```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
!unzip gdrive/MyDrive/birds/test_data.zip
```

```
Archive:  gdrive/MyDrive/birds/test_data.zip
   creating: test_data/
   creating: test_data/blasti/
  inflating: test_data/blasti/DSC_6396.jpg
  inflating: test_data/blasti/DSC_6397.jpg
  inflating: test_data/blasti/DSC_6398.jpg
  inflating: test_data/blasti/DSC_6399.jpg
  inflating: test_data/blasti/DSC_6400.jpg
  inflating: test_data/blasti/DSC_6401.jpg
  inflating: test_data/blasti/DSC_6402.jpg
  inflating: test_data/blasti/DSC_6403.jpg
  inflating: test_data/blasti/DSC_6405.jpg
  inflating: test_data/blasti/DSC_6406.jpg
  inflating: test_data/blasti/DSC_6407.jpg
  inflating: test_data/blasti/DSC_6408.jpg
  inflating: test_data/blasti/DSC_6409.jpg
  inflating: test_data/blasti/DSC_6410.jpg
  inflating: test_data/blasti/DSC_6411.jpg
   creating: test_data/bonegl/
  inflating: test_data/bonegl/DSC_4587.jpg
  inflating: test_data/bonegl/DSC_4588.jpg
  inflating: test_data/bonegl/DSC_4589.jpg
  inflating: test_data/bonegl/DSC_4590.jpg
  inflating: test_data/bonegl/DSC_4591.jpg
  inflating: test_data/bonegl/DSC_4592.jpg
  inflating: test_data/bonegl/DSC_4593.jpg
   creating: test_data/brhkyt/
  inflating: test_data/brhkyt/D72_0473.jpg
  inflating: test_data/brhkyt/D72_0474.jpg
  inflating: test_data/brhkyt/D72_0475.jpg
  inflating: test_data/brhkyt/D72_0477.jpg
  inflating: test_data/brhkyt/D72_0478.jpg
  inflating: test_data/brhkyt/D72_0479.jpg
   creating: test_data/cbrtsh/
  inflating: test_data/cbrtsh/_D32_10310.jpg
  inflating: test_data/cbrtsh/_D32_10311.jpg
  inflating: test_data/cbrtsh/_D32_10312.jpg
  inflating: test_data/cbrtsh/_D32_10313.jpg
  inflating: test_data/cbrtsh/_D32_10314.jpg
  inflating: test_data/cbrtsh/_D32_10317.jpg
  inflating: test_data/cbrtsh/_D32_10318.jpg
   creating: test_data/cmnmyn/
  inflating: test_data/cmnmyn/DSC_2443.jpg
  inflating: test_data/cmnmyn/DSC_4681.jpg
  inflating: test_data/cmnmyn/DSC_5137.jpg
  inflating: test_data/cmnmyn/DSC_7625.jpg
  inflating: test_data/cmnmyn/P1050277.jpg
  inflating: test_data/cmnmyn/_D32_12426.jpg
  inflating: test_data/cmnmyn/_D32_12427.jpg
  inflating: test_data/cmnmyn/_D32_12428.jpg
   creating: test_data/gretit/
  inflating: test_data/gretit/11620454726_31a35c26da_o.jpg
  inflating: test_data/gretit/11776135285_ccf938fa2e_o.jpg
  inflating: test_data/gretit/11905645146_6a5d4ff9f9_o.jpg
  inflating: test_data/gretit/8537646712_0b282c4c6a_o.jpg
  inflating: test_data/gretit/D72_0693.jpg
  inflating: test_data/gretit/D72_0694.jpg
  inflating: test_data/gretit/D72_0695.jpg
```

```
!unzip gdrive/MyDrive/birds/train_data.zip
```

```
Archive:  gdrive/MyDrive/birds/train_data.zip
   creating: train_data/
   creating: train_data/blasti/
  inflating: train_data/blasti/DSC_6382.jpg
  inflating: train_data/blasti/DSC_6383.jpg
  inflating: train_data/blasti/DSC_6384-2.jpg
  inflating: train_data/blasti/DSC_6384.jpg
  inflating: train_data/blasti/DSC_6385.jpg
  inflating: train_data/blasti/DSC_6386.jpg
  inflating: train_data/blasti/DSC_6387.jpg
  inflating: train_data/blasti/DSC_6388.jpg
  inflating: train_data/blasti/DSC_6389.jpg
  inflating: train_data/blasti/DSC_6390.jpg
  inflating: train_data/blasti/DSC_6391.jpg
  inflating: train_data/blasti/DSC_6392.jpg
  inflating: train_data/blasti/DSC_6393.jpg
  inflating: train_data/blasti/DSC_6394.jpg
  inflating: train_data/blasti/DSC_6395.jpg
   creating: train_data/bonegl/
  inflating: train_data/bonegl/DSC_4570.jpg
  inflating: train_data/bonegl/DSC_4571.jpg
  inflating: train_data/bonegl/DSC_4583.jpg
  inflating: train_data/bonegl/DSC_4584.jpg
  inflating: train_data/bonegl/DSC_4585.jpg
```

```
       inflating: train_data/bonegl/DSC_4586.jpg
        creating: train_data/brhkyt/
       inflating: train_data/brhkyt/D72_0400.jpg
       inflating: train_data/brhkyt/D72_0401.jpg
       inflating: train_data/brhkyt/D72_0470.jpg
       inflating: train_data/brhkyt/D72_0471.jpg
       inflating: train_data/brhkyt/D72_0472.jpg
        creating: train_data/cbrtsh/
       inflating: train_data/cbrtsh/100_5097.JPG
       inflating: train_data/cbrtsh/_D32_10303.jpg
       inflating: train_data/cbrtsh/_D32_10305.jpg
       inflating: train_data/cbrtsh/_D32_10306.jpg
       inflating: train_data/cbrtsh/_D32_10307.jpg
       inflating: train_data/cbrtsh/_D32_10308.jpg
       inflating: train_data/cbrtsh/_D32_10309.jpg
        creating: train_data/cmnmyn/
       inflating: train_data/cmnmyn/100_5763.JPG
       inflating: train_data/cmnmyn/5866682091_870ccc946c_o.jpg
       inflating: train_data/cmnmyn/6154956165_64266b8b53_o.jpg
       inflating: train_data/cmnmyn/DSCN5784.jpg
       inflating: train_data/cmnmyn/DSCN5787.jpg
       inflating: train_data/cmnmyn/DSCN5790.jpg
       inflating: train_data/cmnmyn/DSCN5791.jpg
        creating: train_data/gretit/
       inflating: train_data/gretit/100_5042.JPG
       inflating: train_data/gretit/100_5043.JPG
       inflating: train_data/gretit/100_5044.JPG
       inflating: train_data/gretit/100_5045.JPG
       inflating: train_data/gretit/100_5046.JPG
       inflating: train_data/gretit/100_5047.JPG
        creating: train_data/hilpig/
       inflating: train_data/hilpig/DSC_6272.jpg
       inflating: train_data/hilpig/DSC_6273.jpg
       inflating: train_data/hilpig/DSC_6274.jpg
```

```
from tensorflow.keras.layers import Dense,Flatten,Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
import numpy as np


train_path = '/content/train_data'
test_path = '/content/test_data'


train_gen = ImageDataGenerator(rescale=1./255,
                               shear_range=0.2,
                               zoom_range=0.2,
                               horizontal_flip=True)

test_gen = ImageDataGenerator(rescale=1./255)


train = train_gen.flow_from_directory(train_path,
                                      target_size=(224,224),
                                      batch_size=22,
                                      class_mode='categorical')

test = test_gen.flow_from_directory(test_path,
                                      target_size=(224,224),
                                      batch_size=22,
                                      class_mode='categorical')
```

```
Found 150 images belonging to 16 classes.
Found 157 images belonging to 16 classes.
```

VGG16

```
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input


vgg = VGG16(include_top=False,weights='imagenet',input_shape=(224,224,3))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 2s 0us/step
```

```
for layer in vgg.layers:
  print(layer)
```

```
<keras.engine.input_layer.InputLayer object at 0x7f56f7406320>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f7407be0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f7448b50>
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f56f7448c40>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f7449870>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f744a6e0>
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f56f744b760>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f744ac50>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f744bfd0>
```

```
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f7448580>
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f56f5364f70>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f53661a0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f5367010>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f5366d70>
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f56f5380640>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f5365f90>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f5366710>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f5382320>
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f56f53833d0>
```

```
for layer in vgg.layers:
  layer.trainable=False
```

```
x = Flatten()(vgg.output)
```

```
prediction = Dense(16,activation='softmax')(x)
```

```
model = Model(inputs=vgg.input,outputs=prediction)
```

```
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 16)                401424

=================================================================
Total params: 15,116,112
Trainable params: 401,424
Non-trainable params: 14,714,688
_________________________________________________________________
```

```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
model.fit_generator(train,validation_data=test,epochs=10,steps_per_epoch=len(train),validation_steps=len(test))
```

```
<ipython-input-19-c19f13cb6f36>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit
  model.fit_generator(train,validation_data=test,epochs=10,steps_per_epoch=len(train),validation_steps=len(test))
Epoch 1/10
7/7 [==============================] - 108s 15s/step - loss: 3.9631 - accuracy: 0.1400 - val_loss: 3.2677 - val_accuracy: 0.1847
Epoch 2/10
7/7 [==============================] - 91s 14s/step - loss: 1.9688 - accuracy: 0.4600 - val_loss: 2.9406 - val_accuracy: 0.2484
Epoch 3/10
7/7 [==============================] - 92s 14s/step - loss: 0.9985 - accuracy: 0.6867 - val_loss: 3.1222 - val_accuracy: 0.3185
Epoch 4/10
7/7 [==============================] - 121s 19s/step - loss: 0.7099 - accuracy: 0.7933 - val_loss: 2.8847 - val_accuracy: 0.3631
Epoch 5/10
7/7 [==============================] - 92s 15s/step - loss: 0.3683 - accuracy: 0.9000 - val_loss: 2.9352 - val_accuracy: 0.3503
```

```
Epoch 6/10
7/7 [==============================] - 88s 14s/step - loss: 0.2516 - accuracy: 0.9467 - val_loss: 2.9953 - val_accuracy: 0.3567
Epoch 7/10
7/7 [==============================] - 92s 15s/step - loss: 0.1970 - accuracy: 0.9467 - val_loss: 3.0808 - val_accuracy: 0.3694
Epoch 8/10
7/7 [==============================] - 93s 15s/step - loss: 0.1391 - accuracy: 0.9733 - val_loss: 3.1347 - val_accuracy: 0.3694
Epoch 9/10
7/7 [==============================] - 91s 14s/step - loss: 0.0923 - accuracy: 1.0000 - val_loss: 2.8206 - val_accuracy: 0.4204
Epoch 10/10
7/7 [==============================] - 123s 20s/step - loss: 0.0668 - accuracy: 1.0000 - val_loss: 2.8776 - val_accuracy: 0.4140
<keras.callbacks.History at 0x7f56f40d2e60>
```

ResNet50

```
from tensorflow.keras.applications.resnet50 import ResNet50
```

```
resnet = ResNet50(include_top=False,input_shape=(224,224,3))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [==============================] - 3s 0us/step
```

```
for layer in resnet.layers:
  print(layer)
```

```
<keras.engine.input_layer.InputLayer object at 0x7f56f40d3bb0>
<keras.layers.reshaping.zero_padding2d.ZeroPadding2D object at 0x7f56f539a950>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f17d60>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f56f539bb20>
<keras.layers.core.activation.Activation object at 0x7f5667f17820>
<keras.layers.reshaping.zero_padding2d.ZeroPadding2D object at 0x7f5667f17970>
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f5667f24ac0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f26ec0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f27dc0>
<keras.layers.core.activation.Activation object at 0x7f5667f24f70>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f252a0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f54850>
<keras.layers.core.activation.Activation object at 0x7f5667f57a60>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f7cd8c70>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f57190>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f27280>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f56080>
<keras.layers.merging.add.Add object at 0x7f56f539aa40>
<keras.layers.core.activation.Activation object at 0x7f56f7d1f0a0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f25510>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f57430>
<keras.layers.core.activation.Activation object at 0x7f5667f569e0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f552a0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f6a380>
<keras.layers.core.activation.Activation object at 0x7f5667f6a0e0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f6afb0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f6a6b0>
<keras.layers.merging.add.Add object at 0x7f56e00c2770>
<keras.layers.core.activation.Activation object at 0x7f5667f6b880>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f6b6a0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f923e0>
<keras.layers.core.activation.Activation object at 0x7f5667f6b160>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f927d0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667f93c10>
<keras.layers.core.activation.Activation object at 0x7f5667f93640>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667f93010>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667fac9a0>
<keras.layers.merging.add.Add object at 0x7f5667f92f50>
<keras.layers.core.activation.Activation object at 0x7f5667f92140>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667fae4a0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667fafa00>
<keras.layers.core.activation.Activation object at 0x7f5667fafcd0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667dd5a20>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667dd6f80>
<keras.layers.core.activation.Activation object at 0x7f5667dd6d40>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667faee30>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667dd57b0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667fafca0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667dd7460>
<keras.layers.merging.add.Add object at 0x7f5667dd7c70>
<keras.layers.core.activation.Activation object at 0x7f5667df9300>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667df9990>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667dfb070>
<keras.layers.core.activation.Activation object at 0x7f5667dfa890>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667dfbca0>
<keras.layers.normalization.batch_normalization.BatchNormalization object at 0x7f5667df8940>
<keras.layers.core.activation.Activation object at 0x7f5667dfa4d0>
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f5667dfb3a0>
```

```
x = Flatten()(resnet.output)
```

```
out = Dense(16, activation='softmax')(x)
```

```
res_model = Model(inputs=resnet.input,outputs=out)
```

```
res_model.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 conv1_pad (ZeroPadding2D)      (None, 230, 230, 3)  0           ['input_2[0][0]']

 conv1_conv (Conv2D)            (None, 112, 112, 64  9472        ['conv1_pad[0][0]']
                                )

 conv1_bn (BatchNormalization)  (None, 112, 112, 64  256         ['conv1_conv[0][0]']
                                )

 conv1_relu (Activation)        (None, 112, 112, 64  0           ['conv1_bn[0][0]']
                                )

 pool1_pad (ZeroPadding2D)      (None, 114, 114, 64  0           ['conv1_relu[0][0]']
                                )

 pool1_pool (MaxPooling2D)      (None, 56, 56, 64)   0           ['pool1_pad[0][0]']

 conv2_block1_1_conv (Conv2D)   (None, 56, 56, 64)   4160        ['pool1_pool[0][0]']

 conv2_block1_1_bn (BatchNormal  (None, 56, 56, 64)  256         ['conv2_block1_1_conv[0][0]']
 ization)

 conv2_block1_1_relu (Activatio  (None, 56, 56, 64)  0           ['conv2_block1_1_bn[0][0]']
 n)

 conv2_block1_2_conv (Conv2D)   (None, 56, 56, 64)   36928       ['conv2_block1_1_relu[0][0]']

 conv2_block1_2_bn (BatchNormal  (None, 56, 56, 64)  256         ['conv2_block1_2_conv[0][0]']
 ization)

 conv2_block1_2_relu (Activatio  (None, 56, 56, 64)  0           ['conv2_block1_2_bn[0][0]']
 n)

 conv2_block1_0_conv (Conv2D)   (None, 56, 56, 256)  16640       ['pool1_pool[0][0]']

 conv2_block1_3_conv (Conv2D)   (None, 56, 56, 256)  16640       ['conv2_block1_2_relu[0][0]']

 conv2_block1_0_bn (BatchNormal  (None, 56, 56, 256)  1024       ['conv2_block1_0_conv[0][0]']
 ization)

 conv2_block1_3_bn (BatchNormal  (None, 56, 56, 256)  1024       ['conv2_block1_3_conv[0][0]']
 ization)

 conv2_block1_add (Add)         (None, 56, 56, 256)  0           ['conv2_block1_0_bn[0][0]',
                                                                  'conv2_block1_3_bn[0][0]']

 conv2_block1_out (Activation)  (None, 56, 56, 256)  0           ['conv2_block1_add[0][0]']

 conv2_block2_1_conv (Conv2D)   (None, 56, 56, 64)   16448       ['conv2_block1_out[0][0]']

 conv2_block2_1_bn (BatchNormal  (None, 56, 56, 64)  256         ['conv2_block2_1_conv[0][0]']
 ization)
```

```
res_model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
res_model.fit(train,epochs=10,validation_data=test,steps_per_epoch=len(train),
              validation_steps=len(test))
```

```
Epoch 1/10
7/7 [==============================] - 145s 15s/step - loss: 15.9062 - accuracy: 0.2000 - val_loss: 101.7114 - val_accuracy: 0.0573
Epoch 2/10
7/7 [==============================] - 92s 14s/step - loss: 8.1270 - accuracy: 0.4933 - val_loss: 1438412.2500 - val_accuracy: 0.0573
Epoch 3/10
7/7 [==============================] - 92s 14s/step - loss: 7.5458 - accuracy: 0.4533 - val_loss: 5777523.0000 - val_accuracy: 0.1274
Epoch 4/10
7/7 [==============================] - 122s 19s/step - loss: 4.0220 - accuracy: 0.5467 - val_loss: 7797352.5000 - val_accuracy: 0.0510
Epoch 5/10
7/7 [==============================] - 91s 14s/step - loss: 6.7724 - accuracy: 0.6333 - val_loss: 2681508.5000 - val_accuracy: 0.1274
Epoch 6/10
7/7 [==============================] - 90s 14s/step - loss: 10.9263 - accuracy: 0.5067 - val_loss: 10694477.0000 - val_accuracy: 0.1274
Epoch 7/10
7/7 [==============================] - 93s 14s/step - loss: 2.5144 - accuracy: 0.5667 - val_loss: 128072608.0000 - val_accuracy: 0.1274
Epoch 8/10
7/7 [==============================] - 91s 14s/step - loss: 2.1205 - accuracy: 0.6333 - val_loss: 52614392.0000 - val_accuracy: 0.1274
Epoch 9/10
7/7 [==============================] - 90s 14s/step - loss: 1.8227 - accuracy: 0.6800 - val_loss: 355240640.0000 - val_accuracy: 0.1274
Epoch 10/10
7/7 [==============================] - 93s 14s/step - loss: 3.5036 - accuracy: 0.6867 - val_loss: 752114048.0000 - val_accuracy: 0.1274
<keras.callbacks.History at 0x7f5667e4e4a0>
```

VGG16 gives higher val_accuracy.

```
import numpy as np
from tensorflow.keras.preprocessing import image


img = image.load_img('/content/bird-img.jpg',target_size=(224,224))
img = image.img_to_array(img)
img = np.expand_dims(img,axis=0)
pred = np.argmax(model.predict(img))
print(pred)
output = ['bird-0','bird-1','bird-2','bird-3','bird-4','bird-5','bird-6','bird-7','bird-8','bird-9','bird-10','bird-11','bird-12','bird-13','bird-14','b
print(output[pred])
```

```
1/1 [==============================] - 1s 716ms/step
4
bird-4
```