PROJECT REPORT

ON

TOPIC OF PROJECT

"YOGA POSTURE CLASSIFICATION USING KERAS"

Submitted by: Jayesh Sukdeo Patil

(MSc Computer Science Part-1 (SEM-2))

Academic Year: 2021-2022

Project Guide: Dr.Rajeshri Shinkar

SIES (Nerul) College of Arts, Science and Commerce

NAAC Re-Accredited 'A' Grade

Sri Chandrasekarendra Saraswathy Vidyapuram,

Plot 1-C, Sector V,

Nerul, Navi Mumbai-400 706.

Certificate

This is to certify that the Project entitled "**YOGA POSTURE CLASSIFICATION USING KERAS**" successfully completed by **JAYESH SUKDEO PATIL** of Part-1(Sem-1) Masters in Science (Computer Science) as per the requirement of University of Mumbai in part fulfillment for the completion of PG Degree of Master of Science (Computer Science). It is also to certify that this is the original work of the candidate done during the academic year 2021-2022.

Roll No: **24** **Date of Submission: 30-04-2022**

Subject:

Prof.Rajeshri Shinkar Date: 30-04-2022

(Project Guide)

INDEX

Abstract:

As today's world is getting aware about the importance of yoga in their daily routine to help them keep healthy and fit. as there are many different types of postures Many people don’t know the names and correct postures of yoga asanas and gets confused among them. So, this project can help to identify the correct name and postures for yoga practioners This project focuses on identifying different yoga postures using keras classification libraries. Keras includes a specialized Image Classification model called InceptionV3 made using CNN (Convolution Neural Network). Inception v3 is an image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset.

Introduction:

Health is the major concern of each and every individual. Being fit both physically and mentally is not an easy task. Yoga and meditation are considered as an optimal solution for the same Asana is defined as “posture or pose;” its literal meaning is “seat.” Originally, there was only one asana–a stable and comfortable pose for prolonged seated meditation. More than just stretching and toning the physical body, the yoga poses open the nadis (energy channels) and chakras (Psychic centers) of the body. Yoga poses also purify and help heal the body, as well as control, calm and focus the mind.

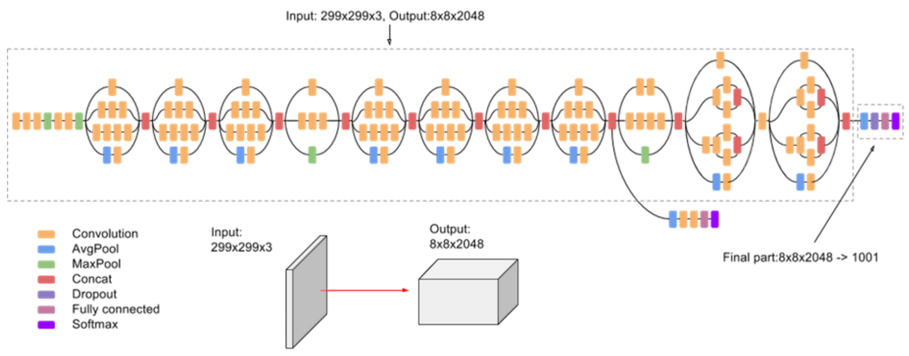
Neural Network:

1. Classification is a type of supervised machine learning algorithm used to predict a categorical label. A few useful examples of classification include predicting whether a customer will churn or not, classifying emails into spam or not, or whether a bank loan will default or not. The basic architecture of the deep learning neural network, Input Layer: This is where the training observations are fed. The number of predictor variables is also specified here through the neurons.
2. Hidden Layers: These are the intermediate layers between the input and output layers. The deep neural network learns about the relationships involved in data in this component.

3. Output Layer: This is the layer where the final output is extracted from what's happening in the previous two layers. In case of regression problems, the output layer will have one neuron.

InceptionV3 (CNN)

The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using SoftMax.



CNN: A convolutional neural network is a specific kind of neural network with multiple layers. It processes data that has a grid-like arrangement then extracts important features.
Convolutional layers represent the presence of features in an input image. But they present a problem, they're sensitive to location of features in the input. This gives us specific data rather than generalized data, deepening the problem of overfitting and doesn't deliver good results for data outside the training set.
Pooling: To generalize the presence of features. This is done by means of pooling layers.
Concat: It takes as input a list of tensors, all of the same shape except for the concatenation axis, and returns a single tensor that is the concatenation of all inputs.
Dropout: refers to data, or noise, that's intentionally dropped from a neural network to improve processing and time to results.
SoftMax: It is mainly used to normalize neural networks output to fit between zero and one. It is used to represent the certainty "probability" in the network output.

Inception is a convolutional neural network architecture introduced by Google which achieved top results in ImageNet Large Scale Visual Recognition Challenge 2014. This function returns a Keras image classification model, optionally loaded with weights pre-trained on ImageNet. Inception-v3 is a convolutional neural network that is 48 layers deep.

Features of InceptionV3: -

Batch Normalization in the fully connected layer of Auxiliary classifier.
Use of 7×7 factorized Convolution
Label Smoothing Regularization: It is a method to regularize the classifier by estimating the effect of label-dropout during training. It prevents the classifier to predict a class too confidently. The addition of label smoothing gives 0.2% improvement from the error rate

Objectives:

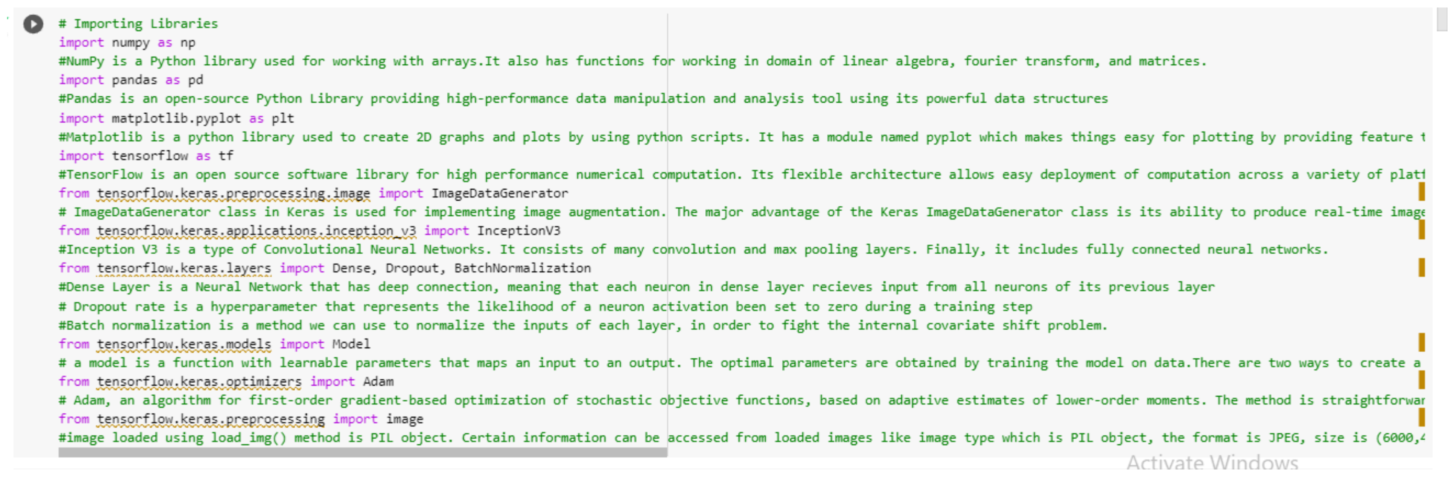
To make a model that can help yoga practioner identify yoga posture along with their names.
To increase the performance using yoga assistant kit with prediction intelligence which will assist the person to perform suitable yoga postures.

Methodology:

- Yoga Pose Image datasets. This dataset contains 5993 images and 107 different yoga asanas. using pre-trained model, which can be done using Transfer Learning. The reuse of a previously learned model on a new problem is known as transfer learning.

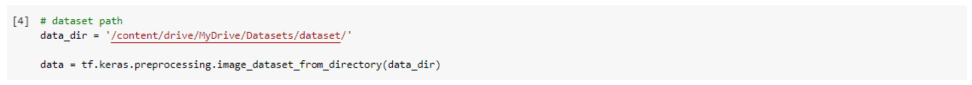
Following Are the Steps Followed for Implementation of Model:

Loading The Required Libraries And Modules



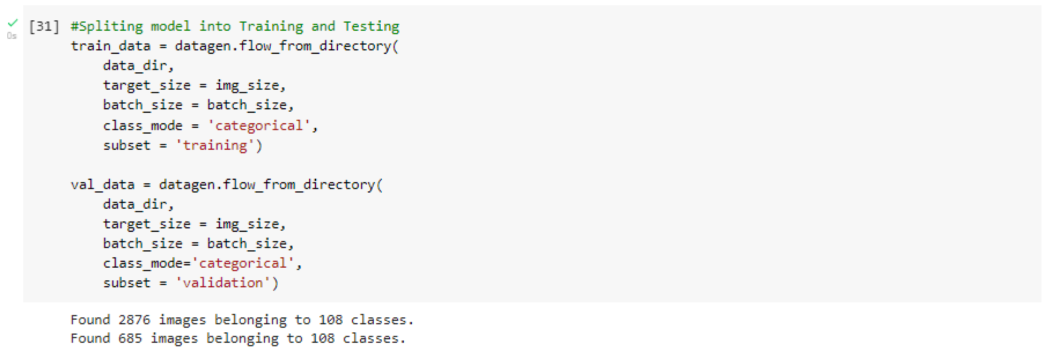
```
# Importing Libraries
import numpy as np
#NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices.
import pandas as pd
#Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures
import matplotlib.pyplot as plt
#Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature t
import tensorflow as tf
#TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# ImageDataGenerator class in Keras is used for implementing image augmentation. The major advantage of the Keras ImageDataGenerator class is its ability to produce real-time image
from tensorflow.keras.applications.inception_v3 import InceptionV3
#Inception V3 is a type of Convolutional Neural Networks. It consists of many convolution and max pooling layers. Finally, it includes fully connected neural networks.
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
#Dense Layer is a Neural Network that has deep connection, meaning that each neuron in dense layer recieves input from all neurons of its previous layer
# Dropout rate is a hyperparameter that represents the likelihood of a neuron activation been set to zero during a training step
#Batch normalization is a method we can use to normalize the inputs of each layer, in order to fight the internal covariate shift problem.
from tensorflow.keras.models import Model
# a model is a function with learnable parameters that maps an input to an output. The optimal parameters are obtained by training the model on data.There are two ways to create a
from tensorflow.keras.optimizers import Adam
# Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforwar
from tensorflow.keras.preprocessing import image
#image loaded using load_img() method is PIL object. Certain information can be accessed from loaded images like image type which is PIL object, the format is JPEG, size is (6000,4
```

Loading The Data And Giving Directory Path



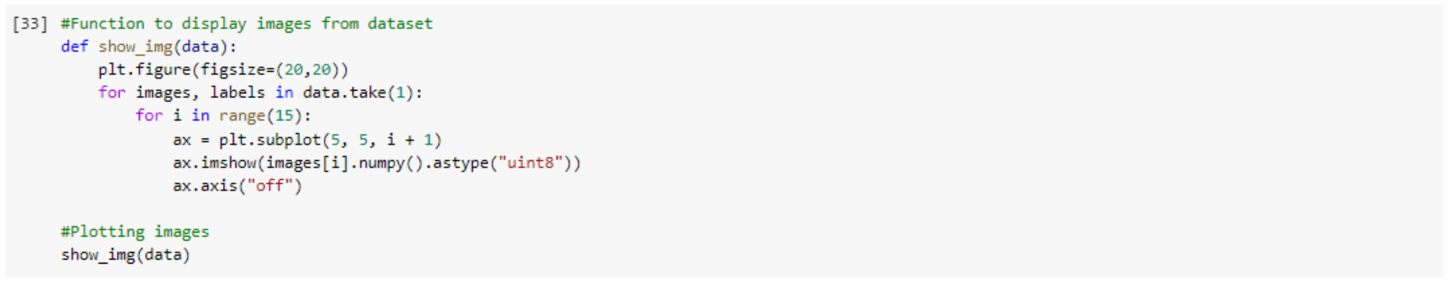
```
[4] # dataset path
    data_dir = '/content/drive/MyDrive/Datasets/dataset/'

    data = tf.keras.preprocessing.image_dataset_from_directory(data_dir)
```

Building Model



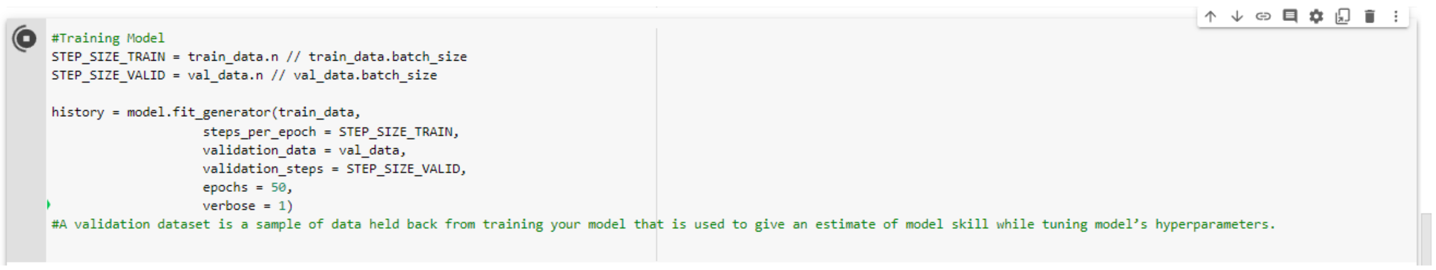
```
[31] #Spliting model into Training and Testing
     train_data = datagen.flow_from_directory(
         data_dir,
         target_size = img_size,
         batch_size = batch_size,
         class_mode = 'categorical',
         subset = 'training')

     val_data = datagen.flow_from_directory(
         data_dir,
         target_size = img_size,
         batch_size = batch_size,
         class_mode='categorical',
         subset = 'validation')

     Found 2876 images belonging to 108 classes.
     Found 685 images belonging to 108 classes.
```

Visualize Images



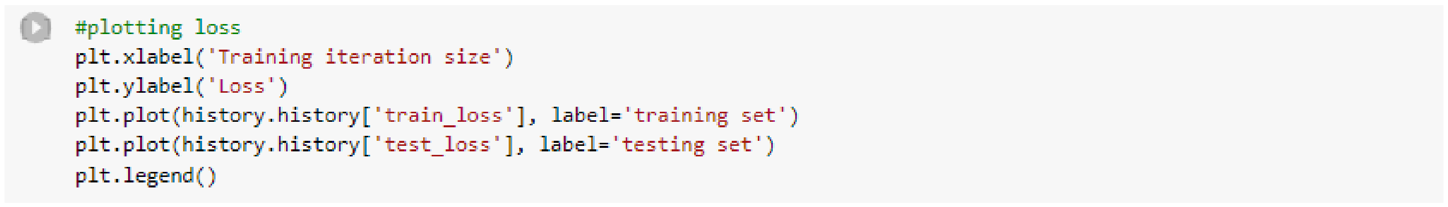
```
[33] #Function to display images from dataset
     def show_img(data):
         plt.figure(figsize=(20,20))
         for images, labels in data.take(1):
             for i in range(15):
                 ax = plt.subplot(5, 5, i + 1)
                 ax.imshow(images[i].numpy().astype("uint8"))
                 ax.axis("off")

     #Plotting images
     show_img(data)
```

Train The Model

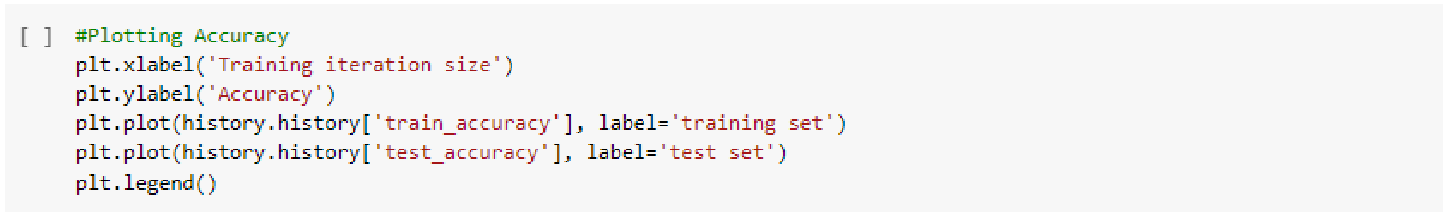


```
#Training Model
STEP_SIZE_TRAIN = train_data.n // train_data.batch_size
STEP_SIZE_VALID = val_data.n // val_data.batch_size

history = model.fit_generator(train_data,
                    steps_per_epoch = STEP_SIZE_TRAIN,
                    validation_data = val_data,
                    validation_steps = STEP_SIZE_VALID,
                    epochs = 50,
                    verbose = 1)
#A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model’s hyperparameters.
```

Plotting Accuracy and loss



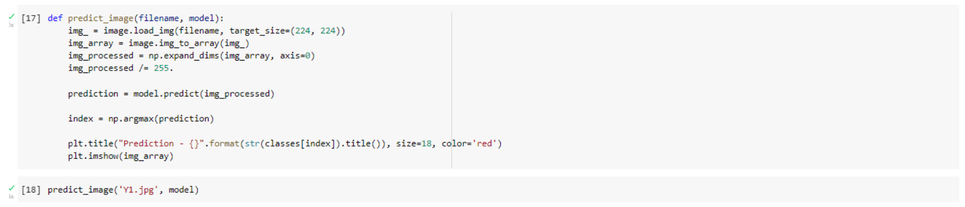
```
#plotting loss
plt.xlabel('Training iteration size')
plt.ylabel('Loss')
plt.plot(history.history['train_loss'], label='training set')
plt.plot(history.history['test_loss'], label='testing set')
plt.legend()
```



```
[ ] #Plotting Accuracy
plt.xlabel('Training iteration size')
plt.ylabel('Accuracy')
plt.plot(history.history['train_accuracy'], label='training set')
plt.plot(history.history['test_accuracy'], label='test set')
plt.legend()
```

A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model’s hyperparameters.

Predict Using New Data

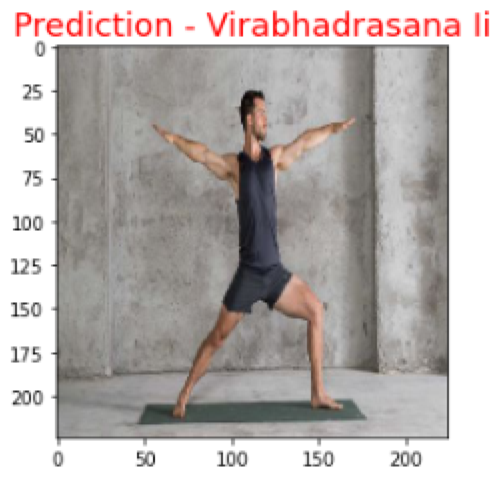


```
[17] def predict_image(filename, model):
         img_ = image.load_img(filename, target_size=(224, 224))
         img_array = image.img_to_array(img_)
         img_processed = np.expand_dims(img_array, axis=0)
         img_processed /= 255.

         prediction = model.predict(img_processed)

         index = np.argmax(prediction)

         plt.title("Prediction - {}".format(str(classes[index]).title()), size=18, color='red')
         plt.imshow(img_array)
```

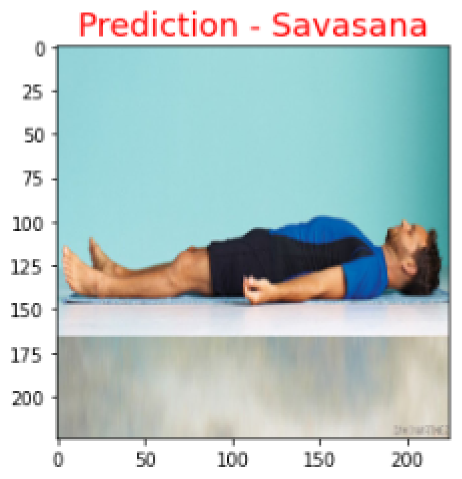
```
[18] predict_image('Y1.jpg', model)
```

Results:

```
[ ]  predict_image('Y1.jpg', model)
```



```
[ ]  predict_image('Y3.jpg', model)
```



Limitation & Future Scope:

The current model lacks Real Time Monitoring of Yoga Postures. In future this model can help to predict postures in real time to help performers
Using IOT this model can be useful for performers to analyze their yoga routine like today we have Different modes in Smart fit bands showing their activity same way it can include as a new function in bands helping the yoga performers.

Conclusion:

The model can successfully predict yoga postures using inception v3 pretrained model. It can help yoga practioner to identify yoga postures easily. With overcoming present limitations this model can be upgraded to make a full fledge user friendly application.

Appendix:

Below is the link for colab file:

https://colab.research.google.com/drive/1gkkP4fjyMwP2LQEUMWFcAe0sAiheeRDj?usp=sharing

References:

- “**Yoga Pose Classification Using Deep Learning**” -Shruti Kothari San Jose State University https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1932&context=etd_projects
- 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.
- **“Prediction Intelligence System Based Real Time Monitoring of Yoga Performers” -**Prasanna M, Arunkumar T, Arun Kumar S , Nazir M.I.J. Published in Bentham Science Publishers Ltd. 2019