Yoga pose classification from images using transfer learning approach

Deepali Handgar[1], Sumon Singh[2]

[1,2] *Department of Computer Science, University of Mumbai, Mumbai, Maharashtra, India*

Abstract-Yoga is a practice that originated in ancient India. *Hatha* Yoga is a type of physical activity that consists of postures (also called *'asanas'*) in continuous sequence along with regulated breathing. Yoga is helpful to balance our mind and body through meditation, exercise, and breathing work. Various studies show yoga's benefits for arthritis, mental health, women's health, and other specialties. For all these reasons, yoga has seen immense popularity throughout the world. It's important to correctly identify the asana a person needs to perform according to his/her needs. This paper uses a transfer learning approach to identify the yoga pose shown in each picture. A total of 1551 images of 5 different yoga postures were used and resized for ease of computing. 10 models were used for the classification of the images and their evaluation metrics were compared to see which model gave better results. The models used were VGG16, VGG19, InceptionV3, DenseNet201, ResNet50V2, ResNet152V2, ResNet101V2, MobileNet, MobileNetV2 and InceptionResNetV2. Out of all these models, VGG16 outperforms and the validation accuracy witnessed is 94.47% (with un-augmented data).

***Index Terms* - Deep learning, Image classification, Transfer learning, Yoga pose classification**

I. INTRODUCTION

Yoga, in essence, is a spiritual discipline that brings harmony between the mind and the body. The word "Yoga" originates from its Sanskrit root *'yuj'* which means 'to unite' [2]. Yoga has multiple benefits for not just the mind but the body as well. Yoga asanas help with arthritis, insomnia, stomach problems, PCOD, weight loss, and many more problems [3].

Because of the hectic modern lifestyle, people suffer from various kinds of body pain and mental stress. These problems can lead to serious chronic illnesses if they are not taken care of at the earlier stages. Yoga can be considered to be one of the most effective solutions to this problem.

For yoga to be effective, it needs to be learned with the help of certified trainers. Because, if not performed correctly, it can cause sprains or muscle contractions. However, with the modern lifestyle, one has many constraints, and learning under supervision or with the help of trainers may not be possible. Hence, people turn to online measures and seek support and guidance from the internet. Applications or websites are being made to help people learn yoga. There are various opportunities in this field for further study and research. Various papers are published, where research is done not only to increase the accuracy of the model to identify the yoga poses but also to identify if the posture is correct or not [4].

In this paper, we have used images of five yoga asanas, viz. The downward dog pose, the goddess pose, the plank, the tree, and the warrior pose. Various pre-trained deep-learning CNN models were used to classify the yoga pose in each image. Their results are compared and analysed in this paper.

II. LITERATURE REVIEW

Image-classification is a popular topic for computer vision. We have reviewed a few papers about the classification of images of yoga poses and pose estimation using deep learning.

The paper by Debabrata Swain et al [5] used a combination of CNN and long-short term memory on real-time monitored videos to recognize yoga poses. In their paper, the CNN layer was used to extract features, and the LSTM layer was used to understand the occurrence of the sequence of frames.

S. Abarna et al [6] used 1D CNN and RNN for classification and feature selection, OpenPose and CenterNet pre-trained models were used. The experiment gave better results when they used OpenPose with 1D CNN as compared to other models.

Josvin Jose et al [7] have created a system that can recognize a yoga pose or asana from an image or a video. They used the deep learning technique of Convolutional-Neural- Network (CNN) and transfer learning. Of all the models used, the VGG16 pre-trained model gave the highest accuracy of 85% on their test data.

C. Long et al [4] collected the data by asking participants to pose in each yoga asana 10 times. The pictures were taken with an RGB camera and then used as data. They used a total of six transfer learning models for classification. The MobileNet-DA model was the best-performing model with an overall accuracy of 98.43%. S. Gupta et al [12] made use of media pipe library along with various machine learning algorithms for classification to detect the yoga poses from videos of yoga poses done by volunteers. The system got an accuracy of 94%. S. K. Yadav et al [13] proposed an end-to-end deep learning pipeline to detect yoga poses from videos. They used CNN, LSTM and Openpose to achieve an accuracy of 99.04% on single frames.

J. Kutalek [14] created a project to recognize and classify yoga poses from video frames using simple CNN model. The data used consisted of video frames from 162 videos. A total of 22000 images of 22 yoga poses were used and OpenCV library was used to capture the frames. The model achieved an accuracy of 91%. U. Bahukhandi et al [15] also used machine learning techniques to detect and classify yoga poses. The authors first extracted the data points from videos using the media pipe library and then performed the pre-processing, training and testing on the data using the machine learning methods such as Logistic Regression, SVM classifier, Random Forest Classifier, KNN classifier and Naïve Bayes classifier. Of these, Logistic regression achieved the highest accuracy of 94%. Random forest classifier got the least accuracy of 89%. On the other hand, Y. Agarwal et al [16] observed that for their work, Random Forest Classifier got highest accuracy of 99.04%. They also used various machine learning models on a dataset of 5500 images with ten different yoga poses. Tf-pose estimation library was used to extract the angles of the joints in the human body.

The field of yoga in technology has not been extensively explored yet and is slowly making progress. This topic of research is somewhat new in this field, wherewith transfer learning approach we are using a few pre-trained models and comparing their performance to see how accurately the models classify the data.

III. METHODOLOGY

A] Dataset

The dataset used for this research paper was taken from Kaggle and uploaded by Niharika Pandit [1]. It consists of yoga pose images of 5 yoga asanas. The 5 asanas are the 'downward dog pose', the 'goddess pose', the 'plank pose', the 'tree pose', and the 'warrior pose'. A total of 1551 images were used. Of these, some were used for testing and the remaining were used for training. Fig.1 shows sample images from the dataset

The images were resized from 1280x720 pixels to 100x100 pixels for ease of computing using the PIL library in python.

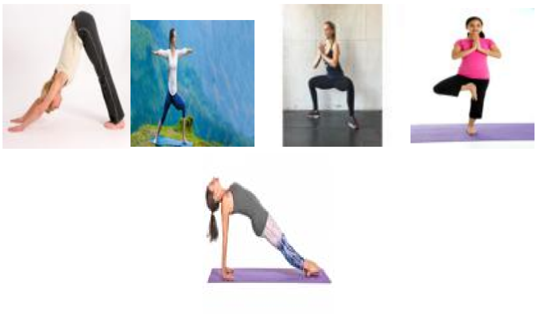

Fig. 1. A sample of images from dataset

A dataset of 1000 images and 100x100 pixels was also generated using the 'ImageDataGenerator' from the tensorflow library.

B] Input Layer

In this study, the yoga pose images had to be compatible with a pre-trained transfer learning model in order to extract features from the yoga pose images and classify them correctly. A simple preprocessing step normalized the input image (100 × 100 × 3 pixels) to the interval [0, 1]. Next, split the 5-class image dataset into categories for training and testing. The training images were the input to the pretrained model layers and features were extracted. The pretrained models were able to classify yoga poses based on the class labels assigned to the training data set.

C] Pre-trained model layers

Pretrained models are trained and developed by other developers. They are typically used to solve problems based on deep learning and are constantly trained on huge data sets. Each pre-trained model is constructed using two parts: a convolutional basis and a classifier. The convolutional foundation consists of convolutional layers and pooling layers. The convolutional layers transform the image and extract features from it and the pooling layers reduce the dimensions of the features without losing any important information. Classifiers are responsible for classifying images based on their characteristics. In the pre-trained model layers, the convolution base used the weights/bias from the ImageNet dataset and the classifier was removed. Extra layers were added to the model and the classifier was replaced with another for yoga pose image classification.

D] Additional layers

Additional layers were added to one of the pre-trained models. Flatten layer was used to make the multidimensional output into 1D so that it could go through the next dense layers. The first layer was added with 256 nodes, the second with 128 nodes, the third with 64 nodes, the fourth with 32, and the fifth with 16 nodes. The 'ReLU activation' function was used in the dense layers. The graphical representation is shown in Fig. 2

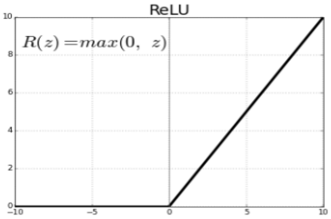

Fig. 2: ReLU activation function

Dropout layers were added to avoid overfitting. On the final layer, the softmax activation function was used. The 'softmax function' is represented by, $\sigma(\hat{z}_j)$ in the below equation [9].

$$\sigma(\hat{z}_j) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

These layers are then stacked sequentially to form an entire connected network. Since this is a classification task, the loss function used was the cross entropy loss function and the optimizer used was the Adam optimizer. 'Early stopping' was also used to avoid overfitting. The categorical cross-entropy calculates the loss by computing the sum represented in the equation below [10], where $\hat{y}_i$ is the ith scalar value in the model's output, y_i is the corresponding target value and the number of scalar values in the model output is the output size.

$$L(y_i\ ,\hat{y}_i\ ) = -\sum_{i=1}^{k} y_i . log\hat{y}_i$$

where k is the size of the output

E] Data augmentation

Data augmentation was performed on each training set of yoga pose images to maintain dataset diversity and prevent overfitting. In the proposed study, data augmentation was performed in five methods: (i) Rotation angles between −30° and 30° were randomly chosen for image rotation. (ii) an image shift in the range of 0.2 was used, (iii) horizontal flip was enabled, (iv) image zooming in the range of 0.2 was implemented, and (v) image shearing in the range of 0.2 was also applied. All mentioned pre-trained models, including additional layers, were trained on both the augmented and un-augmented yoga pose datasets and their individual performances were evaluated and compared.

F] Transfer learning

Transfer learning is the learning or the improvement of learning on a new task through the transfer of knowledge from a similar task that has previously been learned [8]. Transfer learning is widely used for building models with small datasets and in a short span of time. The advantages of using transfer learning over the conventional machine-learning methods are (i) less pre-processing time is required for the dataset (ii) the learning process is faster (iii) such models are trained on extremely large datasets for a long time as a result these models work very well on limited datasets (iv) optimization and regulation of various parameters can adjust in time complexity. A total of 10 transfer learning models were used in this paper and their results were compared. The models used are VGG16, VGG19, InceptionV3, DenseNet201, ResNet50V2,

ResNet152V2, ResNet101V2, MobileNet, MoileNetV2, and InceptionResNetV2. These models were previously trained on the ImageNet dataset for classification of 1000 object categories. These pre-trained models are using weights of the 'ImageNet' dataset for the classification of the yoga pose dataset into five classes. All ten models require an input size of (100 × 100 × 3). Fig. 3 shows the overall architecture of the model.

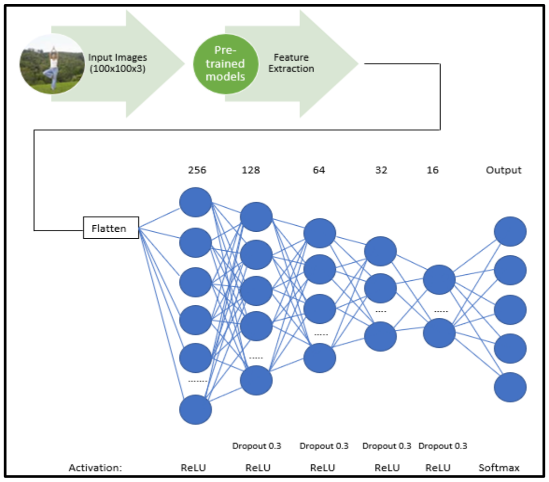

Fig. 3: Architecture of the model

IV. RESEARCH FINDINGS

The table1 shows the time taken by the pre-trained models during the training process on the augmented as well as unaugmented data. For better time complexity management and to avoid overfitting we have used the 'Early stopping' callback function. It observes the model's performance in each epoch on held-out validation set during training and stops training depending on validation performance. We have kept the patience value 3, so each model is trained for a different epoch according to its performance.

Table1: Time taken by pre-trained models

Models	Time (ms/step) [Unaugmented Data]	Time (ms/step) [Augmented Data]
ResNet152V2	3393	5325
ResNet101V2	2271	4213
VGG19	3365	5302
VGG16	3300	4248
DenseNet201	2213	3160
InceptionResNetV2	1169	2135
ResNet50V2	1146	2118
InceptionV3	1088	1067
MobileNetV2	1064	1053
MobileNet	56	1040

Analysis and Results

For each model, the classification is evaluated using the accuracy score, weighted average recall score, weighted average precision, and weighted average f1 score.

A] Precision

Precision is an evaluation metric that gives the number of correct positive predictions generated by the model. Precision, therefore, finds the accuracy for the minority class. It is the ratio of true positives (samples that were correctly predicted as positive) divided by the number of samples that were predicted as positive.

In a classification problem where classes are not balanced and with more than two classes, precision is the ratio of total number of true positives in all classes to the total of true positives and false positives in all classes.

$$Precision\ =\frac{Sum(True\ Positives)}{Sum(True\ Positives+False\ Positives)}$$

The weighted-average precision is calculated by finding the mean of precision scores for all classes considering the support of each class. Support is the actual number of occurrences of the class in the dataset.

The weighted average precision score for each model on augmented as well as unaugmented data is illustrated in table 2.

Table2: Weighted average precision score

Models	Weighted Average Precision Score [Unaugmented data]	Weighted Average Precision Score [Augmented data]
ResNet152V2	0.92	0.92
ResNet101V2	0.94	0.94
VGG19	0.93	0.95
VGG16	0.95	0.96
DenseNet201	0.94	0.92
InceptionResNetV2	0.87	0.88
ResNet50V2	0.94	0.94
InceptionV3	0.87	0.85
MobileNetV2	0.94	0.93
MobileNet	0.93	0.93

B] Recall

Recall is a scoring metric that indicates the number of correct positive predictions out of all possible positive predictions produced by the model. Precision only gives an indication of the positive predictions that are correct out of all positive predictions, but recall provides an indication of the positive predictions that were missed by the model.

In a classification problem where classes are not balanced and with more than two classes, one class, the recall is the ratio of the total true positives for all classes to the total true positives and false negatives for all classes.

$$Recall = \frac{Sum(True\ Positives)}{Sum(True\ Positives + False\ Negative)}$$

The weighted-average recall score is the mean of all recall scores for all classes while considering the support of each class.

The weighted average recall score for each model is illustrated in table 3.

Table3: Weighted average recall score

Models	Weighted Average Recall Score [Unaugmented data]	Weighted Average Recall Score [Augmented data]
ResNet152V2	0.91	0.79
ResNet101V2	0.94	0.94
VGG19	0.93	0.94
VGG16	0.94	0.96
DenseNet201	0.94	0.91
InceptionResNetV2	0.86	0.88
ResNet50V2	0.94	0.94
InceptionV3	0.86	0.84
MobileNetV2	0.93	0.93
MobileNet	0.93	0.92

C] F1 score

F1 score combines both precision and recall in a single quantity that captures both properties. Once precision and recall are calculated for classification problems, the two scores can be combined to calculate the F1 score.

$$F1\ score\ = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$$

The weighted-average F1 score is the mean of all F1 scores for all classes considering the support of each class.

The weighted average f1 score for each model is illustrated in table 4.

Table4: Weighted average F1 score

Models	Weighted Average F1 Score [Unaugmented data]	Weighted Average F1 Score [Augmented data]
ResNet152V2	0.92	0.84
ResNet101V2	0.94	0.94
VGG19	0.93	0.94

VGG16	0.94	0.96
DenseNet201	0.94	0.91
InceptionResNetV2	0.86	0.88
ResNet50V2	0.94	0.94
InceptionV3	0.86	0.84
MobileNetV2	0.93	0.93
MobileNet	0.93	0.92

D] Accuracy

The accuracy score gives us the fraction of the predictions that the model got correct. It is useful when all classes are equally important. It is computed as the ratio between the number of correct predictions and the total number of predictions.

The accuracy is computed as follows:

$$Accuracy = \frac{Number\ of\ predictions\ that\ are\ correct}{Total\ number\ of\ predictions}$$

The following table 5 shows the accuracy achieved by each model on the test dataset.

Table5: Accuracy of the model

Model	Accuracy (Unaugmented Data)	Accuracy (Augmented Data)
ResNet152V2	91.49 %	78.94 %
ResNet101V2	93.83 %	93.83 %
VGG19	93.19 %	94.47 %
VGG16	94.47 %	95.74 %
DenseNet201	93.26 %	91.06 %
InceptionResNetV2	85.74 %	87.66 %
ResNet50V2	93.83 %	93.83 %
InceptionV3	86.38 %	84.47 %
MobileNetV2	93.40 %	92.55 %
MobileNet	92.55 %	91.70 %

As we can see, the VGG16 performs better as compared to other models on augmented data as well as on unaugmented data with an accuracy of 95.74% and 94.47% respectively. The other models also give fairly good results.

The confusion matrix, accuracy, and loss of the VGG16 on augmented data are shown in Fig. 4.

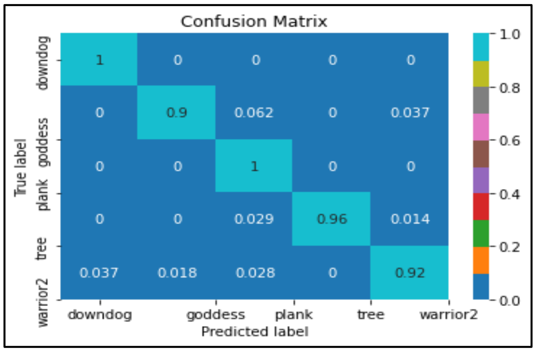

Fig. 4 Confusion matrix of VGG16 on augmented data

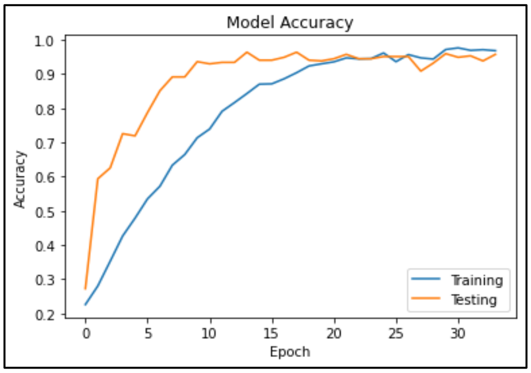

Fig.5 Model accuracy [For VGG16 on augmented data]

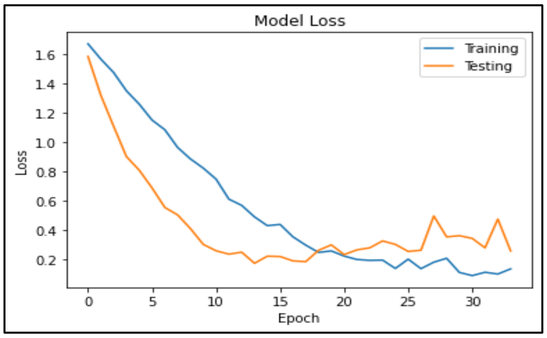

Fig.6 Loss plot [For VGG16 on augmented data].

The below images show the confusion matrix, accuracy, and loss for VGG16 on unaugmented data.

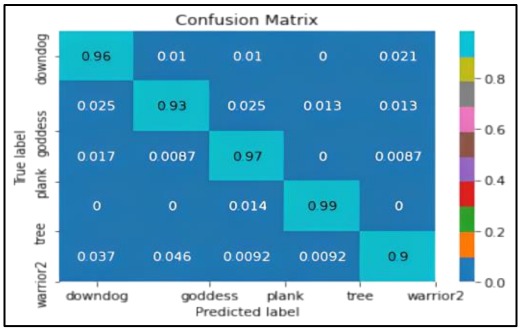

Fig.7 Confusion matrix of VGG16 on unaugmented data.

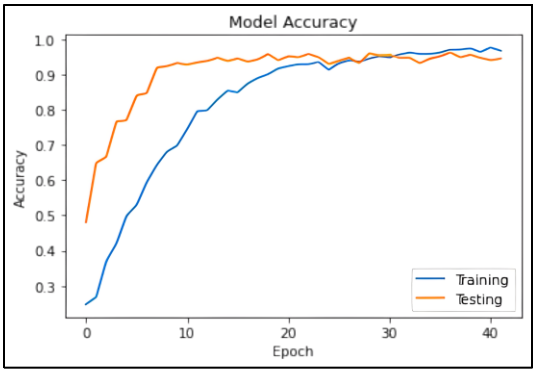

Fig. 8 Model accuracy [For VGG16 on unaugmented data].

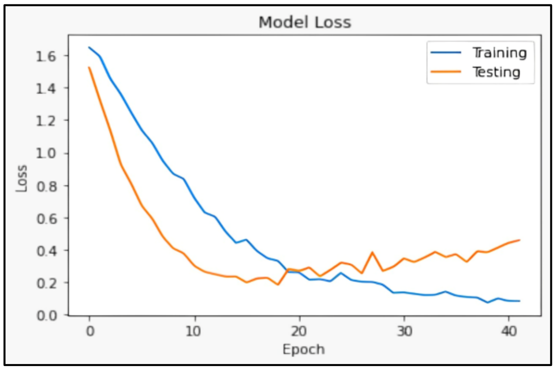

Fig. 9 Loss incurred [For VGG16 on unaugmented data].

V. CONCLUSION AND FUTURE SCOPE

In our paper, VGG16 gave the best performance. VGG16 is a 16 layer deep convolutional neural network. It is mostly used for classification and object detection. It is one of the popular choices for classification, as it achieves high accuracy. Several variants of ResNetV2 model have been used due to the difference in their architecture and extracting features based on it. Several variants of the model were used to test whether the model produced significant differences.

It can also be seen that augmented data does not give us much different results as compared to unaugmented. Usually, augmented data is used to improve accuracy, but in our observation, it performs almost the same. In some cases, the accuracy is also less as compared to unaugmented data. This could be due to the models not being able to read the new generated data well.

During the 2019 Covid pandemic, people were forced to stay isolated in their homes. This limited their physical movement and exercises were then done indoors. During this time, online applications which provide guidance for exercise and yoga proved to be extremely beneficial. Technologies like these can be used which provide and guide users with the means to identify the correct postures of yoga asanas. Instead of using images, we can also use videos to analyse yoga postures. Deep-pose estimators, LSTM, etc are compatible with video analysis.

VI. REFERENCES

[1]. Kaggle.com, ‘Yoga Poses Dataset’, 2022. [Online]. Available: https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset [Accessed: Sep-2022]

[2]. wikipedia.com, ‘Yoga’, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Yoga [Accessed: Sep-2022]

[3]. hopkinsmedicine.org,,‘9 benefits of Yoga’, 2022. [Online]. Available: https://www.hopkinsmedicine.org/health/wellness-and-prevention/9-benefits-of-yoga [Accessed: Sep-2022]

[4]. Long, C., et al., “Development of a yoga posture coaching system using an interactive display based on transfer learning.” https://rdcu.be/cYKjv

[5]. Debabrata Swain., et al, ‘Yoga Pose Monitoring System using Deep Learning’,*Research Square*, June 2022 https://assets.researchsquare.com/files/rs-1774107/v1/c134f9a8-c453-45b5-a16a-541b34fde471.pdf?c=1656926080

[6]. S. Abarna., et al, ‘Skeleton Pose Estimation Features-based Classification of Yoga Asana using Deep Learning Techniques’, *International Journal of Mechanical Engineering*, vol.7,Feb 2022 https://kalaharijournals.com/resources/FebV7_I2_281.pdf

[7]. J. Jose., et al, ‘Yoga Asana Identification: A Deep Learning Approach’, *IOP Conference Series: Materials Science and Engineering*, 2021 https://iopscience.iop.org/article/10.1088/1757-899X/1110/1/012002/pdf

[8]. L. Torrey, ‘Transfer Learning’, *University of Wisconsin, Madison WI, USA* https://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf

[9]. wikipedia.com ‘Softmax activation function’,2022.[Online]. Available: https://en.wikipedia.org/wiki/Softmax_function [Accessed: Sep-2022]

[10]. oreilly.com, ‘Multi-class cross entropy loss’, 2022. [Online]. Available: https://www.oreilly.com/library/view/hands-on-convolutional-neural/9781789130331/7f34b72e-f571-49d2-a37a-4ed6f8011c93.xhtml [Accessed: Sep-2022]

[11]. machinelearningmastery.com, ‘How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification’, 2022.[Online]. Available: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/ [Accessed: Sep-2022]

[12]. S. Gupta, et al., ‘Yoga Pose Detection Using Deep Learning’, *International Journal of Innovative Research in Engineering, Volume 3, Issue 6 (November-December 2022), PP: 92-94.* https://theijire.com/assets/pdf/archives/1668833195_78987abdde335ec66914.pdf

[13]. S. K. Yadav, et al, ‘Real-time Yoga recognition using deep learning’, *Neural Computing and Applications (2019) 31:9349–9361* https://link.springer.com/article/10.1007/s00521-019-04232-7

[14]. J. Kutalek, ‘Detection of Yoga Poses in Image and Video’, *BRNO University of Technology, Excel@FIT2021* https://excel.fit.vutbr.cz/submissions/2021/024/24.pdf

[15]. U. Bahukhandi, et al, ‘Yoga Pose Detection and Classification using Machine Learning Techniques’, *International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal) Volume:03/Issue:12/December-2021* https://www.irjmets.com/uploadedfiles/paper/volume_3/issue_12_december_2021/17486/final/fin_irjmets1638881020.pdf

[16]. Y. Agarwal, et al, ‘Implementation of Machine Learning Technique for Identification of Yoga Poses’, *9th IEEE International Conference on Communication Systems and Network Technologies* https://ieeexplore.ieee.org/abstract/document/9115758