

```
In [3]: from google.colab import drive
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

```
In [4]: !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
```

```
In [5]: !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
```

```
In [6]: 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
        import numpy as np
```

```
In [7]: train_path = '/content/train_data'
        test_path = '/content/test_data'
```

```
In [8]: train_gen = ImageDataGenerator(rescale=1./255,
                                       shear_range=0.2,
                                       zoom_range=0.2,
                                       horizontal_flip=True)

        test_gen = ImageDataGenerator(rescale=1./255)
```

```
In [9]: 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

```
In [10]: from tensorflow.keras.applications.vgg16 import VGG16, preprocess_i
```

```
In [11]: vgg = VGG16(include_top=False, weights='imagenet', input_shape=(224, 2
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_no_top.h5 (https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5)
58889256/58889256 [=====] - 2s 0us/step

```
In [12]: 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 0x7f56f  
7448c40>  
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f7449870  
>  
<keras.layers.convolutional.conv2d.Conv2D object at 0x7f56f744a6e0  
>  
<keras.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7f56f  
744b760>  
<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 0x7f56f  
5364f70>  
<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 0x7f56f  
5380640>  
<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 0x7f56f  
53833d0>
```

```
In [13]: for layer in vgg.layers:  
         layer.trainable=False
```

```
In [14]: x = Flatten()(vgg.output)
```

```
In [15]: prediction = Dense(16,activation='softmax')(x)
```

```
In [16]: model = Model(inputs=vgg.input,outputs=prediction)
```

```
In [17]:
```

```
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		

```
In [18]: model.compile(loss='categorical_crossentropy',optimizer='adam',metr
```

```
In [19]: model.fit_generator(train,validation_data=test,epochs=10,steps_per_
```

```
<ipython-input-19-c19f13cb6f36>:1: UserWarning: `Model.fit_generat
or` is deprecated and will be removed in a future version. Please
use `Model.fit`, which supports generators.
```

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

```
Epoch 1/10
```

```
7/7 [=====] - 108s 15s/step - loss: 3.963
1 - 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: 0.709
9 - 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: 0.066
8 - accuracy: 1.0000 - val_loss: 2.8776 - val_accuracy: 0.4140
```

```
Out[19]: <keras.callbacks.History at 0x7f56f40d2e60>
```

```
##ResNet50
```

```
In [20]: from tensorflow.keras.applications.resnet50 import ResNet50
```

```
In [21]: 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 (https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5)
 94765736/94765736 [=====] - 3s 0us/step

```
In [22]: for layer in resnet.layers:
          print(layer)
```

```
<keras.engine.input_layer.InputLayer object at 0x7f56f40d3bb0>
<keras.layers.resizing.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.resizing.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 0x7f5667f252a0>
```

```
In [23]: x = Flatten()(resnet.output)
```

```
In [24]: out = Dense(16, activation='softmax')(x)
```

```
In [25]: res_model = Model(inputs=resnet.input, outputs=out)
```

In [26]: `res_model.summary()`

Model: "model_1"

Layer (type) connected to	Output Shape	Param #	C
=====			
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]']			

In [27]: `res_model.compile(loss='categorical_crossentropy',optimizer='adam',`

```
In [28]: res_model.fit(train,epochs=10,validation_data=test,steps_per_epoch=
          validation_steps=len(test))
```

```
Epoch 1/10
7/7 [=====] - 145s 15s/step - loss: 15.90
62 - 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: 4.022
0 - accuracy: 0.5467 - val_loss: 7797352.5000 - val_accuracy: 0.05
10
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: 10.926
3 - accuracy: 0.5067 - val_loss: 10694477.0000 - val_accuracy: 0.1
274
Epoch 7/10
7/7 [=====] - 93s 14s/step - loss: 2.5144
- accuracy: 0.5667 - val_loss: 128072608.0000 - val_accuracy: 0.12
74
Epoch 8/10
7/7 [=====] - 91s 14s/step - loss: 2.1205
- accuracy: 0.6333 - val_loss: 52614392.0000 - val_accuracy: 0.127
4
Epoch 9/10
7/7 [=====] - 90s 14s/step - loss: 1.8227
- accuracy: 0.6800 - val_loss: 355240640.0000 - val_accuracy: 0.12
74
Epoch 10/10
7/7 [=====] - 93s 14s/step - loss: 3.5036
- accuracy: 0.6867 - val_loss: 752114048.0000 - val_accuracy: 0.12
74
```

```
Out[28]: <keras.callbacks.History at 0x7f5667e4e4a0>
```

VGG16 gives higher val_accuracy.

```
In [29]: import numpy as np
         from tensorflow.keras.preprocessing import image
```



```
In [30]: 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','bi
print(output[pred])
```

1/1 [=====] - 1s 716ms/step

4

bird-4