

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: import tensorflow as tf
from tensorflow import keras
```

```
In [3]: train_dir = r'C:\Users\shant\OneDrive\Documents\Python Scripts\train'
validation_dir = r'C:\Users\shant\OneDrive\Documents\Python Scripts\validation'
test_dir = r'C:\Users\shant\OneDrive\Documents\Python Scripts\test'
```

```
In [4]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [5]: train_datagen = ImageDataGenerator(rescale = 1.0/255,
rotation_range = 40,
width_shift_range = 0.2,
height_shift_range = 0.2,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1.0/255)

train_generator = train_datagen.flow_from_directory(train_dir,
target_size = (150,150),
batch_size = 20,
class_mode = 'binary')
validation_generator = train_datagen.flow_from_directory(validation_dir,
target_size = (150,150),
batch_size = 20,
class_mode = 'binary')
```

Found 2000 images belonging to 2 classes.  
Found 1000 images belonging to 2 classes.

```
In [6]: from tensorflow.keras.applications import VGG16
```

```
In [7]: conv_base = VGG16(weights = 'imagenet',
include_top = False,
input_shape = (150,150,3))
```

```
In [8]: conv_base.summary()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, 150, 150, 3)]	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
=====		
Total params: 14,714,688		
Trainable params: 14,714,688		
Non-trainable params: 0		

```
In [9]: from tensorflow.keras import models
        from tensorflow.keras import layers
```

```
In [10]: model = models.Sequential()
```

```
In [11]: model.add(conv_base)
        model.add(layers.Flatten())
        model.add(layers.Dense(256,activation = 'relu'))
        model.add(layers.Dense(1,activation = 'sigmoid'))
```

```
In [12]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
vgg16 (Functional)	(None, 4, 4, 512)	14714688
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 256)	2097408
dense_1 (Dense)	(None, 1)	257
=====		
Total params: 16,812,353		
Trainable params: 16,812,353		
Non-trainable params: 0		

```
In [13]: conv_base.trainable = False
```

```
In [14]: from tensorflow.keras import optimizers
```

```
In [15]: model.compile(loss = 'binary_crossentropy',
                    optimizer = optimizers.RMSprop(lr = 2e-5),
                    metrics = ['accuracy'])
```

```
C:\Users\shant\anaconda3\lib\site-packages\keras\optimizers\optimizer_v2\rmsprop.py:140: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.  
super().__init__(name, **kwargs)
```

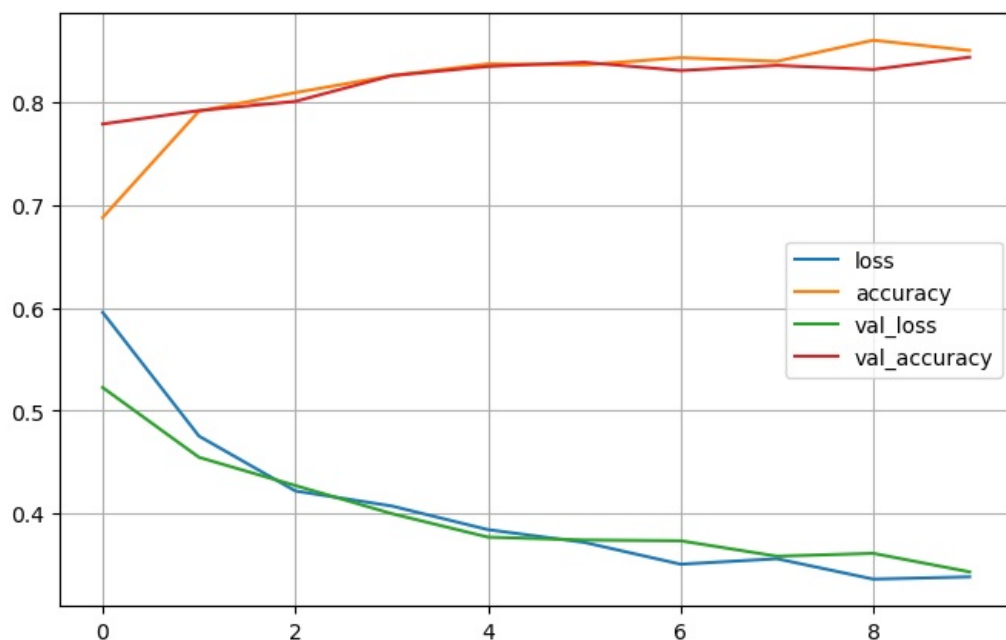
```
In [16]: import os  
os.environ['KMP_DUPLICATE_LIB_OK'] = 'True'
```

```
In [17]: checkpoint_cb = keras.callbacks.ModelCheckpoint('transfer_learning-{epoch:02d}.h5', save_best = True)
```

```
In [18]: history = model.fit(train_generator,  
                             steps_per_epoch = 100,  
                             epochs = 10,  
                             validation_data = validation_generator,  
                             callbacks = [checkpoint_cb])
```

```
Epoch 1/10  
100/100 [=====] - 455s 5s/step - loss: 0.5957 - accuracy: 0.6880 - val_loss: 0.5226 -  
val_accuracy: 0.7790  
Epoch 2/10  
100/100 [=====] - 471s 5s/step - loss: 0.4753 - accuracy: 0.7915 - val_loss: 0.4547 -  
val_accuracy: 0.7920  
Epoch 3/10  
100/100 [=====] - 461s 5s/step - loss: 0.4220 - accuracy: 0.8095 - val_loss: 0.4272 -  
val_accuracy: 0.8010  
Epoch 4/10  
100/100 [=====] - 466s 5s/step - loss: 0.4073 - accuracy: 0.8260 - val_loss: 0.3999 -  
val_accuracy: 0.8260  
Epoch 5/10  
100/100 [=====] - 452s 5s/step - loss: 0.3843 - accuracy: 0.8375 - val_loss: 0.3770 -  
val_accuracy: 0.8350  
Epoch 6/10  
100/100 [=====] - 454s 5s/step - loss: 0.3719 - accuracy: 0.8365 - val_loss: 0.3743 -  
val_accuracy: 0.8390  
Epoch 7/10  
100/100 [=====] - 492s 5s/step - loss: 0.3507 - accuracy: 0.8435 - val_loss: 0.3735 -  
val_accuracy: 0.8310  
Epoch 8/10  
100/100 [=====] - 503s 5s/step - loss: 0.3560 - accuracy: 0.8400 - val_loss: 0.3585 -  
val_accuracy: 0.8360  
Epoch 9/10  
100/100 [=====] - 626s 6s/step - loss: 0.3362 - accuracy: 0.8605 - val_loss: 0.3613 -  
val_accuracy: 0.8320  
Epoch 10/10  
100/100 [=====] - 641s 6s/step - loss: 0.3384 - accuracy: 0.8505 - val_loss: 0.3432 -  
val_accuracy: 0.8440
```

```
In [19]: pd.DataFrame(history.history).plot(figsize = (8,5))  
plt.grid(True)  
plt.show()
```



```
In [ ]:
```