Real-Time Yoga Pose Detection using Machine Learning Algorithm

Research Project
MSc.Data Analytics

Jothika Sunney
Student ID: x20224532

School of Computing
National College of Ireland

Supervisor: Paul Stynes, Pramod Pathak, Musfira Jilani

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jothika Sunney
Student ID:	x20224532
Programme:	MSc.Data Analytics
Year:	2022
Module:	Research Project
Supervisor:	Paul Stynes, Pramod Pathak, Musfira Jilani
Submission Due Date:	15/08/2022
Project Title:	Real-Time Yoga Pose Detection using Machine Learning Algorithm
Word Count:	6362
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jothika Sunney
Date:	16th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Abstract

Yoga is an ancient art that provides physical and mental fitness. Yoga incorporates self-learning, but incorrect postures can cause serious muscle and ligament damage. During Covid-19, the importance of self-learning yoga practices has increased, and many people include yoga as part of their routines. A yoga pose detection system based on human pose estimation techniques and Machine Learning can assist people in practicing yoga correctly by themselves. The major challenge with current yoga pose detection methods is that most of them are computationally expensive and unsuitable for real-time applications. This research proposes a computationally inexpensive approach for real-time yoga pose detection by combining the Mediapipe Framework and Classification algorithms. An artificial intelligence-based system was built based on Mediapipe's Blazepose model and XgBoost Classifier to predict yoga postures in real-time. A publically available dataset of Five Yoga poses was analyzed in this study (down-dog pose, goddess pose, tree pose, plank pose, and warrior pose). In this Research, 3D landmark features in x,y,z directions were extracted from the dataset using the Blazepose model and then classified by four machine learning classifiers - Random Forest, Support Vector Machine, XgBoost, Decision Tree and two neural network classifiers - LSTM (Long Short-Term Memory) and 1D CNN (Convolutional Neural Network). All models were evaluated based on performance metrics. In order to detect yoga poses in real-time, XgBoost classifier was determined to be the optimum model, with an accuracy of 95.14%, precision of 95.36%, Recall of 95.02% and F1 Score of 95.17%. The proposed model is computationally efficient with an optimum latency of 8ms and size of 513KB. The Framework has the potential to be integrated into mobile application which can be used by yoga practitioners to perform yoga in the comfort of their homes.

1 Introduction

The practice of yoga combines mind, body, soul, and consciousness, bringing them all into harmony(Baptiste; 2022). Yoga lessons have become increasingly popular during the second wave of COVID-19. Research has shown that yoga boosts immune function and can prevent Coronavirus(Nagarathna et al.; 2020). Traditionally, yoga is practiced under the guidance of a trained educator. It is, however, not affordable for most people. Increasing awareness of yoga practices and the restrictions of Covid19 have led to people practicing yoga at home by themselves. Since the practitioner is not instructed by a professional, there is a high risk of performing the pose incorrectly. Several steps and guidelines must be followed by the practitioner to maximize the benefits of each pose. Failure to do so can often result in serious consequences. Continuing this incorrect practice for a long time can ultimately cause joint pain and many other health problems.

Yoga poses can be detected using AI systems that use human pose estimation techniques. [(Kothari; 2020), (Huang et al.; 2021a), (Chaudhari et al.; 2021a), (Long et al.; 2021)]. The human pose estimation subfield of deep learning uses skeletal representations to identify various human body elements. A few applications of pose estimation include sign recognition, sports analytics, animations, as well as gesture control. real-time yoga pose detection is challenging to estimate due to their variety, degrees of freedom, occlusions, and variations in appearance. Furthermore, most of the existing systems require a great deal of computing resources and are not suitable for real-time applications.[1]. This

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

study develops a cost-effective yoga pose detection model combining Mediapipe Blazepose and XgBoost Classifier. The study utilized 1551 images comprising five different yoga classes from a publicly available Kaggle dataset.[2]. There were five poses examined in the study: down-dog pose, goddess pose, tree pose, plank pose, and warrior pose.

The Mediapipe framework is an open source cross-platform solution for media processing. There are several ML solutions provided by Mediapipe, such as Face Detection, Iris tracking, Hand detection, Object detection, etc. BlazePose is Mediapipe's Human Pose estimation solution. It has a lightweight architecture based on a convolutional neural network for real-time human pose estimation. Because BlazePose can accurately locate more key points than other pose estimation models, it is uniquely suited for fitness applications. Additionally, the Mediapipe BlazePose model offers superreal-time performance(Bazarevsky et al.; 2020). The proposed model utilizes the Mediapipe BlazePose model to detect features in the form of 33 landmark points in x,y,z directions, and these landmark points are then used by classifiers to detect yoga poses. There hasn't been much work done combining Mediapipe and Machine learning models for yoga pose detection. Most of the existing system uses computationally expensive models like Openpose, PoseNet, AlphaPose models for human pose estimation[(Chaudhari et al.; 2021b),(Huang et al.; 2021b),(Yamao and Kubota; 2021)]. These techniques are computationally expensive and cannot be implemented into real-time applications. The limitations of the current methodologies and the issues associated with self-training yoga practices are the main motivation to develop a model that accurately predicts yoga poses in real-time.

The aim of this research is to investigate to what extent Mediapipe's Blazepose model and Machine Learning algorithms can correctly detect Yoga Pose. In the proposed system a real-time yoga pose detection framework is developed to correctly identify five different yoga poses.

A major contribution of this research is the development of a novel end-to-end framework that combines Mediapipe's Blazepose model, Xgboost classification algorithm, and computer vision methodologies to detect five different yoga poses for unsupervised practice at home. A real-time, accurate and low latency model is proposed for yoga practitioners. real-time feedback is displayed to the user with prediction probability, yoga pose detected, and grade achieved. The Feedback display is based on certain threshold conditions that will be explained in section 4. This research compares Xgboost, Random Forest, Support Vector Machine, Decision Tree, LSTM, and 1D CNN models in order to identify the most effective model for classification. A model is selected for Yoga pose detection based on performance metrics such as accuracy, loss, precision, recall, latency and size. Additionally,This study compares and evaluates the effectiveness of deep learning and machine learning models with the Mediapipe Blazepose Model.

The paper is structured as follows. A literature review of various yoga classification systems is presented in section 2. Section 3 describes the research methodology. Design specifications and implementation are described in sections 4 and 5. The Evaluation section evaluates the experiments conducted based on the results. It is then followed by the Discussion section. In section 7, the research is concluded and future work is discussed.

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

2 Related Work

Yoga pose detection has emerged as an important field of research under human pose estimation. Even before Deep Neural Network architecture and pose estimation frameworks, there had been efforts to create automated and semi-automated systems to analyze exercise and sports activities. (Patil et al.; 2011) uses the Surf algorithm to detect the correct yoga pose based on the video provided by the professional. Surf is a robust image detector and descriptor that is used for transforming images. This method is inaccurate as it considers only contour information for prediction. Some researchers (Chen et al.; 2014), (Islam et al.; 2017), (Trejo and Yuan; 2018) have used Kinect (depth sensors) based yoga pose detection approaches. Although this method gave good accuracy, Depth sensor-based cameras are expensive and are not accessible to ordinary users.

Deep neural networks have revolutionized human pose estimation from the traditional systems. Deep Neural network architectures are used to develop various pose estimation models which improves performance and reduces cost of human pose estimation. A widely used open source pose estimation model, OpenPose, was proposed in (Cao et al.; 2017).The OpenPose model was used by (Chaudhari et al.; 2021a) to extract 15 key points from the yoga pose and the extracted key points were passed on to a CNN classifier for prediction. Using the reference key points, the angles of the pose are calculated and the error is measured. (Rishan et al.; 2020) proposes another method that combines Openpose with MASK RNN and achieves 99.91% accuracy. Even though the Mask RCNN model performed well on test data, more false positives and true negatives were observed when implemented on real-time data. With openpose and CNN-LSTM combined model, (Kothari; 2020) achieves an accuracy of 99.38% on yoga pose videos. Considering that the dataset was generated in a closed environment with a defined group of people, the model accuracy might not be consistent. GPUs are also required for the execution of the model. A system that extracts keypoints using the Lighter OpenPose was proposed by Gupta and Jangid (Gupta and Jangid; 2021). While this method produces impressive results with high-resolution images, it performs poorly with low-resolution images and occlusions. Most Openpose based methodologies produce good accuracy but it require high computational power and cannot be implemented in real-time.

(Thar et al.; 2019) uses openpose algorithm to detect landmark points and calculates the angles between keypoints to detect particular yoga poses. As a result of this method, each yoga pose's angles must be identified beforehand, which seems difficult to accomplish for a wide variety of poses. (Verma et al.; 2020) introduced Yoga-82, a large-scale hierarchical dataset for yoga posture recognition. A DenseNet201 on openpose model was used to classify yoga poses hierarchically. In the 82 yoga dataset, most of the classes are very similar, so more training images are required for each class to predict yoga poses accurately. In (Agrawal et al.; 2020), authors created 5500 images of 10 yoga poses, used the TF pose estimation algorithm and a random forest classifier for yoga pose detection. The model produced an accuracy of 99.04%. The dataset was collected in a closed room with white background. This may cause the model to perform poorly on unseen data. As TF pose estimation identifies only 13 landmark points, additional angular calculations were used to extract features. (Jose and Shailesh; 2021) used CNN and transfer learning methodologies for yoga pose prediction. There were only 700 images for 10 different poses. The limited dataset reports poor results(85% accuracy). In addition, the model was complex, requires more resources, and could not be used for real-time prediction.

A pose estimation solution from Mediapipe Framework is known as BlazePose. The work (Bazarevsky et al.; 2020) proposes BlazePose, an architecture for human pose estimation using lightweight convolutional neural networks suitable for real-time inference on mobile devices. In a single person, the network produces 33 body keypoints 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.

Mediapipe BlazePose model and angular metrics calculations are used by (Anilkumar et al.; 2021) to develop a yoga posture correction system. By using Mediapipe, 33 key points from the yoga posture are extracted, and angles between the joints are calculated using geometric analysis of landmark data. Although this method is effective for 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 in the paper(Lo et al.; 2021). An LSTM neural network combined with Mediapipe's Holistic model is used to classify five yoga poses with an accuracy of 85%. (Jagtap et al.; n.d.) uses Medipipe's holistic model along with logistic regression to predict yoga pose. No detailed information about the test accuracy or dataset has been provided in the study. These methodologies use a holistic model which detects 543 landmarks, but it includes unnecessary face and palm landmarks that are not required for yoga pose detection. The BlazePose model with 33 landmark points is better suited to detecting yoga poses.In (Garg et al.; 2022), a model called yogaConvo2d was presented combining Blazepose and CNN algorithms. Instead of obtaining landmark points from the skeletal images, the skeleton image from Blazepose is directly passed to a 2D CNN. Using skeletal images instead of landmarks may add complexity to a model and may require more time for training. While this method achieved 99.62% accuracy, the research did not analyze its performance on real-time prediction. A Kaggle dataset with 1551 images of yoga poses under five different classes was used to build the model. This dataset was selected to implement the proposed real-time yoga pose detection.

In Conclusion, One of the main challenges of existing yoga pose detection algorithms is the computational cost. Although OpenPose has been well established for pose estimation, it is computationally expensive and requires a GPU for high performance. Depth sensors are 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 study 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 Mediapipe Blazepose model outperforms existing pose estimation frameworks for real-time Yoga/Fitness applications. On a 20-core desktop CPU, BlazePose performs 25–75 times faster than OpenPose. Furthermore, Mediapipe models produce keypoints in the x, y, and z dimensions. Here z corresponds to the distance of the user from the camera. Thus, Mediapipe provides 3D feature extraction without requiring depth sensors. There has not been much research combining Mediapipe and Machine learning algorithms. A novel approach is proposed in this study by combining Mediapipe Blazepose model with XgBoost classifier for real-time yoga pose detection. Additionally, the study evaluates the performance of deep learning and machine learning models on 3D Landmark data generated by Blazepose model.

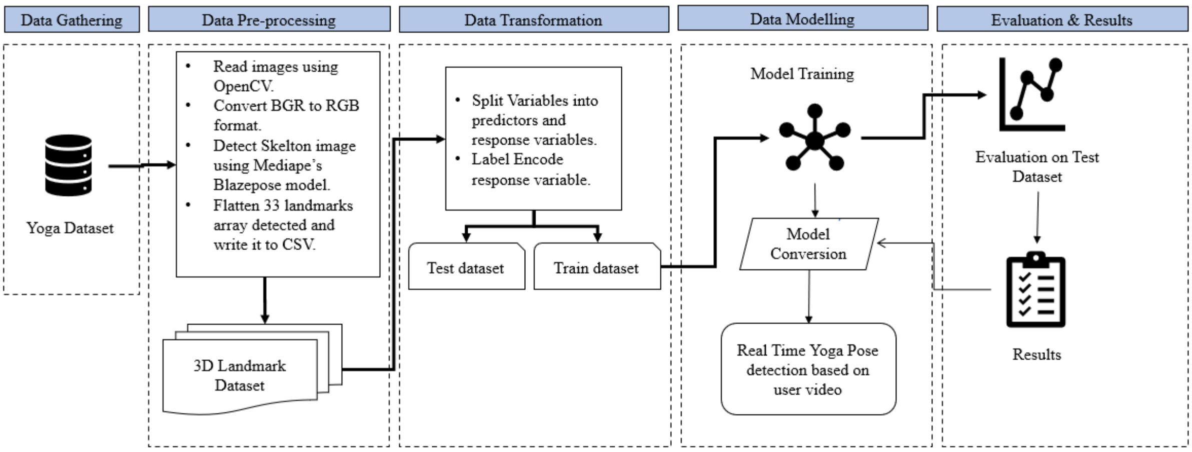


Figure 1: Methodology - Real-time Yoga Pose Detection

3 Methodology

Figure.1 shows Research Methodology followed in this study. It consists of five different steps - Data Gathering, Data Preprocessing , Data Transformation,Data Modelling and Evaluation.

The first step, Data Gathering involves identifying the yoga pose dataset. A publicly available yoga image dataset from Kaggle was used in this project to classify 5 commonly used yoga poses.[3]. The yoga poses examined in this study were downdog(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, consists of different individuals with diverse background. Yoga poses were organized into folders with names corresponding to the respective yoga pose.

The Data preprocessing step involves converting the yoga pose image dataset into Skelton images and then feature extracting 33 3D landmark points in x,y,z directions using Mediapipe's Blazepose model and OpenCV library. With BlazePose, a single image frame provides 33 3D landmarks. The 33 keypoints detected by the Mediapipe's Blazepose model are shown in Figure 2.

Each yoga pose is determined by the x, y, and z coordinates of the 33 points. As compared to current pose estimation models such as openpose, deeppose, etc, Blazepose is specially suited to fitness applications since it locates more key points with greater accuracy. Furthermore, current methods rely on powerful desktop environments to perform yoga pose detection, while this method performs real-time on CPU inference. Firstly the image is read using the OpenCV library. OpenCV reads data in BGR format, but Mediapipe requires RGB input. In the first step of pre-processing, the image was converted from BGR to RGB. In order to detect the skeleton image, the image was passed to the Pose.process() function of the Mediapipe Blazepose object. The 33 landmark points were extracted from the skeleton image and appended to a CSV file. Additionally, the folder names indicating the yoga pose class name were appended to the CSV file as the response variable. In the generated dataset, the independent variables were standardized landmarks in x, y, and z directions. The variables were defined as x1, y1, z1...x33, y33,

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

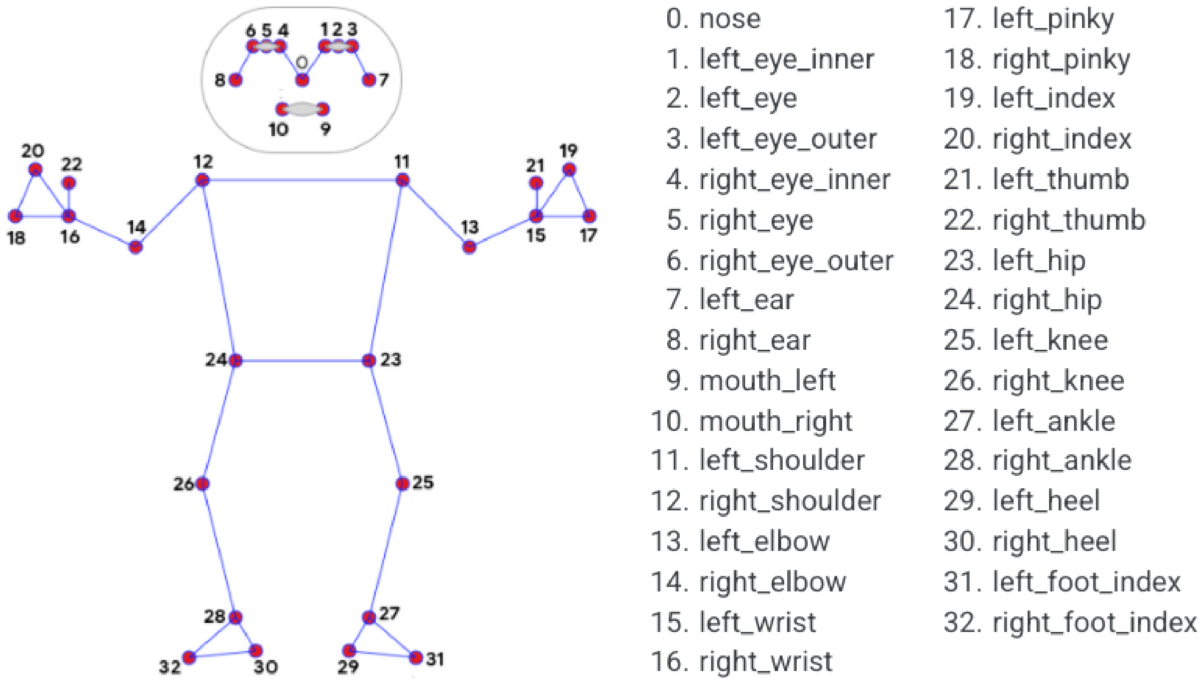


Figure 2: BlazePose Model - Topology as depicted in the orginal paper(Bazarevsky et al.; 2020)

z33, and added as headers to the CSV file. Refer Figure.3 which depicts the 3D landmark data generation using Blazepose model.

Figure 3: 3D Landmark data generation on warrior pose using Blazepose model

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

During the data modeling step, the models were trained using the training dataset. This research trains and evaluates four machine learning models (Random Forest, Xg-Boost, 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 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,precion,recall and time complexity. Models were applied to test data and their performance was assessed using confusion metrics and confusion report. Accuracy and loss plots were analysed for deep learning models. Based on the experimentation, XgBoost Classifier was determined to be the most suitable model for real-time pose prediction in yoga. A real-time yoga pose detection Framework was developed combining Blazepose , XGBoost Classifier and Computer vision methodologies which provides a real-time feedback to the user.

4 Design

The purpose of this research is to improve yoga pose classification using Mediapipe Blazepose model and classification algorithms. Among the different Machine Learning and Deep Learning models evaluated, XgBoost Classifier yields the best results regarding accuracy, latency, and size. Therefore, it is used for the final model. The real-time Yoga Pose detection architecture framework includes Data Capturing Module, Blazepose model, Machine Learning model, and Feedback modules. Figure.4 shows the System Architecture for real-time Yoga Pose Detection.

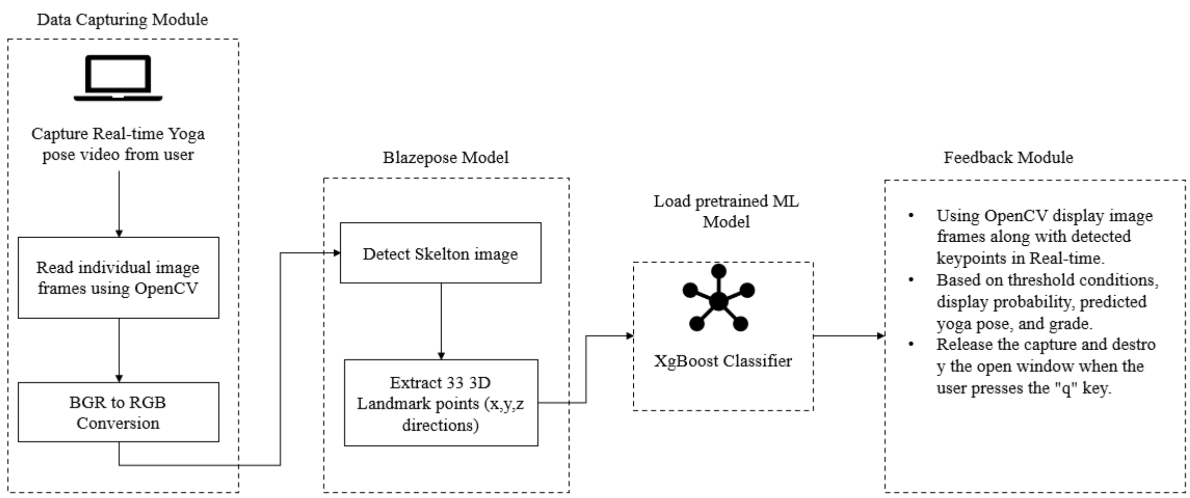


Figure 4: System Architecture - real-time Yoga Pose Detection

Data Capturing Module captures live video from the user webcam using OpenCV's VideoCapture() function. This function allows for 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 Mediapipe's 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 keypoints from this ROI. It uses only the first frame of the video to detect key points. In subsequent frames, ROIs are extracted from keypoints in the previous frame. The detector is triggered only if the ROI cannot be detected(Bazarevsky et al.; 2020). This method improves the performance of BlazePose, making it suitable for real-time applications. Blazepose was used to extract 33 3D landmark points in x,y,z directions for each image frame; it was then flattened, converted into a dataframe, and inputted into the XgBoost Classification model for prediction. Using the XgBoost classifier, the yoga pose is identified for each image frame. Real-time feedback is then displayed to the user based on the output of the classifier.

As part of the Feedback Module, a real-time display of the detected yoga pose, the probability of pose classification, and grade are provided as feedback. 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. It will display ”No Pose Detected” for all probabilities below 0.85. A dialogue box with feedback, image frames, and detected landmarks is displayed using the OpenCV library. The ”q” key is used to release the video capture and close all opened windows.

5 Implementation

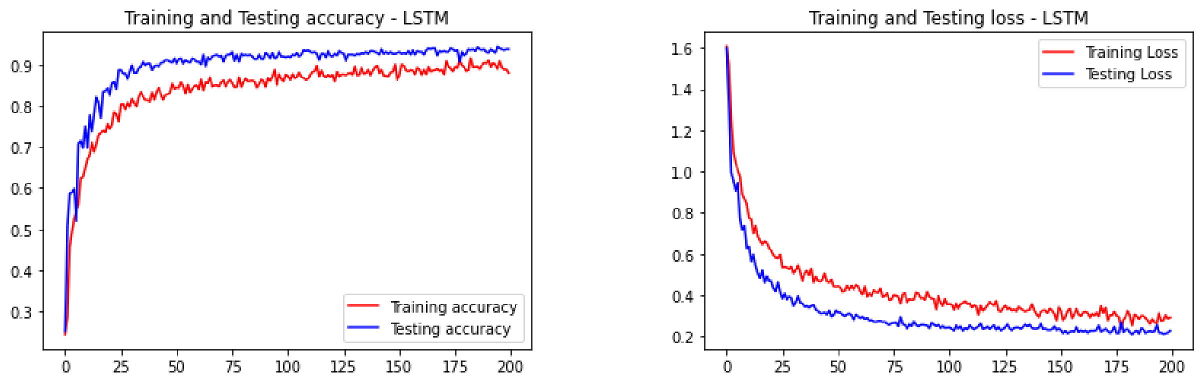
The real-time yoga pose detection was implemented with an XgBoost Classifier and Blazepose model from Mediapipe framework. Jupyter Notebook with Python 3.8.8 was used for implementing data preprocessing, data transformation, model implementation, and evaluation. The main libraries used for the research 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. Using Mediapipe’s Blazepose pose estimation model (Mediapipe.solutions.Pose()), the preprocessed image frame was converted into a skeleton image with 33 3D landmark points. By using pose_landmarks() function, the Pose estimation model extracts x,y,z 3D components for each landmark point. A CSV file was created by flattening the landmark points, which were in the form of a 3D array. Model implementation was then carried out using the generated CSV dataset. Dataset was then divided into 70:30 ratios for training and testing, respectively. The Sklearn library was used to evaluate machine learning models, SVM classifier, Random Forest algorithm, Decision Tree and XgBoost algorithm. In order to select the best-fit parameters for the machine learning models, hyperparameter optimization along with cross-validation was applied. A 1D CNN and a LSTM model were also applied in order to evaluate the performance of deep neural networks on generated 3D landmark dataset. Keras with Tensorflow backend was used for implementing deep learning models. Based on the accuracy, precision, recall, time complexity and model complexity, XgBoost classifier was selected 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 are carried out on a single CPU with 8GB RAM.

6 Evaluation

This section discusses the details of the experiments conducted in this study. The goal of the study was to improve real-time yoga pose classification utilizing Mediapipe's Blazepose model and Machine Learnig algorithm. The performance of different Machine Learning algorithms and Deep Learning algorithms was compared and analysed on 3D landmark dataset generated by the Blazepose model.

Experiment 1: The first experiment aims to replicate the state-of-the-art approach presented in (Lo et al.; 2021). This paper uses Mediapipe's Holistic model and LSTM Deep neural network to detect yoga poses. For four different poses, the authors used their own video dataset and obtained an accuracy of 85%. The dataset was private and no information was provided regarding the number of LSTM layers. Due to this, it was not possible to exactly replicate the state-of-the-art approach. For this experiment, a publically available image dataset with five different poses was used, and output of Mediapipe's Blazepose model was trained using modified LSTM layers. The modified state of art approach gave an improved accuracy of 93.82%.

First, the yoga pose image dataset was converted into 33 3D landmark points (x, y, z) using Mediapipe's Blazepose model. As a next step, the generated dataset is split into 70:30 ratio before being applied to the LSTM model. The LSTM model consists of three LSTM layers with 64, 128 and 64 units, respectively. It is followed by two dense layers with 64 and 32 neurons respectively and one output layer with 5 neurons. The output layer used the softmax activation function, while the other layers used the relu activation function. To avoid overfitting, 20% dropout has been added to all LSTM layers and the model was run for 200 epochs for optimum output. On the test dataset, the model achieved 93.82% accuracy with 0.2279 loss. The accuracy and loss plot of the LSTM model is shown in Figure.5.

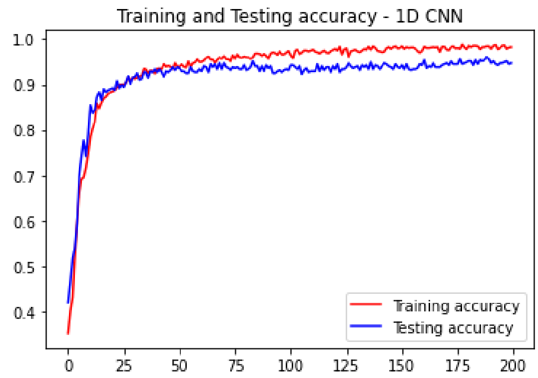


(a) LSTM - Accuracy Plot

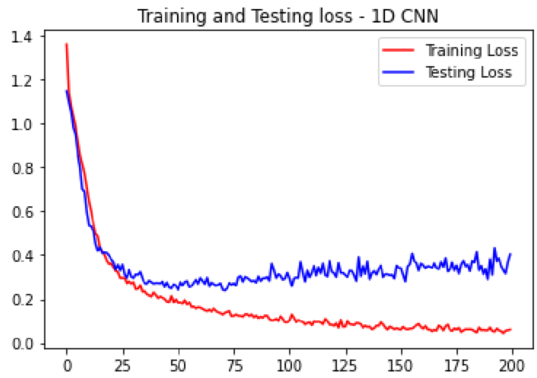
(b) LSTM - Loss plot

Figure 5: LSTM - Accuracy-Loss plot

Experiment 2: The second experiment is to evaluate the performance of 1D CNN on the landmark dataset. This experiment is to check the performance of a less complex neural network with the Blazepose model. Studies have shown that a simple convolutional architecture can outperform the LSTM network[(Bai et al.; 2018),(Zhang et al.; 2015)]. The Implemented 1D CNN network consists of a Conv1D layer with 64 filters and kernel_size of 3. It was followed by a dropout layer with 0.5 dropout rate and a Maxpooling 1D layer with pool_size as 1. This was followed by another dropout layer with 0.5 dropout rate and two dense layers with 64 and 8 neurons respectively. The final layer is an output layer with 5 neurons and a 'softmax' activation function. The hidden layers are activated using the 'Relu' function. The architecture of 1D CNN is shown in fig.1 . This model gives an accuracy of 94.7% and a loss value of 0.4023 on the test dataset. The accuracy and loss plots of the models are shown in Figure.6.



(a) 1D CNN - Accuracy Plot



(b) 1D CNN - Loss plot

Figure 6: 1D CNN - Accuracy-Loss plot

Experiment 3: In the third experiment, the performance of four different machine learning models was evaluated. XgBoost, Random Forest, Decision Tree and Support Vector Machine are the three models considered in this experiment. For each model, hyperparameter tuning with cross validation was used to select the best fit parameters. Table 1 shows the accuracy, precision, recall, and f1 score for all four models. It can be observed from the table that XgBoost gives the best results and the results are comparable to those of deep learning models. Additionally, machine learning models are faster and less complicated than deep learning models. For real-time yoga pose detection, the XgBoost classifier was selected as the optimum model. The confusion metrics of XgBoost classifier is shown in Figure.7.

Classifier	**Accuracy**	**precision**	**Recall**	**F1 Score**
XgBoost	95.14%	95.36%	95.02%	95.17%
Random Forest	94.7%	95.22%	94.41%	94.75%
SVM	92.05%	91.89%	92.27%	91.95%
Decision Tree	86.75%	86.42%	87.15%	86.58%

Table 1: Results From Experiment 3

Experiment 4: In the final experiment, yoga poses were detected in real-time. The user video was retrieved from the webcam using the OpenCV VideoCapture object. The landmark points are extracted using Blazepose model and then fed to the pretrained XgBoost Classifier.As part of the experiment, I performed various yoga poses in front of a webcam. real-time pose detection requires the user to be within 2 to 3 meters from the webcam and all key points from head to toe should be visible. Figure.8 shows a snapshot of real-time video detection when five different poses were performed accurately.

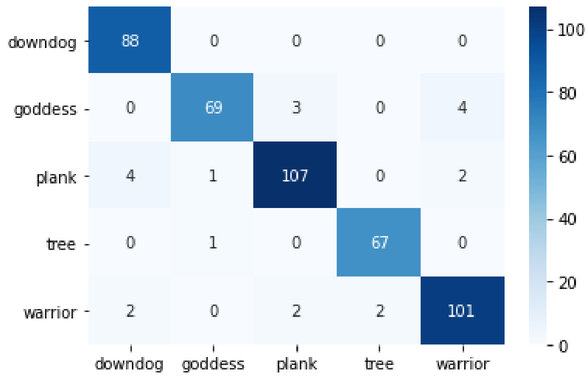
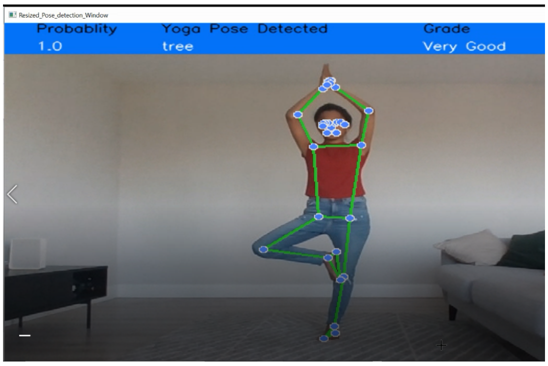
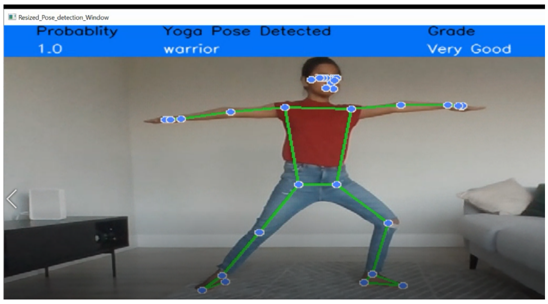


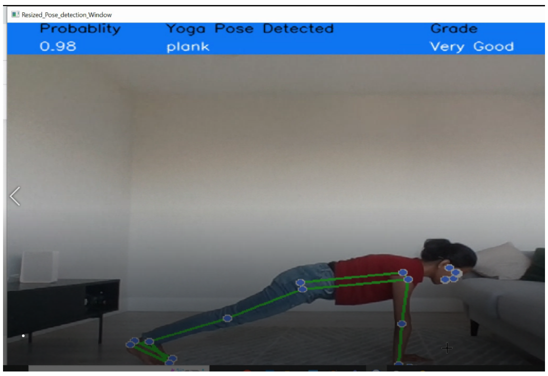
Figure 7: Confusion matrix - XgBoost Classifier



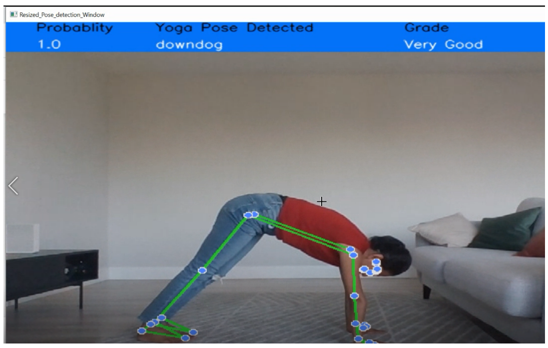
(a) Tree Pose



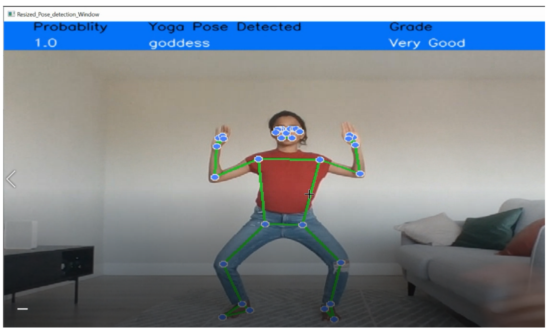
(b) Warrior Pose



(c) Plank Pose



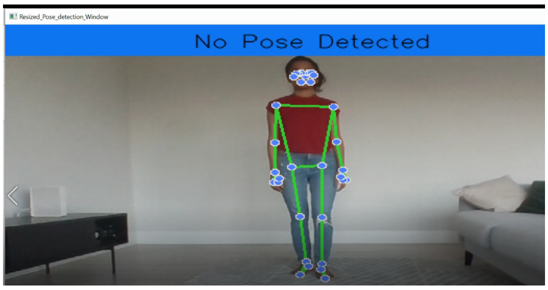
(d) Downdog Pose



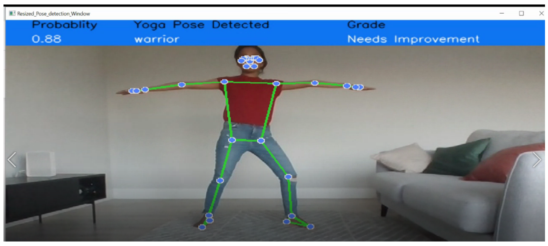
(e) Goddess Pose

Figure 8: Real-Time Yoga Pose Detection

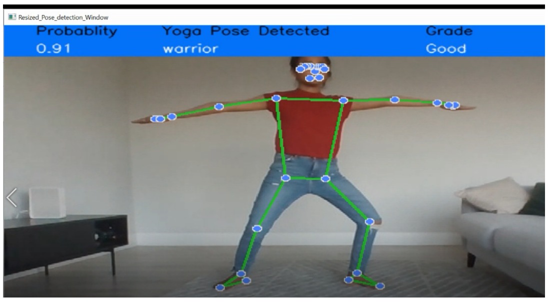
A real-time display of the prediction results, along with probability and grade, was provided to the user. Based on the predicted probability, three grades were calculated: "very good" (probablity greater or equal to 0.95), "good" (probablity greater or equal to 0.90 and less than 0.95), and "needs improvement" (probablity greater than or equal to 0.85 and less than 0.90). The user will be displayed "No Pose detected" if the probability is below 0.85. Refer to Figure.9 for different grades displayed to the user based on performance of warrior pose.



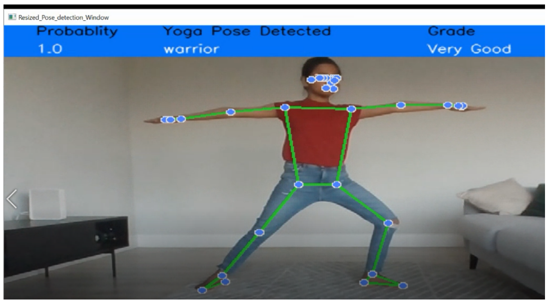
(a) No Pose Detected



(b) "Needs Improvement"



(c) "Good"



(d) Very Good

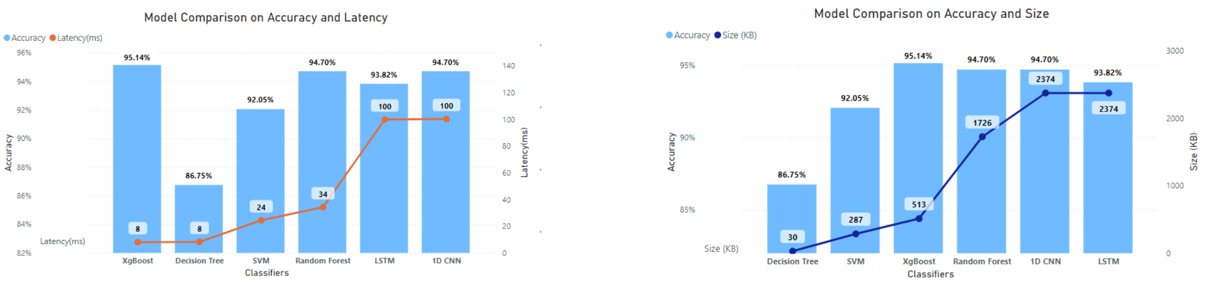
Figure 9: Grade feedback to user based on performance

7 Discussion

This section discusses the implications and interpretations of the experiments conducted. All the models were evaluated based on accuracy, latency, and size, and the optimum model was selected for real-time yoga pose detection. According to the analysis, the XgBoost classifier gave the highest accuracy of 95.14%, while the SVM and Random Forest models gave comparable results to deep learning models.

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. Latency was measured in milliseconds. Figure.10a shows that all the models had less latency within 100 ms. Compared to deep learning models, machine learning models had a lower latency. XgBoost Classifier and Decision Tree had the lowest latency of 8 ms, whereas LSTM and 1D CNN had the highest latency of 100 ms. Random Forest and Support Vector had a latency of 34 ms and 24 ms, respectively. As the Decision Tree had the least accuracy, it could 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 integrated into mobile applications in the future. To integrate with mobile applications with less overhead, the model size should be smaller. An increase in size can result in an increase in power consumption. and adverse performance for mobile devices. From Figure.10b, 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. But when compared to previous research using Mediapipe and 2D CNN on the Skeletal dataset (Garg et al.; 2022), the 1D CNN tested on the landmark dataset required relatively fewer trainable parameters. Among the machine learning models, Decision Tree had the smallest size of 30KB but the least accuracy. The XgBoost classifier had a model size of 513KB which was smaller when compared to LSTM (2374KB), 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.



(a) Accuracy vs latency

(b) Accuracy vs Size

Figure 10: Model Comparison

Researchers have previously reported better accuracy with other pose estimation models than Blazepose model for detecting yoga poses [(Chaudhari et al.; 2021a), (Agrawal et al.; 2020), (Kothari; 2020)]. However, Mediapipe's Blazepose has lower latency and higher computational efficiency. All experiments were conducted on a Jupyter notebook with an 8GB RAM CPU. Even with relatively smaller datasets, the mediapipe framework provides reasonable accuracy. 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. The final experiment implements a framework for real-time yoga pose detection using XgBoost Classifier and Blazepose model. From Figure.8 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.

One major challenge faced during the project was identifying the right dataset. Most of the publically available video 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 was selected for implementation. This dataset contains different individuals with variety backgrounds. real-time feedback on how to correct the pose would have been helpful if Angular calculations had also been added.

8 Conclusion and Future work

In conclusion, This research aimed to understand how effectively real-time yoga pose detection can be done by combining the mediapipe Blazepose model and Machine Learning algorithms. A comparison of deep learning and machine learning models was performed in this research. Deep learning models LSTM and 1D CNN were found to have accuracy of 93.82%, 94.70%, respectively. whereas Machine learning models XgBoost, Random Forest, SVM, and Decision Tree gave an accuracy of 95.14% 94.70%, 92.05%, 86.75% respectively. The study uses a publicly available Kaggle dataset with five yoga poses(Tree, warrior, Goddess, Plank and Downdog). Based on the findings of this study, it was concluded that real-time yoga pose detection can be achieved using a Machine learning model along with the Blazepose pose estimation model.

In this research, Blazepose, a pre-trained model under the Mediapipe framework, extracted 33 3D landmark points from the yoga pose frame and passed them to the XgBoost classifier for real-time prediction. This research proposes a less complex methodology for real-time yoga pose detection which can be easily integrated into mobile or web applications. Mediapipe framework allows the model to be implemented in an 8GB CPU with minimal computational resources. As compared to previous approaches utilizing Openpose pose estimation models, it provides super high real-time performance, low latency, and comparable accuracy in 3D pose estimation. This model has the limitation of not being suitable for multi-person detections.

Potentially, this research could enhance real-time yoga learning applications. Future research can develop an end-to-end application for detecting yoga poses in real-time. Moreover, the research currently focuses on only five yoga poses, but more poses could have been included. In the absence of enough publicly available datasets for yoga pose detection, research could have focused on creating a new dataset with more poses. As well, rather than passing key points directly to the model, this work can be improved by considering the angle between key points. Also, Providing additional feedback on how to correct a yoga pose in real-time could improve real-time feedback. Currently, the Blazepose model identifies only one person in an image frame. Further research should be conducted to detect multiple people. Yoga action detection is also a potential field of research and a video dataset using different sequential models can be used to implement this.

References

Agrawal, Y., Shah, Y. and Sharma, A. (2020). Implementation of machine learning technique for identification of yoga poses, *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 40–43.

Anilkumar, A., KT, A., Sajan, S. and KA, S. (2021). Pose estimated yoga monitoring system, *Available at SSRN 3882498* .

Bai, S., Kolter, J. Z. and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, *arXiv preprint arXiv:1803.01271* .

Baptiste, B. (2022). *Journey into power: How to sculpt your ideal body, free your true self, and transform your life with yoga*, Simon and Schuster.

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

Cao, Z., Simon, T., Wei, S.-E. and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299.

Chaudhari, A., Dalvi, O., Ramade, O. and Ambawade, D. (2021a). Yog-guru: Real-time yoga pose correction system using deep learning methods, *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pp. 1–6.

Chaudhari, A., Dalvi, O., Ramade, O. and Ambawade, D. (2021b). Yog-guru: Real-time yoga pose correction system using deep learning methods, *2021 International Conference on Communication information and Computing Technology (ICCICT)*, IEEE, pp. 1–6.

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

Garg, S., Saxena, A. and Gupta, R. (2022). 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.

Gupta, A. and Jangid, A. (2021). Yoga pose detection and validation, *2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA)*, pp. 319–324.

Huang, X., Pan, D., Huang, Y., Deng, J., Zhu, P., Shi, P., Xu, R., Qi, Z. and He, J. (2021a). Intelligent yoga coaching system based on posture recognition, *2021 International Conference on Culture-oriented Science Technology (ICCST)*, pp. 290–293.

Huang, X., Pan, D., Huang, Y., Deng, J., Zhu, P., Shi, P., Xu, R., Qi, Z. and He, J. (2021b). Intelligent yoga coaching system based on posture recognition, *2021 International Conference on Culture-oriented Science & Technology (ICCST)*, IEEE, pp. 290–293.

Islam, M. U., Mahmud, H., Ashraf, F. B., Hossain, I. and Hasan, M. K. (2017). Yoga posture recognition by detecting human joint points in real time using microsoft kinect, *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 668–673.

Jagtap, R., Zanzane, M. and Patil, R. (n.d.). Yoga pose detection using machine learning.

Jose, J. and Shailesh, S. (2021). Yoga asana identification: a deep learning approach, *IOP Conference Series: Materials Science and Engineering*, Vol. 1110, IOP Publishing, p. 012002.

Kothari, S. (2020). *Yoga Pose Classification Using Deep Learning Ph. D*, PhD thesis, thesis SAN JOSE STATE UNIVERSITY.

Lo, Y.-H., Yang, C.-C., Ho, H. and Sun, S.-W. (2021). richyoga: An interactive yoga recognition system based on rich skeletal joints, *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, IEEE, pp. 256–257.

Long, C., Jo, E. and Nam, Y. (2021). Development of a yoga posture coaching system using an interactive display based on transfer learning, *The Journal of Supercomputing* pp. 1–16.

Nagarathna, R., Nagendra, H. and Majumdar, V. (2020). A perspective on yoga as a preventive strategy for coronavirus disease 2019, *International journal of yoga* **13**(2): 89.

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

Rishan, F., De Silva, B., Alawathugoda, S., Nijabdeen, S., Rupasinghe, L. and Liyanapathirana, C. (2020). Infinity yoga tutor: Yoga posture detection and correction system, *2020 5th International Conference on Information Technology Research (ICITR)*, pp. 1–6.

Thar, M. C., Winn, K. Z. N. and Funabiki, N. (2019). A proposal of yoga pose assessment method using pose detection for self-learning, *2019 International Conference on Advanced Information Technologies (ICAIT)*, pp. 137–142.

Trejo, E. W. and Yuan, P. (2018). Recognition of yoga poses through an interactive system with kinect device, *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*, IEEE, pp. 1–5.

Verma, M., Kumawat, S., Nakashima, Y. and Raman, S. (2020). Yoga-82: a new dataset for fine-grained classification of human poses, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 1038–1039.

Yamao, K. and Kubota, R. (2021). Development of human pose recognition system by using raspberry pi and posenet model, *2021 20th International Symposium on Communications and Information Technologies (ISCIT)*, IEEE, pp. 41–44.

Zhang, X., Zhao, J. and LeCun, Y. (2015). Character-level convolutional networks for text classification, *Advances in neural information processing systems* **28**.