



Yoga Pose Identification Using Deep Learning

Team:

Akshaya Nagarajan

Asha Aher

Swati Narkhede



Abstract

- In day to day life, one must exercise to remain fit. Many physical trainings have been benefitted immensely from leveraging Artificial Intelligence. However, Yoga remains relatively less explored. So, goal of our project is to recognize the yoga poses.
- This project will classify different yoga poses.
- The input to our algorithm would be a Image of a person doing yoga which is given to pretrained Inception model
- Using these we will train model to identify correct Yoga Pose. We are executing the model in TPU hence the speed of execution has increased immensely.
- As a final output,The model classifies the Yoga Poses.

Introduction

- Yoga is an ancient and popular type of exercise. In last few years, yoga has gained more popularity around the world.
- Yoga practice has become a part of daily life of many people. Apart from this, many people are taking interest in learning yoga poses.
- In this project, we have built an application using Deep Learning Neural Networks that performs yoga pose classification.
- A user can upload an image of any one yoga pose out of the total 9 classes of yoga poses, viz. are Low Lunge, Downward Facing Dog, Tree Pose, Warrior Pose, Planks, Reverse Planks, Side Planks, Seated Forward Bend and Triangle Pose.
- The application will return a name of the correct form of yoga pose.

Related Work

- Initially we started Exploring CNN architecture and trained the model accordingly.
- Later we explored more about Vgg16 and trained the model on Vgg16.
- The epochs took lot of time to execute so we explored more about Collab TPU and executed our model into TPU.
- Eventually TPU increased the execution speed of epochs.
- Many works have been performed in Yoga Pose classification.

Dataset

- We are working on the Yoga pose images as an input to our model. These images are with extension png, jpg, etc.
- Initially, we were planning to use images from the yoga pose dataset available in [this](#) article. However, due to less number of images available in this dataset, we collected Yoga pose images from various websites and search engines such as Google, Yahoo, Bing and Baidu using web scraping.
- When we collected the images, there were many irrelevant images present in collected images. We performed cleaning of images and splitted them into test and train folders.
- There was not special treatment needed to be used those images in our project.

Methods

- Since our problem domain was to classify the yoga poses, we started exploring the dataset which can be big enough to train the model, we collected images and manually cleaned them. the dataset Contains 9 classes
- We explored CNN architecture but in order to get higher accuracy we considered another alternative which is VGG16 and trained 2 classes on VGG16.
- For all classes classification we trained Inception model tuning some of the hyper parameters.
- initially we were executing the model on GPU but in order to speed up execution we integrated Collab TPU.
- Below is the Comparison table for approaches we explored.

	Initial Architectural Approach	Second Architectural Approach	Third Architectural Approach	Fourth Architectural Approach	Fifth Architectural Approach
	CNN	VGG16 model with 2 classes	Inception model with 2 classes	Inception model with all classes	VGG16 model with all classes
Results	Predicted Close to pose	57%	86%	19%	32%

Experiments

- We performed image augmentation to the whole dataset and augmented images were used to train model.
- Inception - In this architecture we used transfer learning and tuned the model freezing and unfreezing the layers also we increased epochs in order to get best accuracy. The final accuracy we got is 19% for all the classes in datasets.
 - Layers : GlobalAveragePooling2D
 - Activation Functions: relu and softmax
 - Optimiser: rmsprop
 - Fine Tuning Layer - Freeze and Unfreeze the layer
- Vgg16 - We trained our model using Vgg16 architecture and initially we trained the model only with two classes and it performed fairly good with with 57% accuracy
 - Layers : GlobalAveragePooling2D
 - Activation Functions: relu and softmax
 - Optimiser: SGD
 - Fine Tuning Layer - Freeze and Unfreeze the layer
- CNN - CNN model was used as a part of experiment, it was predicting the image very close to pose so for better performance we thought of trying couple of another alternatives.
 - Layers: Convolution2D and Max Pooling layers.
 - Activation Functions: relu and softmax

Deployment of model

Welcome to Yoga Pose Classifier!

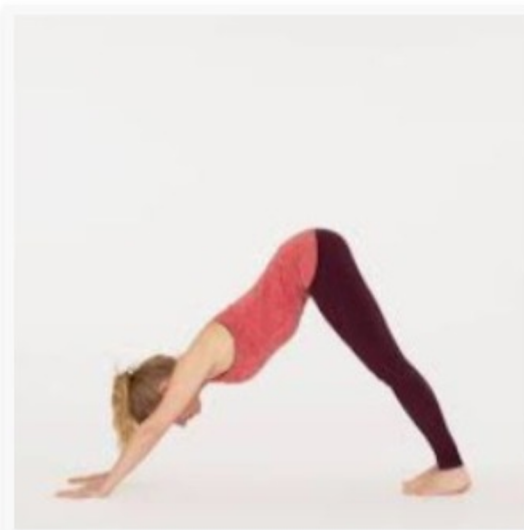
Just upload an image and we will predict the yoga pose depicted in the picture...

Select Image

Welcome to Yoga Pose Classifier!

Just upload an image and we will predict the yoga pose depicted in the picture...

Select Image



Predict Yoga Pose...

Welcome to Yoga Pose Classifier!

Just upload an image and we will predict the yoga pose depicted in the picture...

Select Image



Yoga Pose: DownwardFacingDog

Conclusion

- As a result, our Yoga Pose Classification is successfully able to identify the correct class of Yoga Pose. Our model has achieved an accuracy of 57 and 19%.
- We have successfully deployed our Yoga Poses Classification model on webpage using Python Flask.
- While working on this project, we have learnt implementing CNN, VGG 16 and Inception model. Along with that we learnt tuning the model and visualizations.
- As a part of future extension of this application, we would like to perform the yoga pose in real-time. Along with that, we would like to suggest the user a correct posture to practice certain yoga pose correctly.

Supplementary Material

- Source code: <https://github.com/akshaya-nagarajan/DeepLearningProjects/tree/master/Project>
- Dataset and savedmodel :
<https://drive.google.com/open?id=1FnY7aRFtOl6jTJ93t0RySKqvIHOz82sX>

Thank You!
