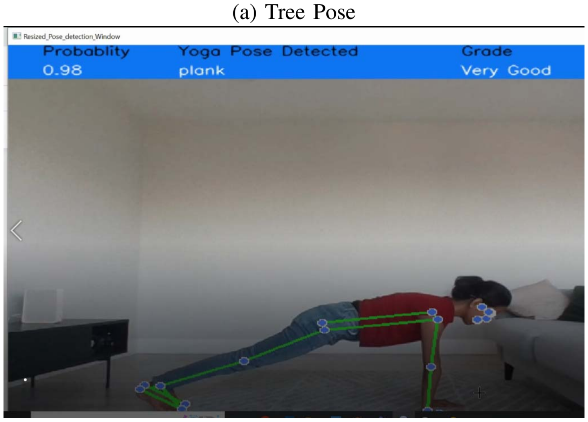
This research can potentially enhance real-time yoga learning applications as the proposed methodology can be integrated into mobile or web applications. Future research can be improved by considering more yoga poses with diverse backgrounds. Moreover, In the absence of enough publicly available datasets for yoga pose detection, future research can focus on creating a new dataset with more yoga poses. This research can also be extended to yoga action detection using a video dataset and sequential model for pose classification.



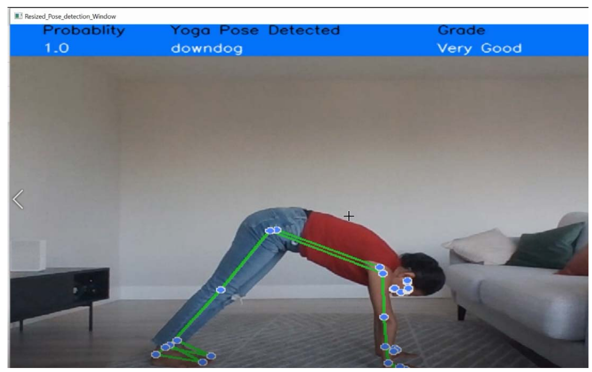
(a) Tree Pose



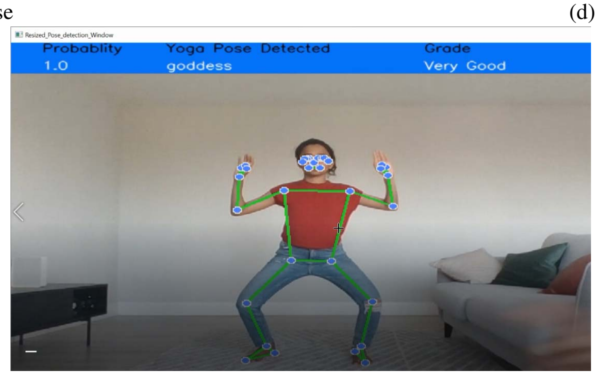
(b) Warrior Pose



(c) Plank Pose



(d) Downdog Pose



(e) Goddess Pose

Fig. 8: Real-time machine learning framework for yoga pose teaching system

In terms of pose estimation, the BlazePose model identifies only one person in an image frame. Further research should be conducted to detect multiple people for real-time yoga pose detection.

References

[1] P. S. Sahni, K. Singh, N. Sharma, and R. Garg, "Yoga an effective strategy for self-management of stress-related problems and wellbeing during covid19 lockdown: A cross-sectional study," *PloS one*, vol. 16, no. 2, p. e0245214, 2021.

[2] S. Sarkar, B. Sa, K. Singh, U. Gaur, A. Bharatha, V. Victor, S. Rahman, M. A. A. Majumder *et al.*, "Psychophysiological effects of yoga on stress management among medical and allied health professional students during covid-19 pandemic: A narrative review," *Advances in Human Biology*, vol. 11, no. 4, p. 3, 2021.

[3] G. S. Irani, "Injuries/harms resulting from incorrect adjustments/alignments performed by yoga asana practitioners." *Archives of Pharmacy Practice*, vol. 1, p. 38, 2020.

[4] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," 2020.

[5] S. Patil, A. Pawar, A. Peshave, A. N. Ansari, and A. Navada, "Yoga tutor visualization and analysis using surf algorithm," in *2011 IEEE Control and System Graduate Research Colloquium*, 2011, pp. 43–46.

[6] H.-T. Chen, Y.-Z. He, C.-C. Hsu, C.-L. Chou, S.-Y. Lee, and B.-S. P. Lin, "Yoga posture recognition for self-training," in *International Conference on Multimedia Modeling*. Springer, 2014, pp. 496–505.

[7] M. U. Islam, H. Mahmud, F. B. Ashraf, I. Hossain, and M. K. Hasan, "Yoga posture recognition by detecting human joint points in real time using microsoft kinect," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, pp. 668–673.
[8] E. W. Trejo and P. Yuan, "Recognition of yoga poses through an interactive system with kinect device," in *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 2018, pp. 1–5.
[9] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.
[10] X. Huang, D. Pan, Y. Huang, J. Deng, P. Zhu, P. Shi, R. Xu, Z. Qi, and J. He, "Intelligent yoga coaching system based on posture recognition," in *2021 International Conference on Culture-oriented Science & Technology (ICCST)*. IEEE, 2021, pp. 290–293.
[11] K. Yamao and R. Kubota, "Development of human pose recognition system by using raspberry pi and posenet model," in *2021 20th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2021, pp. 41–44.
[12] A. Chaudhari, O. Dalvi, O. Ramade, and D. Ambawade, "Yog-guru: Real-time yoga pose correction system using deep learning methods," in *2021 International Conference on Communication information and Computing Technology (ICCICT)*, 2021, pp. 1–6.
[13] F. Rishan, B. De Silva, S. Alawathugoda, S. Nijabdeen, L. Rupasinghe, and C. Liyanapathirana, "Infinity yoga tutor: Yoga posture detection and correction system," in *2020 5th International Conference on Information Technology Research (ICITR)*, 2020, pp. 1–6.
[14] S. Kothari, "Yoga pose classification using deep learning ph. d," Ph.D. dissertation, thesis SAN JOSE STATE UNIVERSITY, 2020.
[15] A. Gupta and A. Jangid, "Yoga pose detection and validation," in *2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA)*, 2021, pp. 319–324.
[16] J. Jose and S. Shailesh, "Yoga asana identification: a deep learning approach," in *IOP Conference Series: Materials Science and Engineering*, vol. 1110, no. 1. IOP Publishing, 2021, p. 012002.
[17] M. C. Thar, K. Z. N. Winn, and N. Funabiki, "A proposal of yoga pose assessment method using pose detection for self-learning," in *2019 International Conference on Advanced Information Technologies (ICAIT)*, 2019, pp. 137–142.
[18] A. Anilkumar, A. KT, S. Sajan, and S. KA, "Pose estimated yoga monitoring system," *Available at SSRN 3882498*, 2021.
[19] Y.-H. Lo, C.-C. Yang, H. Ho, and S.-W. Sun, "richyoga: An interactive yoga recognition system based on rich skeletal joints," in *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2021, pp. 256–257.
[20] S. Garg, A. Saxena, and R. Gupta, "Yoga pose classification: a cnn and mediapipe inspired deep learning approach for real-world application," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2022.