Human Body posture detection and analysis for exercise/yoga with deep learning and open CV

Authors: Nihar Mohanty, Harsh Kumar Mishra, Anand Singh

Concept Description of Project:

Our project is based on Computer vision application where we can able to detect the right posture for yoga ananas. Here we use the deep learning concept using open CV library package to estimate the right posture for yoga asana so that we could do the yoga exercises in a proper way. We built a CNN model to train our dataset where we try to detect the posture and enhance the accuracy of estimating the correct posture of the yoga.

Abstract:

Human pose estimation has been an important research topic for computer vision community. It has been in focus of researchers mainly because of its significant applications in various important fields like human computer interaction, action recognition, surveillance, picture understanding, threat prediction, etc. It's difficult to cover all aspects of this domain because of the diversity of its application areas, therefore this review is focused on the most significant contributions in Human Pose Estimation methods from a single two-dimensional image. Self-training plays an important role in sports exercise, but improper training postures can cause serious harm to muscles and ligaments of the body. Hence, more and more researchers are devoted into the development of computer-assisted b self-training systems for sports exercise. In this paper, we propose a Yoga posture recognition system, which is capable of recognizing what Yoga posture the practitioner is performing, and then retrieving Yoga training information from Internet to remind his/her attention to the posture. This project lays the foundation for building such a system by discussing various machine learning and deep learning approaches to accurately classify yoga poses on train and also in real-time. The project also discusses various pose estimation and key point detection methods in detail and explains different deep learning models used for pose classification.

Introduction:

Human pose estimation is a challenging problem in the discipline of computer vision. It deals with localization of human joints in an image or video to form a

skeletal representation. To automatically detect a person's pose in an image is a difficult task as it depends on a number of aspects such as scale and resolution of the image, illumination variation, background clutter, clothing variations, surroundings, and interaction of humans with the surroundings. An application of pose estimation which has attracted many researchers in this field is exercise and fitness. One form of exercise with intricate postures is yoga which is an age-old exercise that started in India but is now famous worldwide because of its many spirituals, physical and mental benefits. The problem with yoga however is that, just like any other exercise, it is of utmost importance to practice it correctly as any incorrect posture during a yoga session can be unproductive and possibly detrimental. This leads to the necessity of having an instructor to supervise the session and correct the individual's posture. Since not all users have access or resources to an instructor, an artificial intelligence-based application might be used to identify yoga poses and provide personalized feedback to help individuals improve their form. In recent years, human pose estimation has benefited greatly from deep learning and huge gains in performance have been achieved [3]. Deep learning approaches provide a more straightforward way of mapping the structure instead of having to deal with the dependencies between structures manually. [4] used deep learning to identify 5 exercise poses: pull up, swiss ball hamstring curl, push up, cycling and walking. However, using this method for yoga poses is relatively newer application. This project focuses on exploring the different approaches for yoga pose classification and seeks to attain insight into the following: What is pose estimation? What is deep learning? How can deep learning be applied to yoga pose classification in real-time? This project uses references from conference proceedings, published papers, technical reports and journals. Fig. 1 gives a graphical overview of topics this paper covers. The first section of the project talks about the history and importance of yoga. The second section talks about pose estimation and explains different types of pose estimation methods in detail and goes one level deeper to explain discriminative methods – learning based (deep learning) and exemplar. Different pose extraction methods are then discussed along with deep learning-based models - Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Humans are prone to musculoskeletal disorders with aging and accidents. In order to prevent this some, form of physical exercise is needed. Yoga, which is a physical and spiritual exercise, has gained tremendous significance in the community of medical researchers. Yoga has the ability to completely cure diseases without any medicines and improve physical and mental health. A vast body of literature on the medical applications of yoga has been generated which includes positive body image intervention, cardiac rehabilitation, mental illness etc. Yoga comprises of various asanas which represent physical static postures. The application of pose estimation for yoga is challenging as it involves complex configuration of postures. Furthermore, some state-of-the-art methods fail to perform well

when the asana involves horizontal body posture or when both the legs overlap each other. Hence, the need to develop a robust model which can help popularize self-instructed yoga systems arise.

Goals:

The primary goal of this project is to build a prototype model for the prediction of different yoga poses and other related exercises. These exercises, if performed incorrectly can have severe outcomes that includes minor to major damages to bones or muscles. A personal trainer or a gym membership is costly and also not feasible in the current scenario with strict lockdowns being implemented the almost all the regions of the country and the world. Also these places being of the cases provide an ideal home ground for various pathogens to thrive. Doctors, nutritionists as well as scientists have always backed the important role of physical exercise and balanced diet to maintain immunity and lead a healthy life. Their advises have now being backed by numerous researches. The use of a personal trainer cannot be more important than today in the world. With the world fighting against an invisible enemy that is so strong, only our immunity and precautions can help us fight against this virus. Yoga played a very important role in our ancestor's lifestyle. It helped them not just to live a prosperous and healthy life but also conquer the world both financially and spiritually. It is now time for us to make yoga a indistinguishable part of our lives and this project intends to be the guiding path towards this goal.

Design:

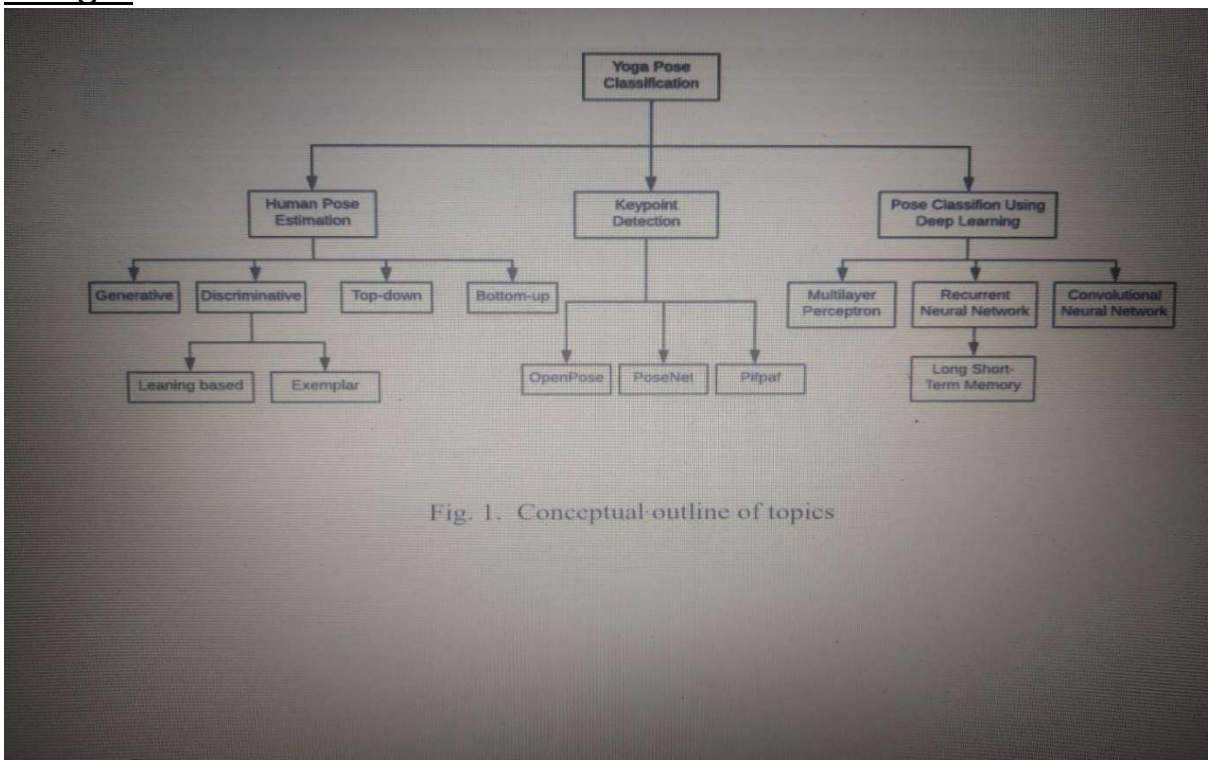


Fig. 1. Conceptual outline of topics

Design and Implementation Chart

Related Works:

There many researches have been done to build an algorithm and deep learning architectures which can detect the pose of yoga image accurately. One of that is kind is pose net. Pose Net is another deep learning framework similar to Open Pose which is used for identification of human poses in images or video sequences by identifying joint locations in a human body. The Pose Net model is invariant to the size of the picture, consequently it can anticipate present situations in the size of the genuine picture regardless of whether the picture has been downscaled. There is another method called PIFPAF method where it is based on the bottom-up approach for 2D multi-person human pose estimation. It uses a Part Intensity Field (PIF) for body part localization and a Part Association Field (PAF) for association of body parts to form full human poses. The model beats other methods in terms of a lower resolution and better performance in overcrowded places mainly due to the following: (a) fine information encoded in a newer composite field PAF, (b) the selection of Laplace loss that integrates an opinion of uncertainty. The model architecture rests upon a completely convolutional box-free design. In other technique like MLP which is an old style neural organization that has one info and one yield layer. The transitional layers between the info and yield layer are known as concealed layers. There can be at least one shrouded layers. MLPs structure a completely associated network as each hub in one layer has an association to each hub in another layer. A completely associated network is an establishment for profound learning. MLP is well known for directed characterization where the information is relegated a mark or class. [21] employments MLP for human posture order by separating key points from low goal pictures utilizing Kinect sensor.

Dataset:

We have taken our dataset from Kaggle (https://www.kaggle.com/niharika41298/yoga-poses-dataset).

This data set contains 1551 image file. These files are categorised into train and test folder. Inside them there are 5 sub folders each corresponding to 5 classes of yoga poses. The train and test data were split in 70-30 proportion. The 5 classes of yoga poses are downward dog pose, goddess pose, tree pose, plank pose and the warrior pose.

Downward dog pose: This is also known as Adho Mukha Shvanasana. Downward Dog stretches the hamstring and calf muscles in the backs of the

legs, and builds strength in the shoulders. Some popular sites have advised against it during pregnancy, but an experimental study of pregnant women found it beneficial.

Downward Dog Pose

Goddess pose: This is also known as Utkata Konasana. It opens the hips, legs, and chest. Strengthens the legs, calves, abs, and knees. Stimulates the uro-genital system and pelvic floor. Strengthens and stretches the shoulder joint.

Goddess Pose

Tree Pose: This is also known as Vrikshasana. It is one of the most fundamental balancing pose of hatha yoga. Here you are willing, trying to stand on one leg becomes an inquiry into your own truth. Honouring your truth might mean lowering the foot to a place below the knee or even to the floor, bringing the lifted knee slightly forward in space to align the hips, or gently engaging the abdomen to remove the arch from the lower back.

Tree Pose

Plank Pose: This is also known as Chaturanga Dandasana. Here the straight body parallel to the ground is supported by the toes and palms, with elbows at a right angle along the body.

Plank Pose

Warrior Pose: This is also known as Virbhadra asana. Warrior Pose is a standing yoga pose that helps build focus, power and stability. This foundational pose stretches the front side of the body and is great for building strength in the legs, core and back.

Warrior Pose

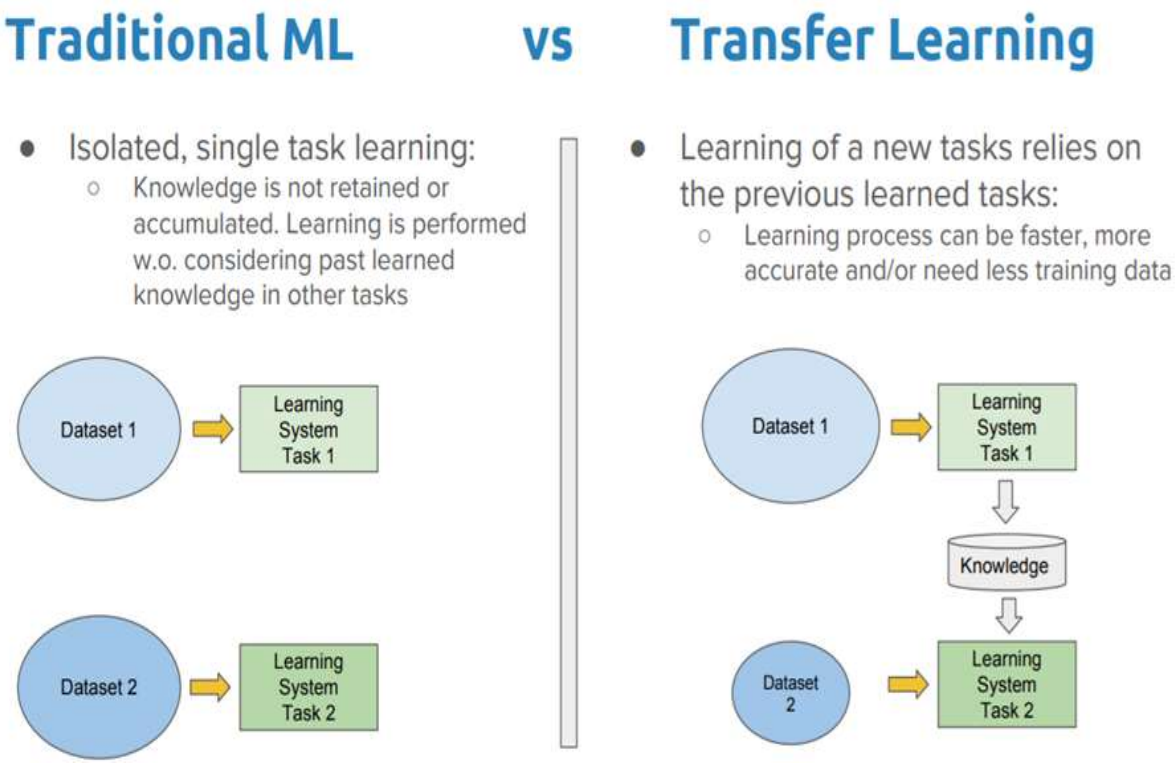
Methodology and Experiment:

As we are using deep learning model, we will use the transfer learning concept by using the pretrained
VGG16 model. We first have to understand what is transfer learning and VGG16 architecture.

Transfer Learning:

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

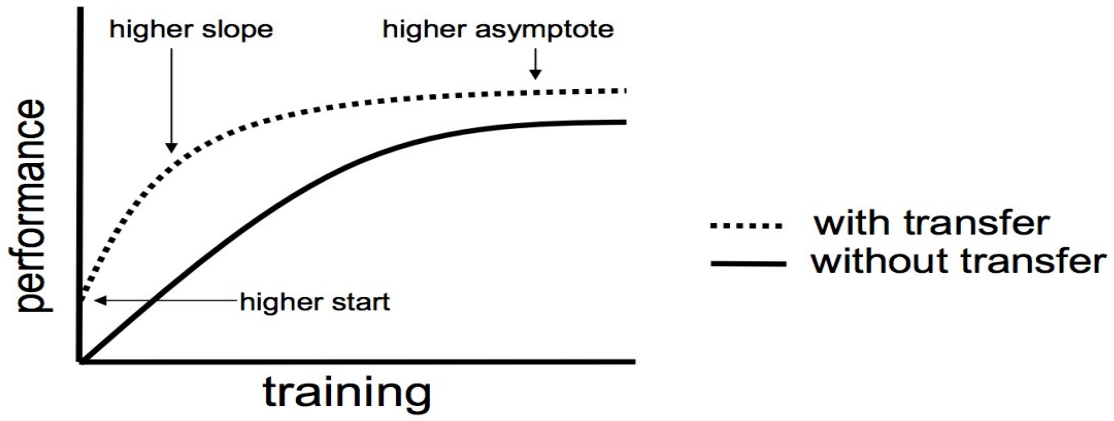
It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems. In transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. The below figure will describe the difference between Traditional ML and Transfer learning.



Transfer Learning and Traditional ML

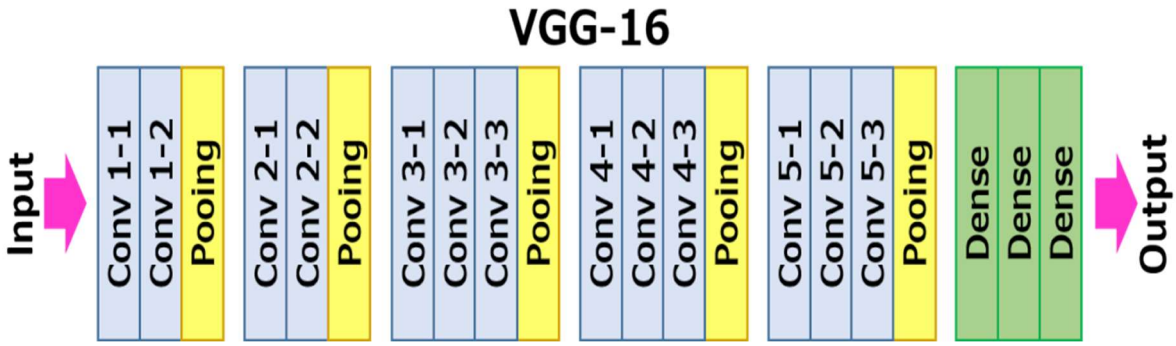
This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task. three possible benefits to look for when using transfer learning:

1. **Higher start**. The initial skill (before refining the model) on the source model is higher than it otherwise would be.
2. **Higher slope**. The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
3. **Higher asymptote**. The converged skill of the trained model is better than it otherwise would be.



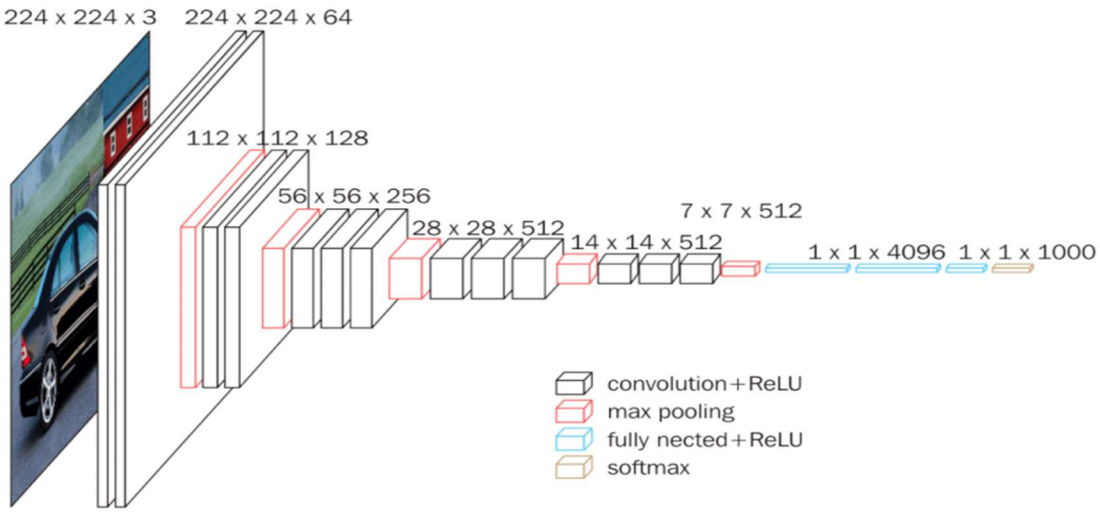
VGG 16 Architecture:

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU’s.



The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.



VGG16 Architecture

The ConvNet configurations are outlined in figure 2. The nets are referred to their names (A-E). All configurations follow the generic design present in architecture and differ only in the depth: from 11 weight layers in the network A (8 conv. and 3 FC layers) to 19 weight layers in the network E (16 conv. and 3 FC layers). The width of conv. layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

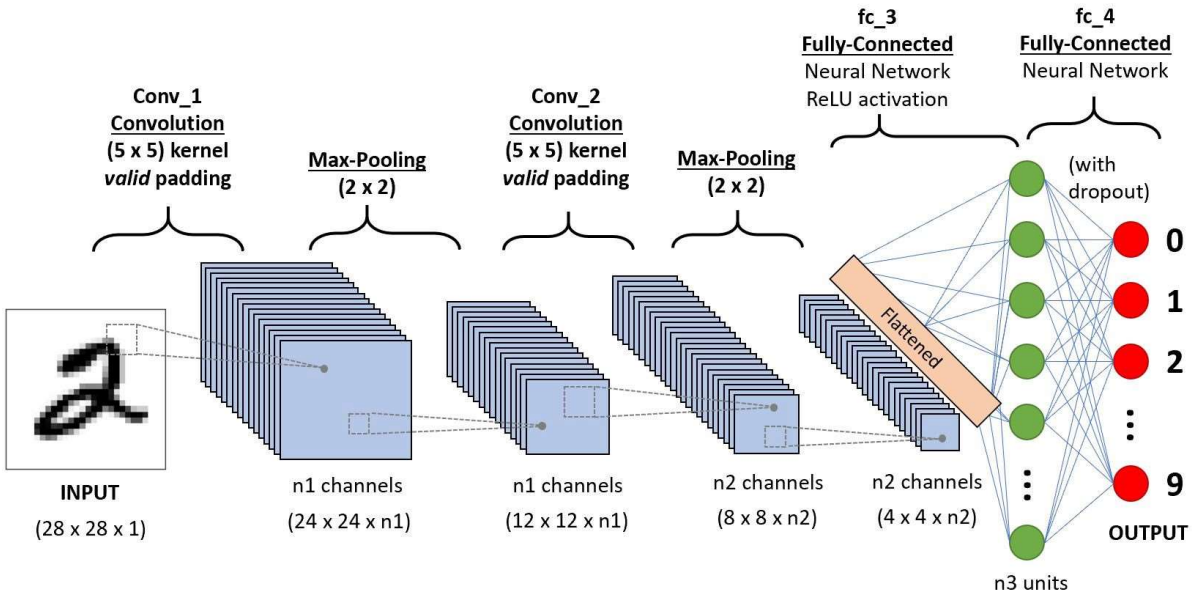
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

We are also doing a comparative study between a customized CNN model and the transfer learning model with VGG16 and we will compare the result which model will be a better fit for predicting the yoga posture. We will discuss about the basic structure of CNN.

What is CNN?

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.



A CNN sequence to classify handwritten digits

A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

So, we build a customized CNN architecture where we will be training our data set and will able to observe the training as well as validation accuracy of our model.

```
model.summary()
```

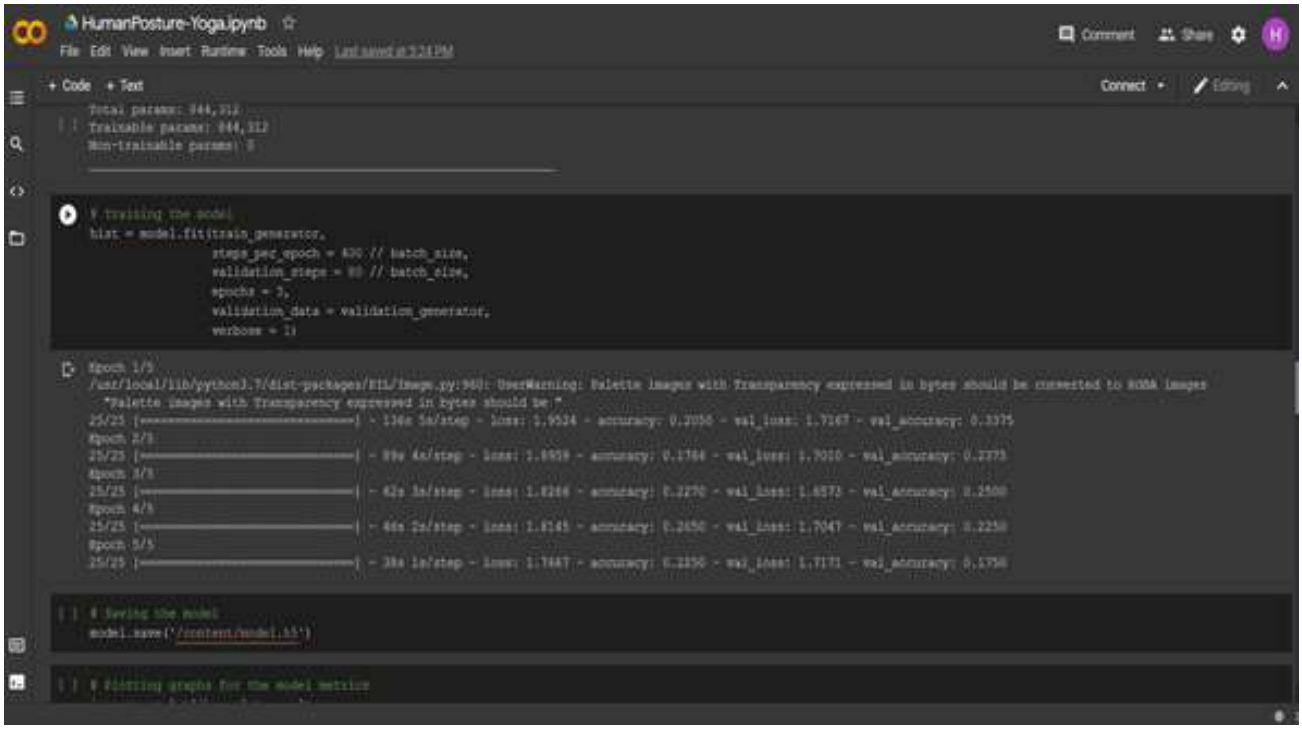
```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_18 (Conv2D)           (None, 253, 253, 8)       80
_________________________________________________________________
activation_24 (Activation)   (None, 253, 253, 8)       0
_________________________________________________________________
average_pooling2d_18 (Averag (None, 84, 84, 8)         0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 82, 82, 16)        1168
_________________________________________________________________
activation_25 (Activation)   (None, 82, 82, 16)        0
_________________________________________________________________
average_pooling2d_19 (Averag (None, 27, 27, 16)        0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 25, 25, 32)        4640
_________________________________________________________________
activation_26 (Activation)   (None, 25, 25, 32)        0
_________________________________________________________________
average_pooling2d_20 (Averag (None, 8, 8, 32)          0
_________________________________________________________________
flatten_6 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_18 (Dense)             (None, 500)               1024500
_________________________________________________________________
dropout_12 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_19 (Dense)             (None, 204)               102204
_________________________________________________________________
activation_27 (Activation)   (None, 204)               0
_________________________________________________________________
dropout_13 (Dropout)         (None, 204)               0
_________________________________________________________________
dense_20 (Dense)             (None, 5)                 1025
=================================================================
Total params: 1,133,617
Trainable params: 1,133,617
Non-trainable params: 0
```

Model Summary of our Customized CNN model

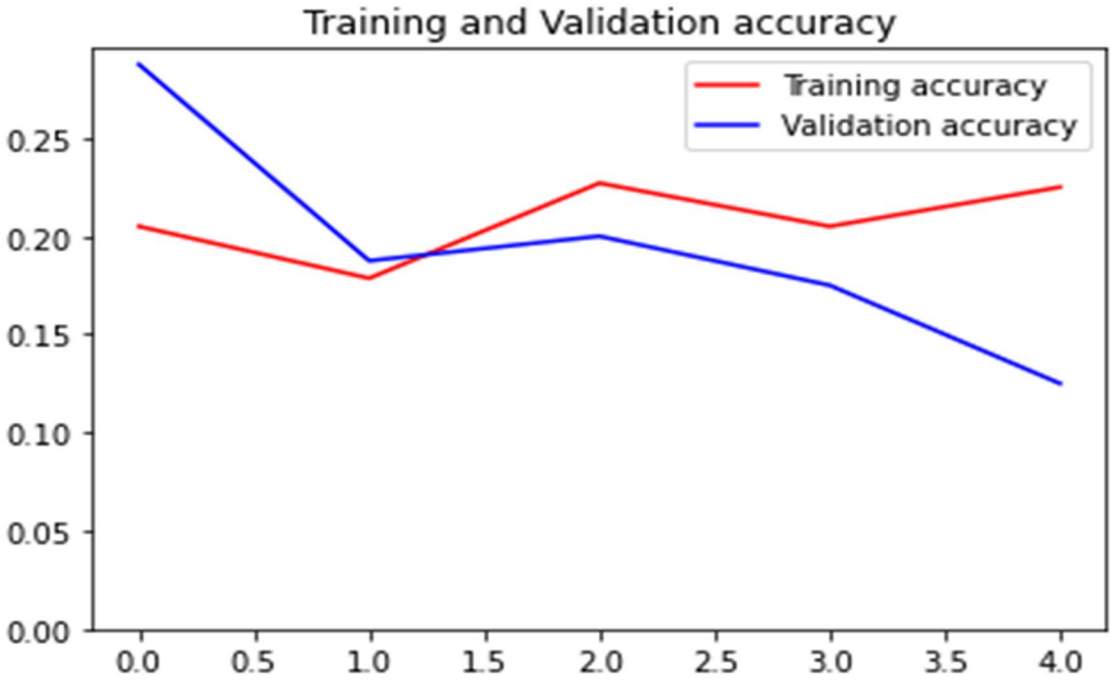
Results:

We did a comparative analysis between a customized CNN model and the transfer learning from pretrained model of VGG16 architecture. As we train our data in normal CNN model, we get the validation loss is 1.7667 and accuracy is 0.1750 which shows that the model predicting the yoga pose is worse. The same dataset when trained on transfer learning model on VGG16 model, we got the validation loss is 0.8759 and accuracy is 0.7312. As we can see that we are getting better accuracy score in transfer learning model than in customized CNN model.

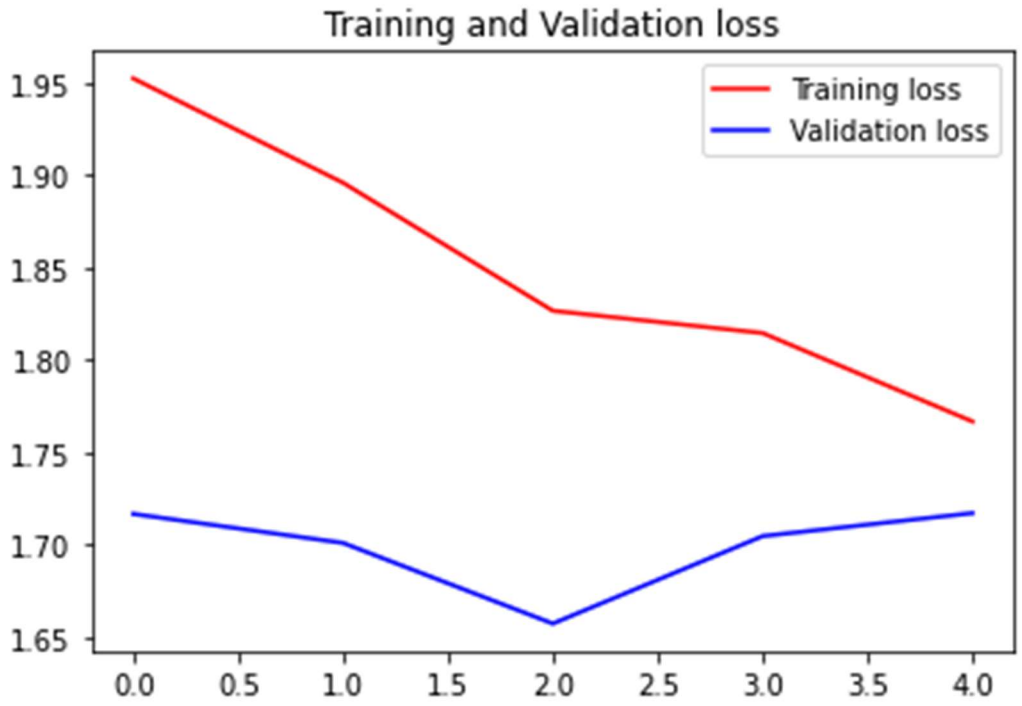
The results are shows in a pictorial representation for both on customized CNN architecture as well as the transfer learning model of VGG16. Through this we can have a clarity on the model simulation which is done in Google Collab using GPU processor.



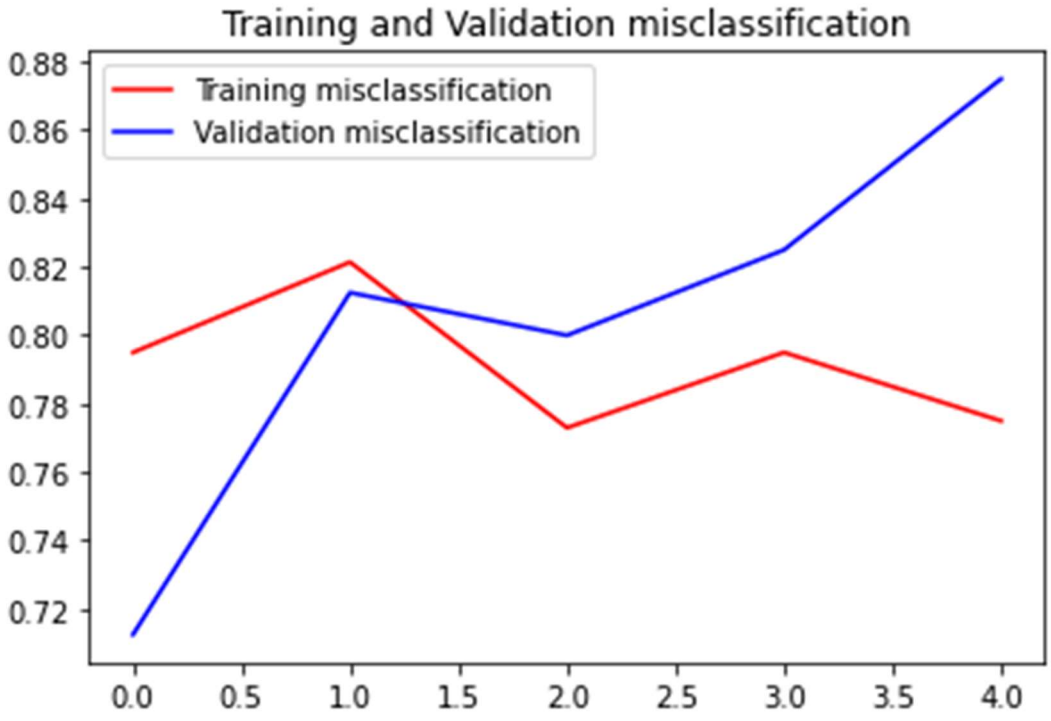
Accuracy and loss score of customized CNN model from Google collab



Accuracy Graph of Training and Validation of Customized CNN

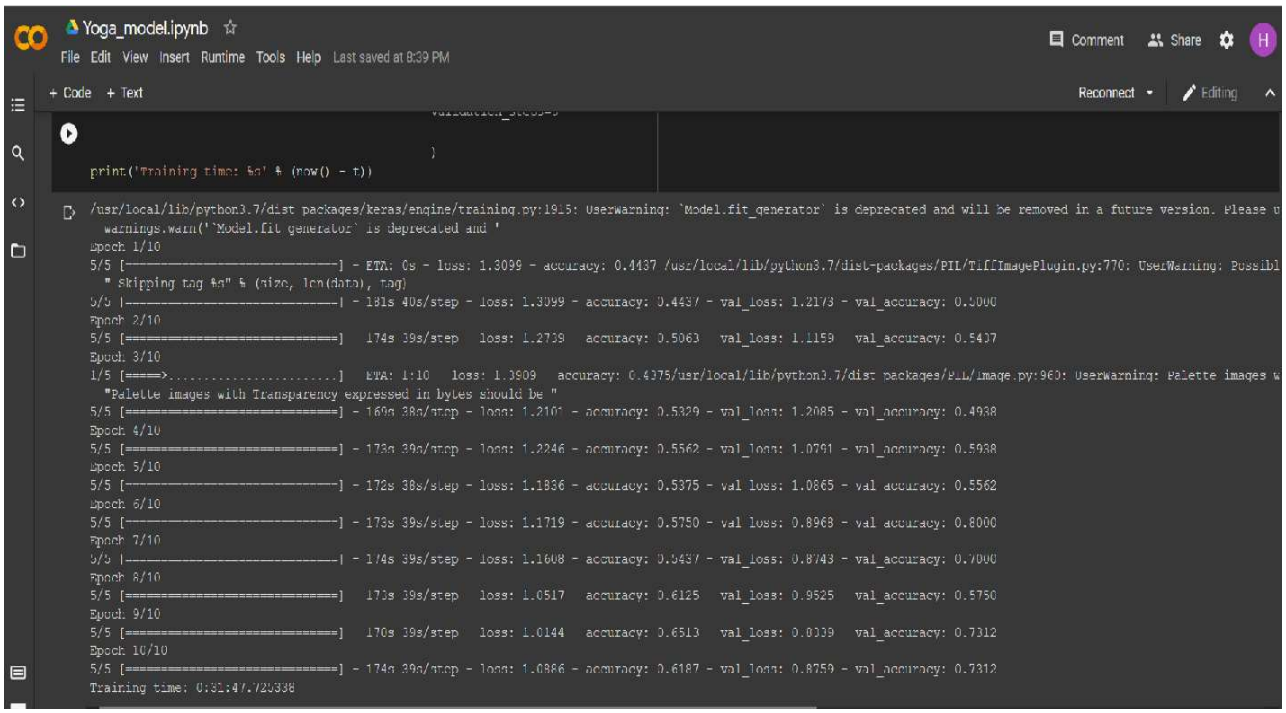


Loss Graph of Training and Validation of Customized CNN

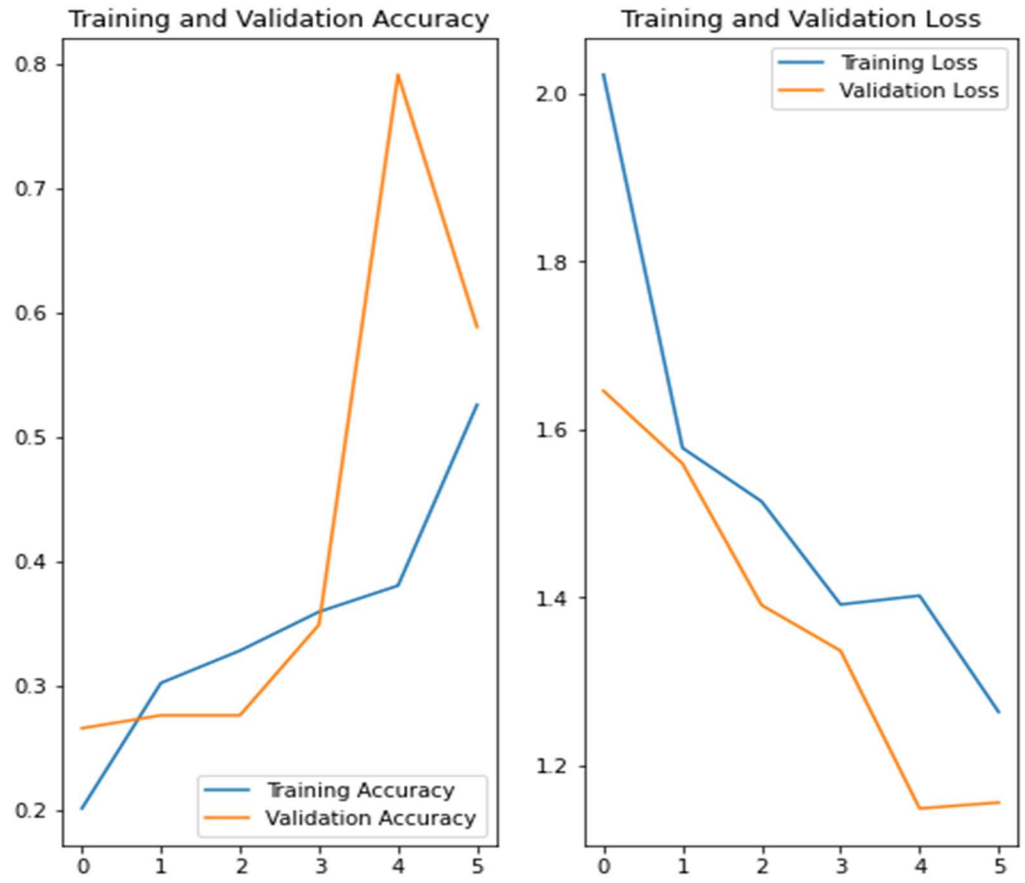


Misclassification Graph of Training and Validation of Customized CNN

The Results of Transfer learning using VGG16 model



Accuracy and loss score of Transfer learning model using VGG16 from Google collab



Accuracy and Loss Graph of Training and Validation of Transfer learning model using VGG16

Future Scope:

The whole process of this project concludes that the transfer learning technique using pretrained models gives better performance in predicting the posture of yoga in the images than the customized CNN model. The performance will be improved even more when the size of the data set will be much huge than that of the dataset that we used in our project.

Similarly, to yoga posture detection, transfer learning model can also be used in various applications which involves the image detection in CCTV or in the domain of NLP as well. It reduces some of the effort to initialize the weights as it borrows from pretrained model and perform well. It tackle problems like having little or almost no labeled data availability.

Transfer learning is definitely going to be one of the key drivers for machine learning and deep learning success in mainstream adoption in the industry. I definitely hope to see more pre-trained models and innovative case studies which leverage this concept and methodology.

Reference:

1. https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a
2. https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7
3. W. Gong, X. Zhang, J. Gonzàlez, A. Sobral, T. Bouwmans, C. Tu, and H. Zahzah, "Human pose estimation from monocular images: a comprehensive survey", *Sensors*, Basel, Switzerland, vol. 16, 2016.
4. Kothari, Shruti, "Yoga Pose Classification Using Deep Learning" (2020). Master's Projects. 932.
   DOI: https://doi.org/10.31979/etd.rkgu-pc9k
5. https://neurohive.io/en/popular-networks/vgg16/