YOGA POSE CLASSIFICATION USING DEEP LEARNING WITH IBM WATSON STUDIO

1. INTRODUCTION

1.1 Overview

Yoga is a 5000-year-old practice developed in ancient India by the Indus-Sarasvati civilization. The word yoga means deep association and union of mind with the body. It is used to keep both mind and body in equilibration in all flip-flops of life by means of asana, meditation, and several other techniques. Nowadays, yoga has gained worldwide attention due to increased stress levels in the modern lifestyle, and there are numerous methods or resources for learning yoga. Yoga can be practiced in yoga centers, through personal tutors, and can also be learned on one’s own with the help of the Internet, books, recorded clips, etc.

1.2 Purpose

In this Project Deep Learning based techniques are developed to detect yoga pose by uploading an image or by real time video using computer vision technique. It detects the pose and tells exactly the pose name.

2. LITERATURE SURVEY

2.1 Existing problem

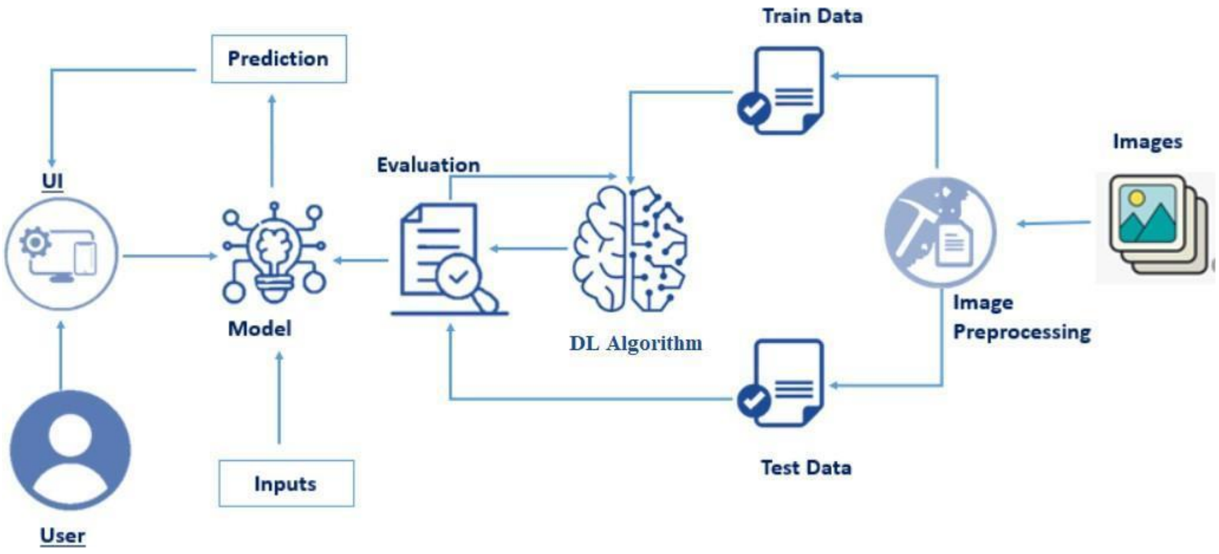
In fast-paced lifestyles, many people prefer self-learning because the above mentioned resources might not be available all the time. But in self-learning, one may not find an incorrect pose. Incorrect posture can be harmful to one’s health, resulting in acute pain and long-term chronic concerns.

2.2 Proposed solution

Incorrect posture can be harmful to one’s health, resulting in acute pain and long-term chronic concerns. In this Project Deep Learning based techniques are developed to detect yoga pose by uploading an image or by real time video using computer vision technique. It detects the pose and tells exactly the pose name.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

Software Requirements:

- Anaconda Navigator
- Tensor flow
- Keras
- Flask

Hardware Requirements:

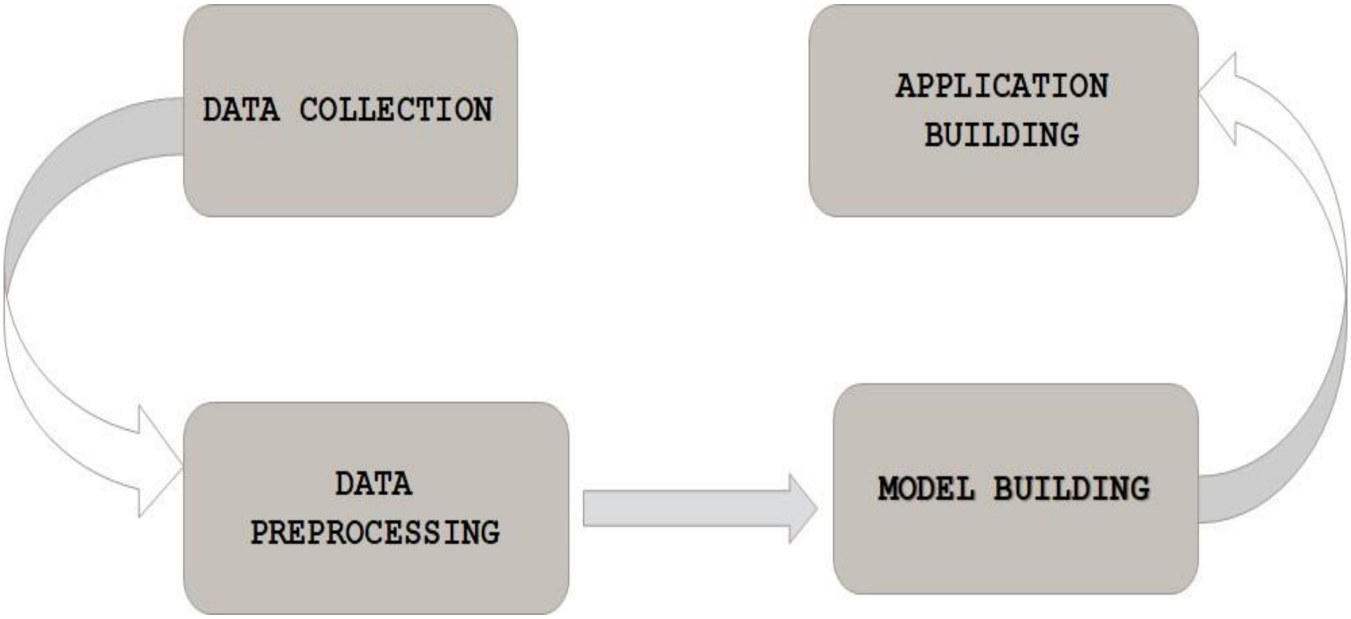
- Processor : Intel Core i3
- Hard Disk Space : Min 100 GB
- Ram : 4 GB
- Display : 14.1 "Color Monitor(LCD, CRT or LED)

Clock Speed : 1.67 GHz

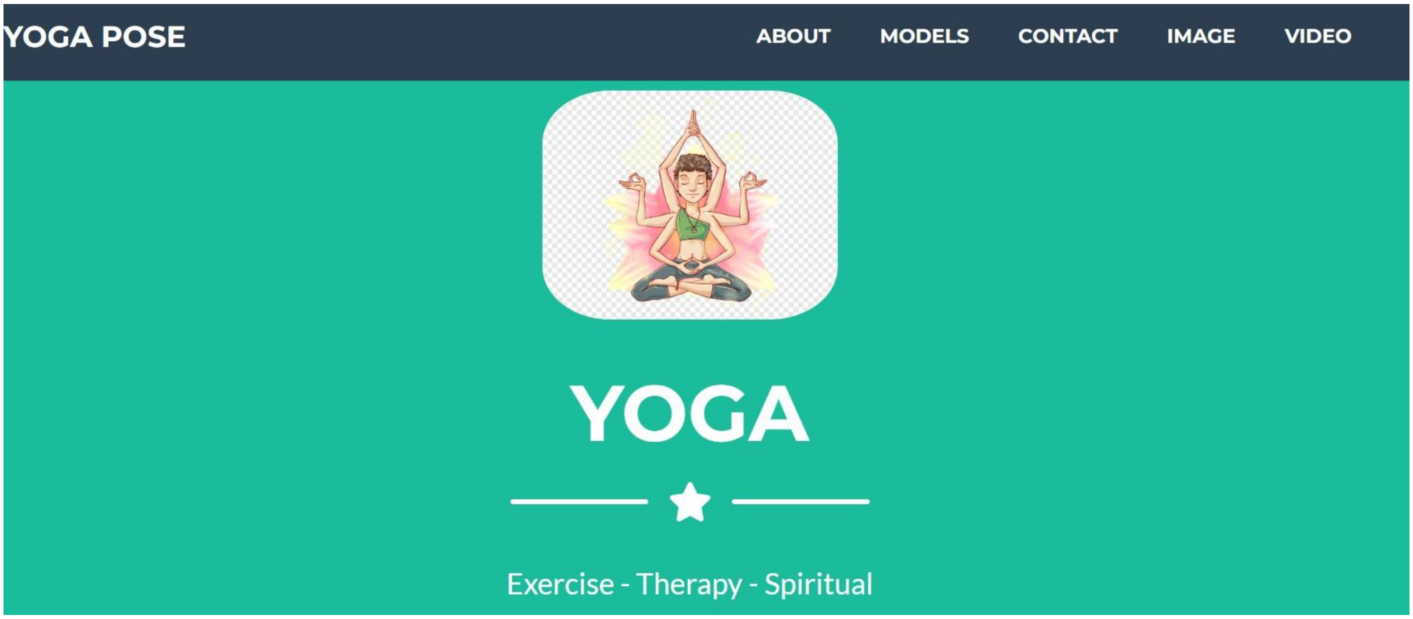
4. EXPERIMENTAL INVESTIGATIONS

Study shows that it provide with different test images of yoga postures, the model detects, if there is any yoga posture in the video stream. If the posture is detects the pose and tells exactly the pose name. When we choose an image and click in to the upload it then it will shows the predicted output.

5. FLOWCHART

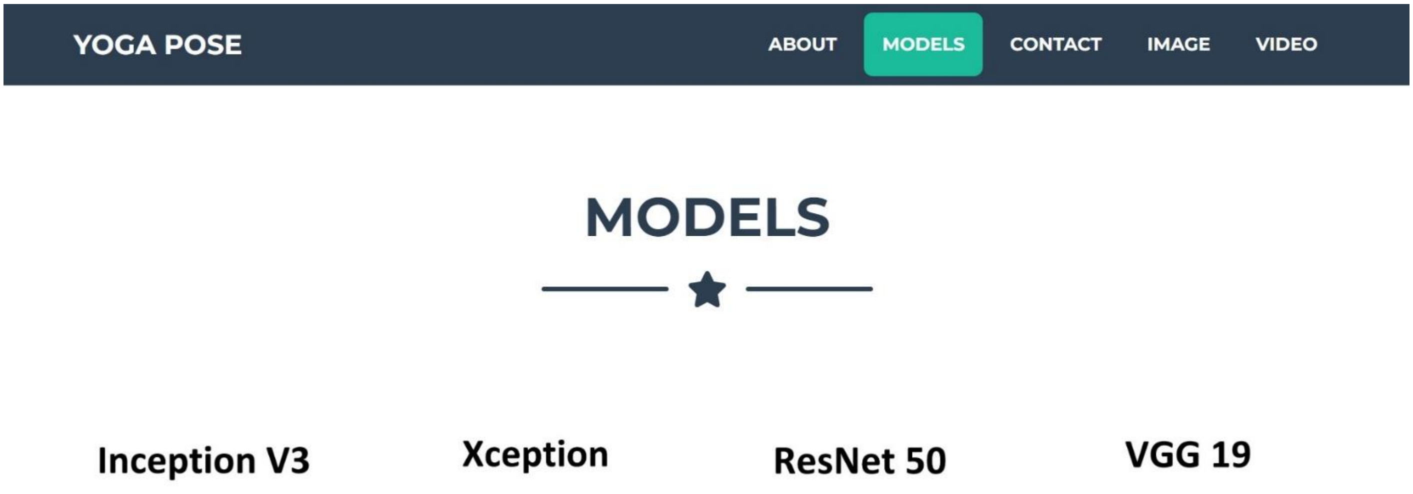


6. RESULT



YOGA

Exercise - Therapy - Spiritual



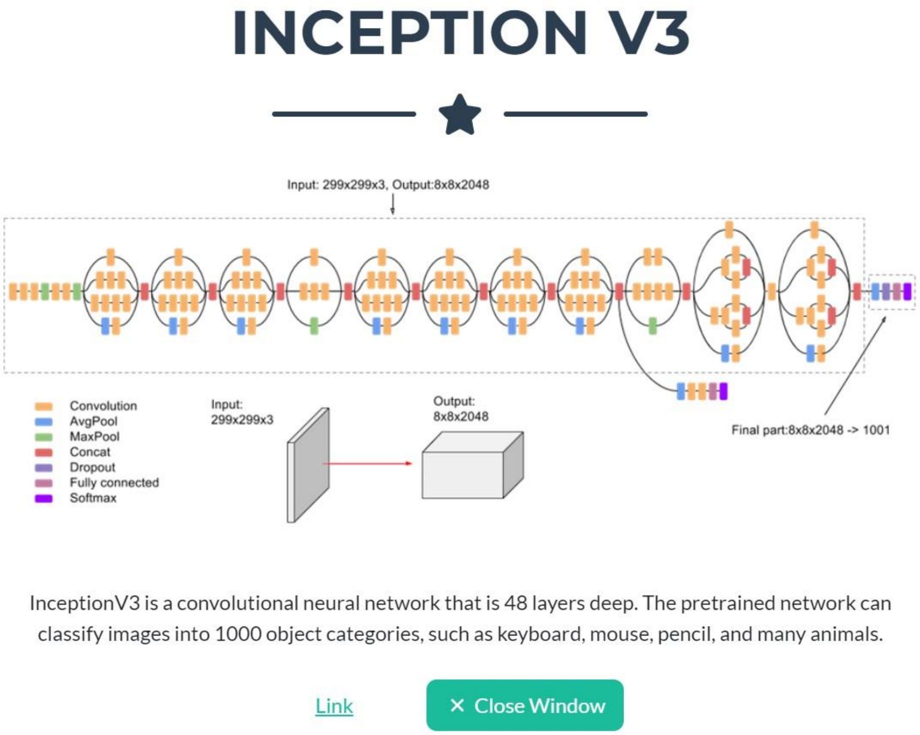
MODELS

Inception V3

Xception

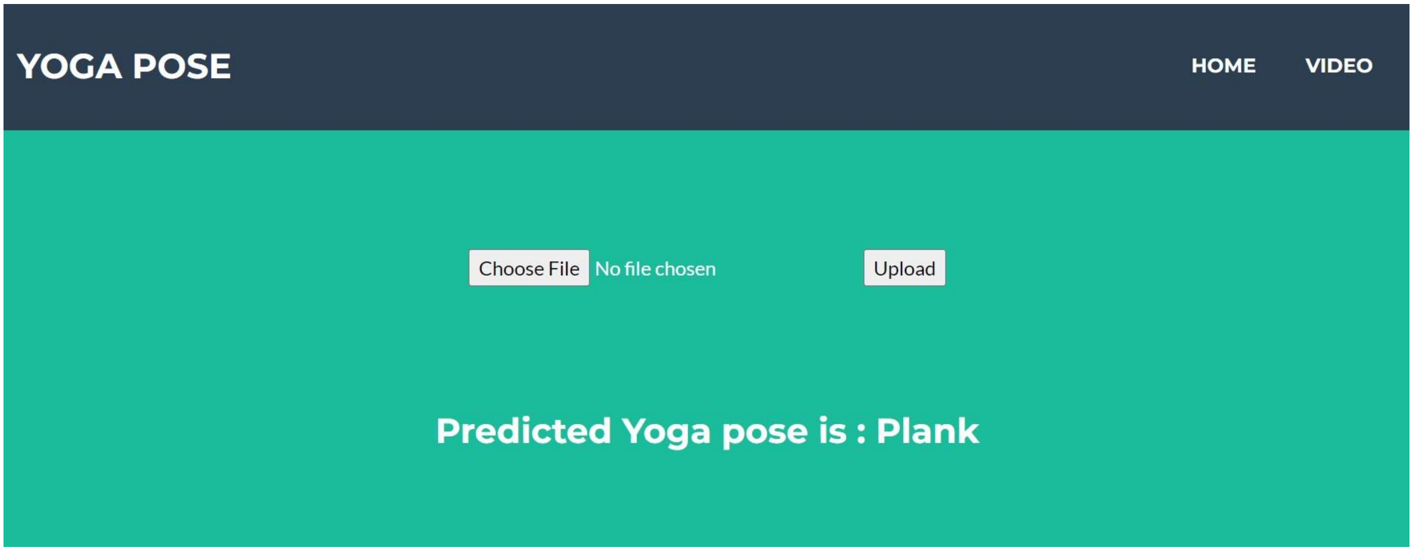
ResNet 50

VGG 19



INCEPTION V3

InceptionV3 is a convolutional neural network that is 48 layers deep. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.



Predicted Yoga pose is : Plank

7. ADVANTAGES & DISADVANTAGES

Advantages:

- The goal of this project is to improve your flexibility, blood circulation, energy and even self-esteem.
- It releases tension and promote emotional growth.

Disadvantages:

- Data mining techniques does not help to provide effective decision making.

8. APPLICATIONS

- Deep Learning technology is considered as one of the key technology used in Yoga pose classification.
- It presents the results obtained by processing input from a camera in real time.

9. CONCLUSION

In this project, we have established the application to predict yoga pose based on the IBM cloud application. Human posture assessment has been concentrated widely over the previous year. This project is very helpful to reduce the human physical and mental diseases.

10. FUTURE SCOPE

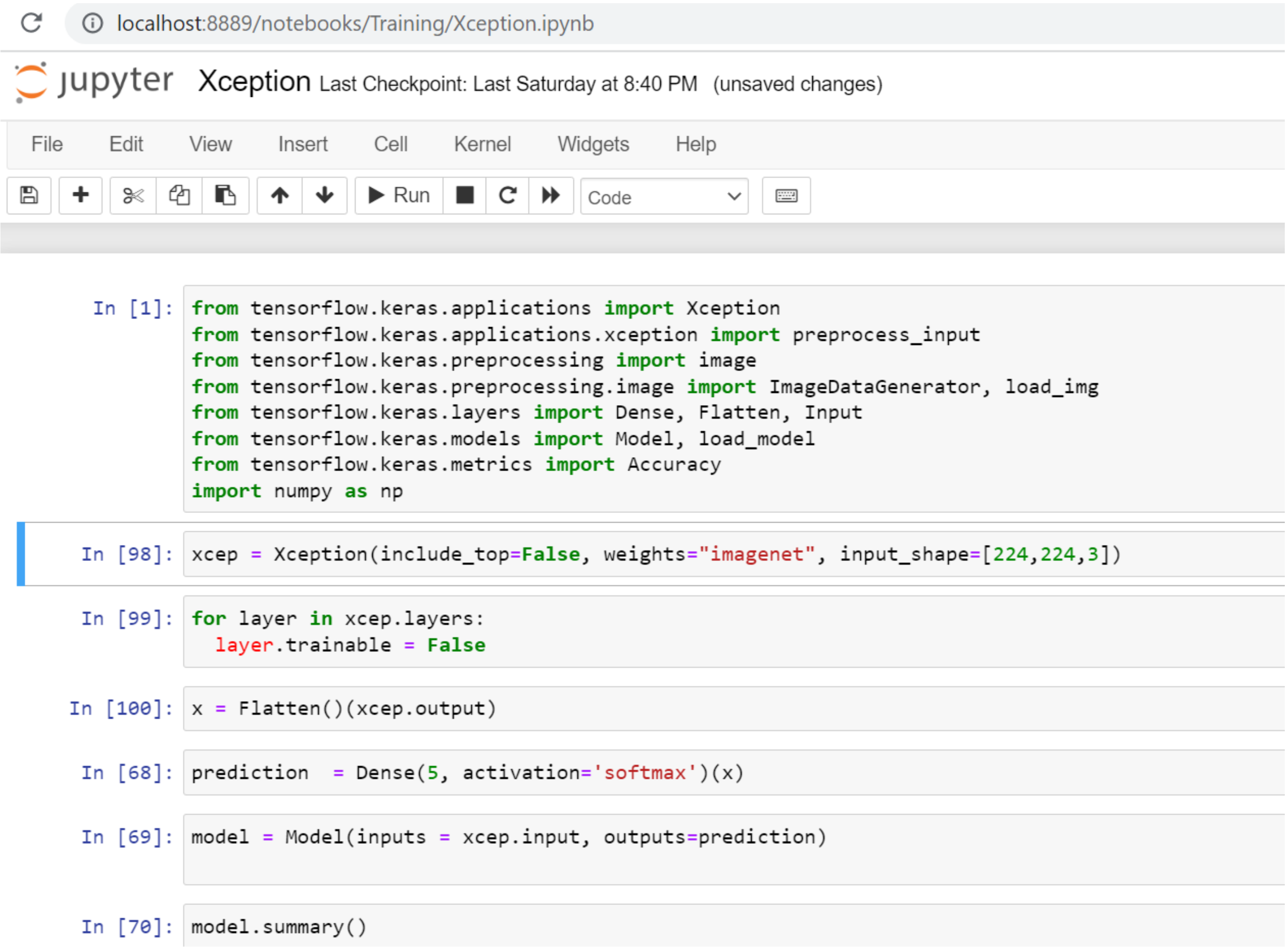
The proposed models right now characterize the dataset into 5. There are various yoga classifications and subsequently making a posture assessment model that can be effective for human. The dataset can be extended by adding more yoga poses.

11. BIBILOGRAPHY

- L. Signal “Human pose Estimation”, Ency .of Comput Vision, Springer 2011
- Kothari, Shruthi, “Yoga pose classification using deep learning”, 2020

APPENDIX

Source Code



localhost:8889/notebooks/Training/Xception.ipynb
jupyter
Xception Last Checkpoint: Last Saturday at 8:40 PM (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
In [1]: from tensorflow.keras.applications import Xception
        from tensorflow.keras.applications.xception import preprocess_input
        from tensorflow.keras.preprocessing import image
        from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
        from tensorflow.keras.layers import Dense, Flatten, Input
        from tensorflow.keras.models import Model, load_model
        from tensorflow.keras.metrics import Accuracy
        import numpy as np
```

```
In [98]: xcep = Xception(include_top=False, weights="imagenet", input_shape=[224,224,3])
```

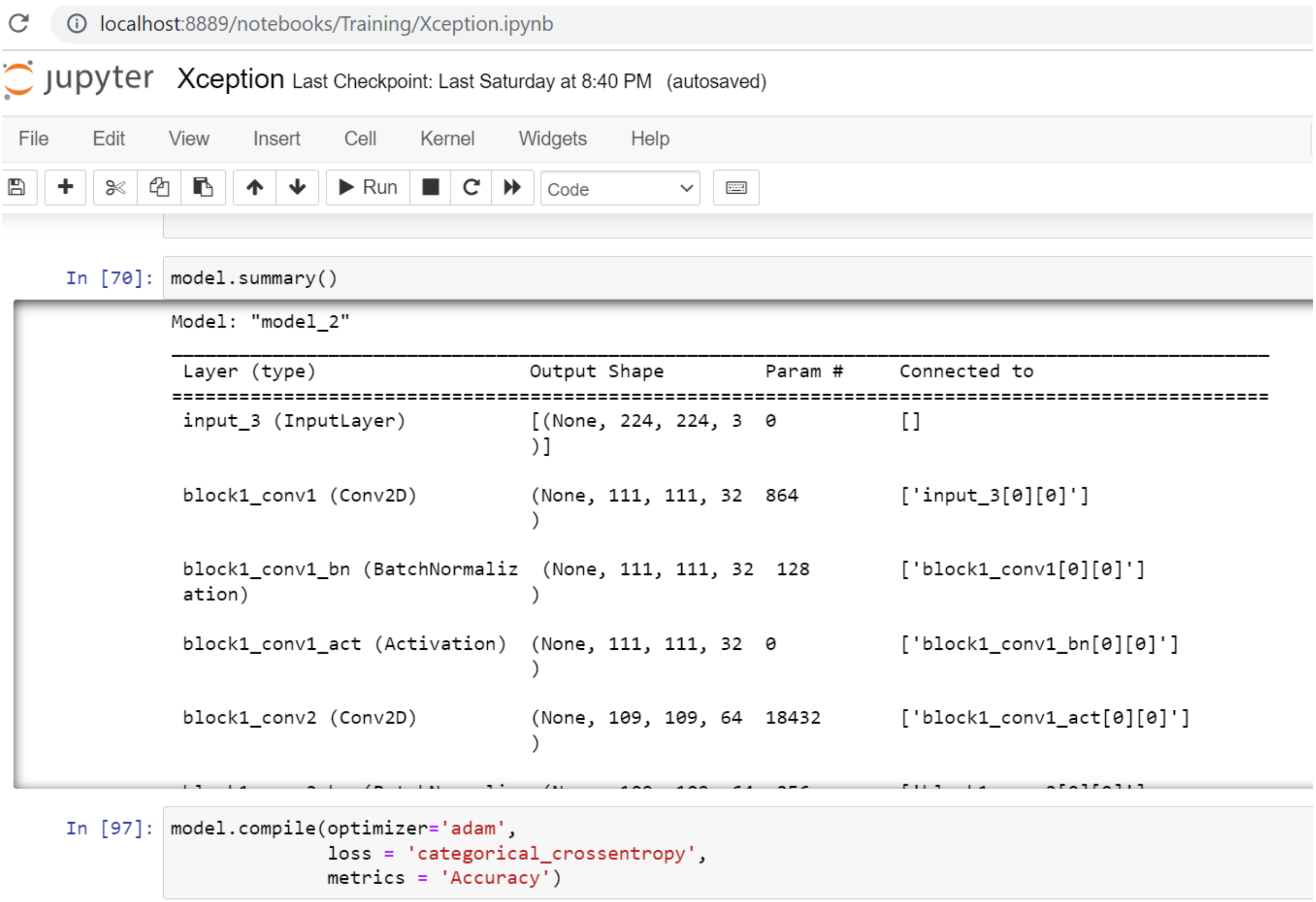
```
In [99]: for layer in xcep.layers:
           layer.trainable = False
```

```
In [100]: x = Flatten()(xcep.output)
```

```
In [68]: prediction  = Dense(5, activation='softmax')(x)
```

```
In [69]: model = Model(inputs = xcep.input, outputs=prediction)
```

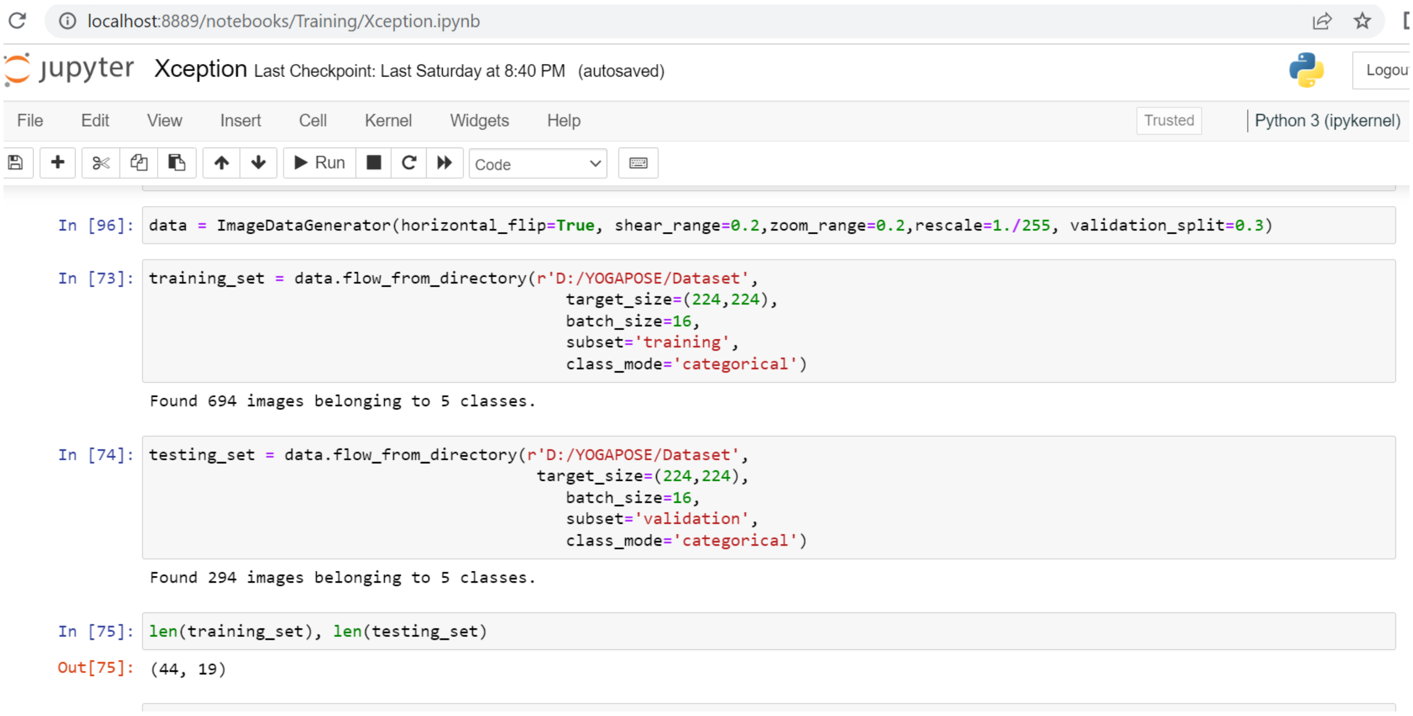
```
In [70]: model.summary()
```


localhost:8889/notebooks/Training/Xception.ipynb
jupyter Xception Last Checkpoint: Last Saturday at 8:40 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run
Code

In [70]:
```
model.summary()
```

```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 111, 111, 32  864         ['input_3[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz  (None, 111, 111, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 111, 111, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 109, 109, 64  18432       ['block1_conv1_act[0][0]']
                                )

 block1_conv2_bn (BatchNormaliz  (None, 109, 109, 64  256        ['block1_conv2[0][0]']
```

In [97]:
```
model.compile(optimizer='adam',
              loss = 'categorical_crossentropy',
              metrics = 'Accuracy')
```


localhost:8889/notebooks/Training/Xception.ipynb
jupyter Xception Last Checkpoint: Last Saturday at 8:40 PM (autosaved)
Logou
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [96]:
```
data = ImageDataGenerator(horizontal_flip=True, shear_range=0.2,zoom_range=0.2,rescale=1./255, validation_split=0.3)
```

In [73]:
```
training_set = data.flow_from_directory(r'D:/YOGAPOSE/Dataset',
                                           target_size=(224,224),
                                           batch_size=16,
                                           subset='training',
                                           class_mode='categorical')
```

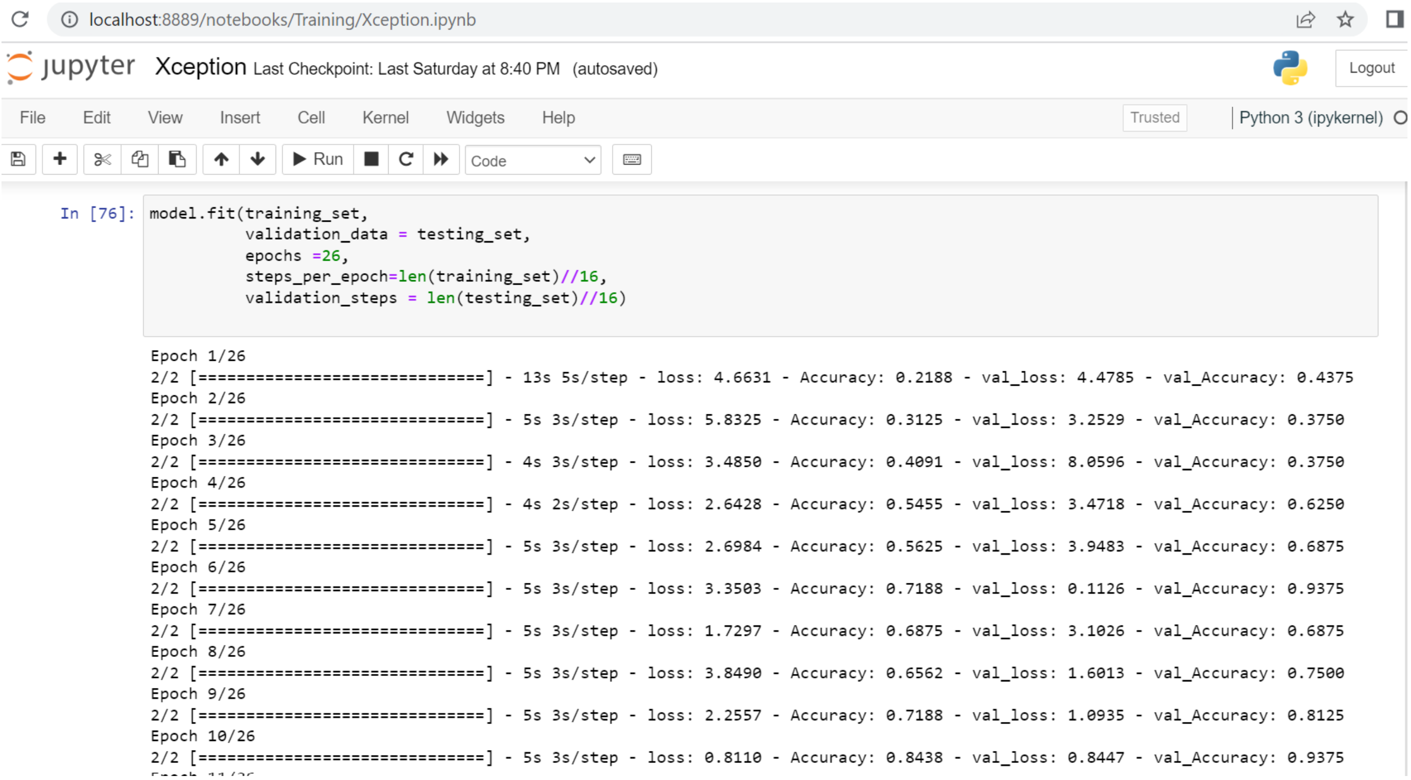
```
Found 694 images belonging to 5 classes.
```

In [74]:
```
testing_set = data.flow_from_directory(r'D:/YOGAPOSE/Dataset',
                                        target_size=(224,224),
                                           batch_size=16,
                                           subset='validation',
                                           class_mode='categorical')
```

```
Found 294 images belonging to 5 classes.
```

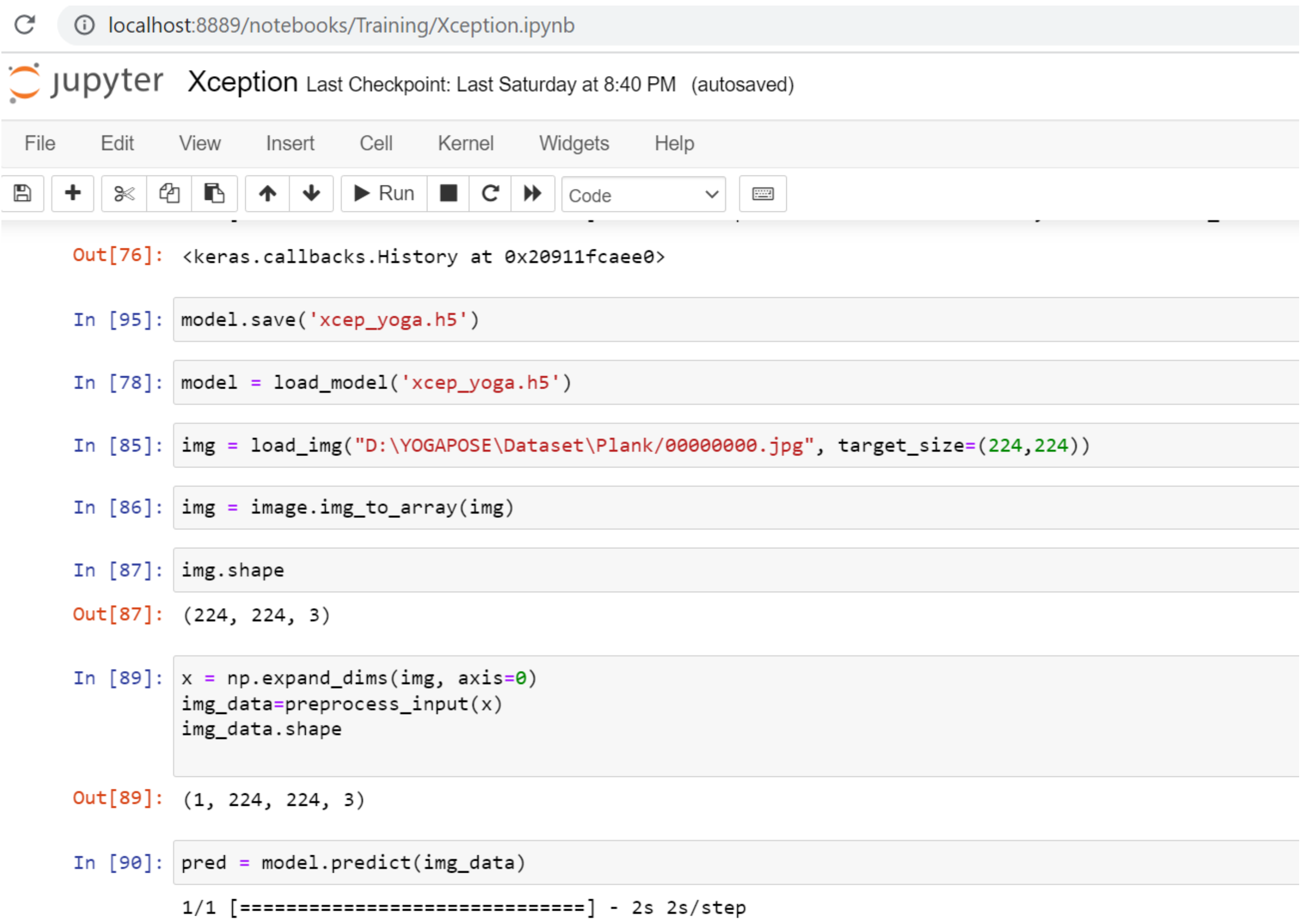
In [75]:
```
len(training_set), len(testing_set)
```

Out[75]:
```
(44, 19)
```

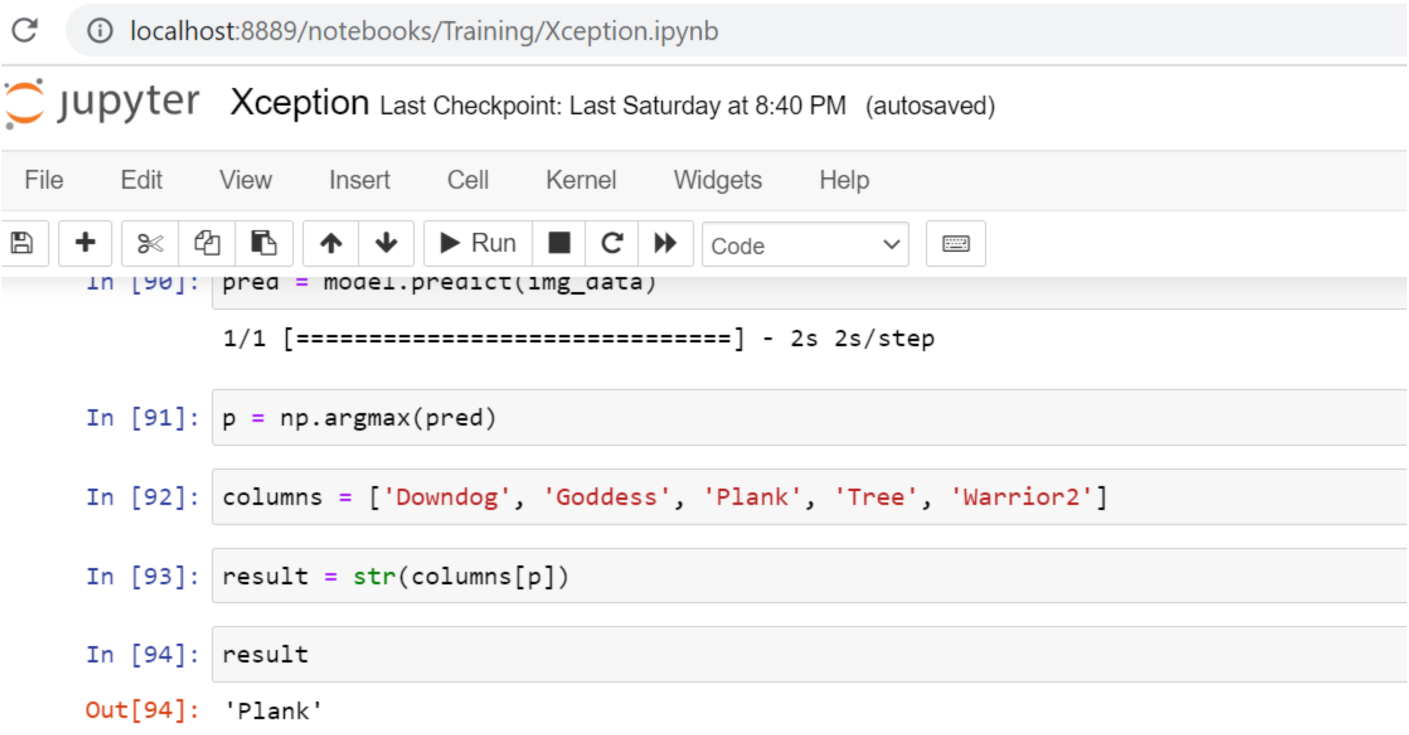

localhost:8889/notebooks/Training/Xception.ipynb
jupyter
Xception Last Checkpoint: Last Saturday at 8:40 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
In [76]:

```
model.fit(training_set,
          validation_data = testing_set,
          epochs =26,
          steps_per_epoch=len(training_set)//16,
          validation_steps = len(testing_set)//16)
```

```
Epoch 1/26
2/2 [==============================] - 13s 5s/step - loss: 4.6631 - Accuracy: 0.2188 - val_loss: 4.4785 - val_Accuracy: 0.4375
Epoch 2/26
2/2 [==============================] - 5s 3s/step - loss: 5.8325 - Accuracy: 0.3125 - val_loss: 3.2529 - val_Accuracy: 0.3750
Epoch 3/26
2/2 [==============================] - 4s 3s/step - loss: 3.4850 - Accuracy: 0.4091 - val_loss: 8.0596 - val_Accuracy: 0.3750
Epoch 4/26
2/2 [==============================] - 4s 2s/step - loss: 2.6428 - Accuracy: 0.5455 - val_loss: 3.4718 - val_Accuracy: 0.6250
Epoch 5/26
2/2 [==============================] - 5s 3s/step - loss: 2.6984 - Accuracy: 0.5625 - val_loss: 3.9483 - val_Accuracy: 0.6875
Epoch 6/26
2/2 [==============================] - 5s 3s/step - loss: 3.3503 - Accuracy: 0.7188 - val_loss: 0.1126 - val_Accuracy: 0.9375
Epoch 7/26
2/2 [==============================] - 5s 3s/step - loss: 1.7297 - Accuracy: 0.6875 - val_loss: 3.1026 - val_Accuracy: 0.6875
Epoch 8/26
2/2 [==============================] - 5s 3s/step - loss: 3.8490 - Accuracy: 0.6562 - val_loss: 1.6013 - val_Accuracy: 0.7500
Epoch 9/26
2/2 [==============================] - 5s 3s/step - loss: 2.2557 - Accuracy: 0.7188 - val_loss: 1.0935 - val_Accuracy: 0.8125
Epoch 10/26
2/2 [==============================] - 5s 3s/step - loss: 0.8110 - Accuracy: 0.8438 - val_loss: 0.8447 - val_Accuracy: 0.9375
Epoch 11/26
```


localhost:8889/notebooks/Training/Xception.ipynb
jupyter
Xception Last Checkpoint: Last Saturday at 8:40 PM (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
Out[76]: <keras.callbacks.History at 0x20911fcaee0>

In [95]: model.save('xcep_yoga.h5')

In [78]: model = load_model('xcep_yoga.h5')

In [85]: img = load_img("D:\YOGAPOSE\Dataset\Plank/00000000.jpg", target_size=(224,224))

In [86]: img = image.img_to_array(img)

In [87]: img.shape

Out[87]: (224, 224, 3)

In [89]: x = np.expand_dims(img, axis=0)
         img_data=preprocess_input(x)
         img_data.shape

Out[89]: (1, 224, 224, 3)

In [90]: pred = model.predict(img_data)

         1/1 [==============================] - 2s 2s/step
```


localhost:8889/notebooks/Training/Xception.ipynb
jupyter
Xception Last Checkpoint: Last Saturday at 8:40 PM (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
In [90]: pred = model.predict(img_data)

         1/1 [==============================] - 2s 2s/step

In [91]: p = np.argmax(pred)

In [92]: columns = ['Downdog', 'Goddess', 'Plank', 'Tree', 'Warrior2']

In [93]: result = str(columns[p])

In [94]: result

Out[94]: 'Plank'
```