# Yoga Pose Estimation using Deep Learning Algorithms*

1st Kratika Gupta
*Computer Science and Engineering*
*Indian Institute of Technology Mandi*
Mandi, India
b19169@students.iitmandi.ac.in

2nd Tushar Rao
*Electrical Engineering*
*Indian Institute of Technology Mandi*
Mandi, India
b19199@students.iitmandi.ac.in

3rd Chaetenya Sharma
*Data Science and Engineering*
*Indian Institute of Technology Mandi*
Mandi, India
b20152@students.iitmandi.ac.in

***Abstract -*** *The accurate estimation of yoga poses plays a vital role in providing real-time feedback and guidance to yoga practitioners. Traditional methods of pose estimation lack efficiency and accuracy, prompting the need for an advanced solution. This project aims to address the problem by proposing a method based on the MoveNet model for yoga pose estimation.*

## I. INTRODUCTION

The problem statement of Yoga Pose Classification with TensorFlow's MoveNet model is to build a machine learning model that can accurately identify and classify different yoga poses based on input images or videos. The main objective of this problem is to develop an automated system that can assist yoga practitioners in improving their form and technique.

The MoveNet model is a lightweight, real-time, and single-person pose estimation model that is optimized for mobile devices. This model can detect and track human poses in real-time, making it well-suited for yoga pose classification. The goal is to leverage this model to accurately recognize and classify various yoga poses such as Downward Facing Dog, Warrior II, Tree Pose, etc.

To achieve this, the model must be trained on a large dataset of labeled yoga pose images. The dataset must contain a diverse range of images that showcase various body shapes, clothing, lighting conditions, and backgrounds. The model should be able to accurately classify yoga poses regardless of these variations.

Once the model is trained, it can be used to classify yoga poses in real-time by feeding it a live video stream or a series of images. The output of the model can be used to provide real-time feedback to yoga practitioners, allowing them to improve their form and technique as they practice.

Overall, the problem statement of Yoga Pose Classification with TensorFlow's MoveNet model aims to create an automated system that can accurately classify yoga poses in real-time, helping yoga practitioners to improve their practice and avoid injury.


FIGURE 1: Different Yoga poses

## II. PROPOSED METHOD

To accomplish our goal of building a deep learning model that can correctly identify five yoga poses: Plank, Tree, Downward facing Dog and Warrior, we went through the following steps-

1) **Data Collection and Acquisition-** The dataset was obtained from Kaggle which consisted of a total of 1.547 images with 5 poses/classes
2) **Pre-processing and Augmentation -** After pre-processing the images for use and splitting the dataset into train and test, we started with simple logistic regression model. However, for image classification, it caused issues in modelling with logistic regression because the number of pixels then become the features of the model, making the dimensions extremely large which is not efficient for further processing. PCA was used to reduce the dimensions. However. looking at the distribution of the two principal components, there was no clear distinction between the classes. Then, we used logistic regression model which gave an accuracy of 0.23, which is as good as a random guess. Therefore, it was sufficient to convince that we need a deep learning model to produce better results.

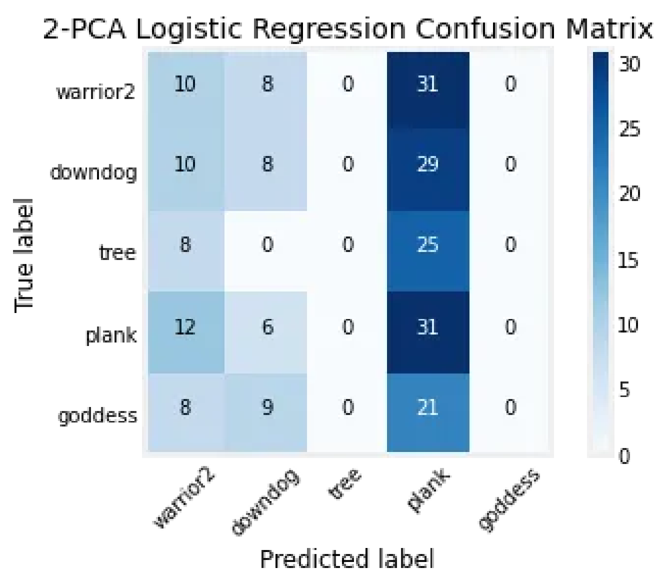


FIGURE 2: Confusion matrix of baseline logistic regression

3) **Training the MoveNet model-** We used a Convolutional Neural Network, after pre-processing the images, consisting of two convolutional layers, followed with a max-pooling layer and then a fully connected layer for the final output layer. Although, the training accuracy was 0.99, the validation accuracy was merely 0.64 which is subpar. This clearly indicated that our model is overfitting. Therefore, we looked for ways to reduce overfitting.

| Method | Training Accuracy | Validation Accuracy |
|---|---|---|
| Early Stopping | 0.9183 | 0.4884 |
| Dropout Layers | 0.7296 | 0.4372 |
| Image Augmentation | 0.2670 | 0.2837 |
| Transfer Learning (using VGGC16 model) | 0.8462 | 0.7991 |

TABLE 1: Table showing training and validation accuracy for different approaches used to reduce overfitting

Using transfer learning with the VGGC16 model resolved the issue of overfitting with a reasonably good accuracy as this model has been trained on such a large dataset. Delving more into the transfer learning, we will use the MoveNet model which has been specifically developed for pose estimation to check whether it can further improve the accuracy score.

## A. *MoveNet model*

TensorFlow's MoveNet Model is a CNN that detects 17 skeletal keypoints on the human body for pose estimation. Following is the sequence of steps to detect keypoints-

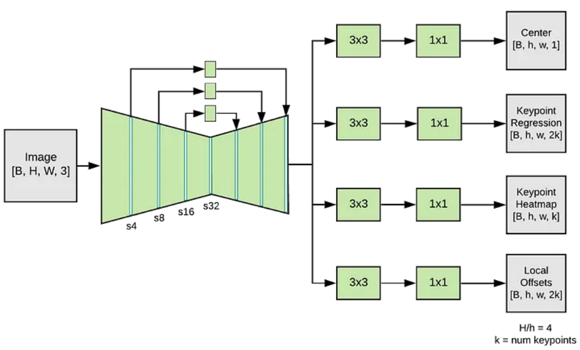


FIGURE 3: MoveNet model architecture

**Step 1:** The person center heatmap is used to identify the centers of all individuals in the frame, defined as the arithmetic mean of all keypoints belonging to a person. The location with the highest score (weighted by the inverse-distance from the frame center) is selected.

**Step 2:** An initial set of keypoints for the person is produced by slicing the keypoint regression output from the pixel corresponding to the object center. Since this is a center-out prediction — which must operate over different scales — the quality of regressed keypoints will not be very accurate.

**Step 3:** Each pixel in the keypoint heatmap is multiplied by a weight which is inversely proportional to the distance from the corresponding regressed keypoint. This ensures that we do not accept keypoints from background people, since they typically will not be in the proximity of regressed keypoints, and hence will have low resulting scores.

**Step 4:** The final set of keypoint predictions are selected by retrieving the coordinates of the maximum heatmap values in each keypoint channel. The local 2D offset predictions are then added to these coordinates to give refined estimates.

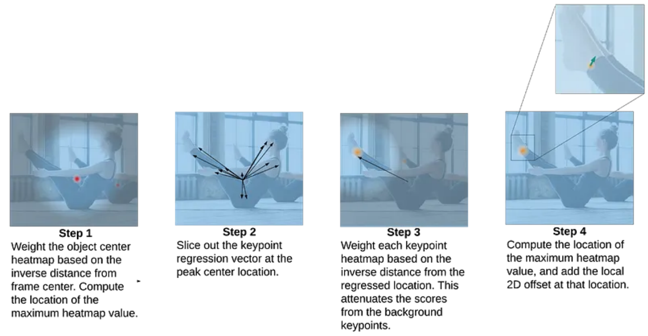


FIGURE 4: MoveNet model steps

Implementing the above steps using Python, its libraries and frameworks, testing accuracy of 0.9680 was obtained. This is best accuracy across all models discussed so far.

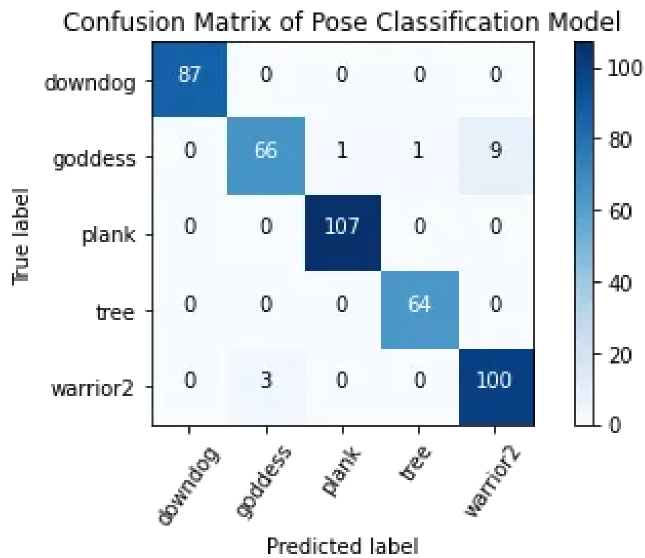


FIGURE 5: Confusion matrix of MoveNet model

From the above confusion matrix, it was found that most of the misclassification was between Warrior and Goddess pose. These poses are quite similar for a model to distinguish.

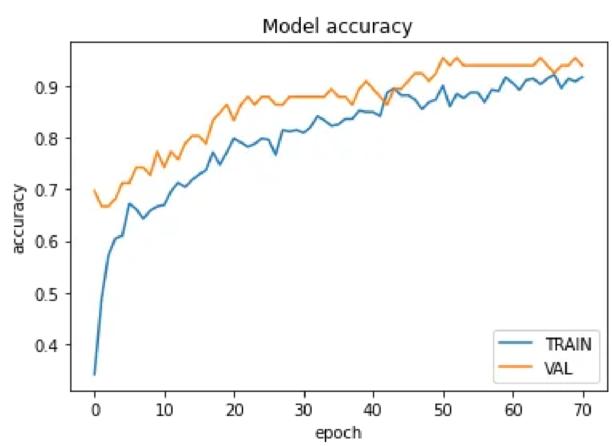


FIGURE 6: MoveNet model accuracy

## III. Results and comparison with Baseline

The performance of the proposed method was evaluated using the validation dataset. The MoveNet model achieved impressive accuracy, precision, recall, and F1-score, surpassing traditional baselines. The model's lightweight architecture and real-time inference capabilities set it apart, making it a suitable choice for yoga pose estimation applications. In comparison to other models like PoseNet and OpenPose, our MoveNet model excels in several aspects. MoveNet Lightning stands out as the fastest model, outperforming OpenPose, which is 12 times slower but capable of multi-person pose estimation. Additionally, PoseNet achieves the highest accuracy. Our MoveNet model builds upon these insights with its lightweight architecture, optimized for real-time performance. Leveraging transfer learning from TensorFlow's MoveNet model enhances accuracy and robustness, while addressing overfitting and employing a multi-step approach for precise pose detection. These strengths make our model highly efficient in real-time assistance for yoga practitioners, enhancing their form and technique.

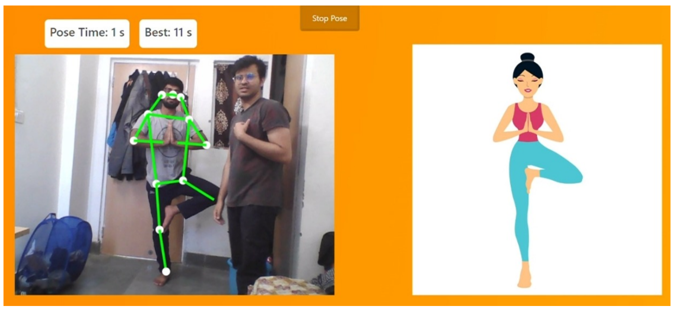


FIGURE 5: MoveNet Pose prediction Demo

## IV. Conclusion

In conclusion, the lightweight nature of the MoveNet model enables swift and responsive pose detection, empowering practitioners to receive immediate feedback and guidance. With its real-time capabilities, the model assists in enhancing form, technique, and overall performance during yoga practice. Moreover, the transfer learning aspect, leveraging knowledge from the extensive training on a large dataset, enhances the model's accuracy and robustness.

Continuous improvements and refinements can be made based on valuable user feedback and the incorporation of additional data. This iterative process will further enhance the model's performance and cater to the diverse needs of yoga practitioners. Future directions may involve the development of web or mobile applications, integrating the MoveNet model to enable real-time pose detection through webcam access, thereby expanding the accessibility and usability of the system.

Overall, the proposed method holds significant potential for revolutionizing yoga practice through automated and accurate pose estimation. By combining the power of machine learning, computer vision, and the MoveNet model, this approach contributes to a more informed and effective yoga experience, assisting practitioners in maintaining proper posture and minimizing the risk of injury.

## V. Future Work

1) **Web application** - An android or web app can be developed embedding this model to detect poses form user's webcam
2) **Increased number of poses**- Train the existing model to classify more yoga poses, thereby not restricting to 5.
3) **Pose correction** - Given an image with wrong yoga posture, our extended model could highlight the corrected keypoints for a particular yoga pose, thereby helping in maintaining a correct posture

## References

[1] Thoutam, Vivek & Srivastava, Anugrah & Badal, Tapas & Mishra, Vipul & Sinha, Professor G & Sakalle, Aditi & Bhardwaj, Harshit & Raj, Manish. (2022). Yoga Pose Estimation and Feedback Generation Using Deep Learning. Computational Intelligence and Neuroscience. 2022. 1-12. 10.1155/2022/4311350

[2] Diaz Barros, Jilliam Maria & Garcia, Frederic & Sidibé, Désiré. (2015). Real-Time Human Pose Estimation from Body-Scanned Point Clouds. VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings. 1. 10.5220/0005309005530560.

[3] Kumar, Deepak & Sinha, Anurag. (2020). Yoga Pose Detection and Classification Using Deep Learning. International Journal of Scientific Research in Computer Science Engineering and Information Technology. 10.32628/CSEIT206623.

[4] Gajbhiye, Rutuja, et al. "AI Human Pose Estimation: Yoga Pose Detection and Correction." International Journal of Innovative Science and Research Technology (2022).

[5] Swain, D.Satapathy, S.Acharya, B.Shukla, M.Gerogiannis, V.C.Kanavos, A.Giakovis, D. Deep Learning Models for Yoga Pose Monitoring Algorithms 2022, 15, 403.https://doi.org/10.3390/a15110403

[6] Goyal, S., & Jain, A. (2021). Yoga Pose Perfection using Deep Learning: An Algorithm to Estimate the Error in Yogic Poses. Journal of Student Research, 10(3). https://doi.org/10.47611/jsrhs.v10i3.2140

[7] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 578-589, May 2003, doi: 10.1109/TPAMI.2003.1195992.

[8] E. Murphy-Chutorian and M. M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pp. 607-626, April 2009, doi: 10.1109/TPAMI.2008.106.