In [ ]:

In [ ]:
```
!wget --header="Host: doc-10-7s-docs.googleusercontent.com" --header="User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) Ap
```

```
--2020-08-30 02:33:13--  https://doc-10-7s-docs.googleusercontent.com/docs/securesc/dpls5385kd35knv027jpvpgk1o51socs/sq5m50qpb
l/1598754525000/06917965946637727718/06917965946637727718/1D7AeGa-bQsZ_5cI4_ZU5ZRCV-sJGf4rH?e=download&authuser=0&nonce=qik5q6
46637727718&hash=0381vvm3ittm8r9rfl7asb8v78op9ibk (https://doc-10-7s-docs.googleusercontent.com/docs/securesc/dpls5385kd35knv0
0qpbc1thqnfqtadocke66050s4l/1598754525000/06917965946637727718/06917965946637727718/1D7AeGa-bQsZ_5cI4_ZU5ZRCV-sJGf4rH?e=downlo
k5q66bsulq4&user=06917965946637727718&hash=0381vvm3ittm8r9rfl7asb8v78op9ibk)
Resolving doc-10-7s-docs.googleusercontent.com (doc-10-7s-docs.googleusercontent.com)... 172.253.123.132, 2607:f8b0:400c:c16:
Connecting to doc-10-7s-docs.googleusercontent.com (doc-10-7s-docs.googleusercontent.com)|172.253.123.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/rar]
Saving to: ‘Data_Final_main.rar’

Data_Final_main.rar     [ <=>                ]   4.36G  74.3MB/s    in 67s

2020-08-30 02:34:21 (66.3 MB/s) - ‘Data_Final_main.rar’ saved [4683832870]
```

In [ ]:

In [ ]:
```
! wget --header="Host: doc-10-7s-docs.googleusercontent.com" --header="User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
```

```
--2020-08-30 02:40:58--  https://doc-10-7s-docs.googleusercontent.com/docs/securesc/dpls5385kd35knv027jpvpgk1o51socs/sq5m50qpb
l/1598754525000/06917965946637727718/06917965946637727718/1D7AeGa-bQsZ_5cI4_ZU5ZRCV-sJGf4rH?e=download&authuser=0&nonce=qik5q6
46637727718&hash=0381vvm3ittm8r9rfl7asb8v78op9ibk (https://doc-10-7s-docs.googleusercontent.com/docs/securesc/dpls5385kd35knv0
0qpbc1thqnfqtadocke66050s4l/1598754525000/06917965946637727718/06917965946637727718/1D7AeGa-bQsZ_5cI4_ZU5ZRCV-sJGf4rH?e=downlo
k5q66bsulq4&user=06917965946637727718&hash=0381vvm3ittm8r9rfl7asb8v78op9ibk)
Resolving doc-10-7s-docs.googleusercontent.com (doc-10-7s-docs.googleusercontent.com)... 172.253.123.132, 2607:f8b0:400c:c16:
Connecting to doc-10-7s-docs.googleusercontent.com (doc-10-7s-docs.googleusercontent.com)|172.253.123.132|:443... connected.
HTTP request sent, awaiting response... 403 Forbidden
2020-08-30 02:40:58 ERROR 403: Forbidden.
```

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

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i
t.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20htt
pis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2f
nly&response_type=code (https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps
&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a
om%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%
esponse_type=code)

Enter your authorization code:
··········
Mounted at /gdrive
/gdrive
Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdrive", force_remount=True).
/gdrive
```

Referance link - to unzip the rar file type https://colab.research.google.com/drive/1xinRwhXtlL-9Y0KbPrTmTxNdcN-Hvq4m#scrollTo=5ScZvnCdzKm3 (https://colab.research.google.com/drive/1xinRwhXtlL-9Y0KbPrTmTxNdcN-Hvq4m#scrollTo=5ScZvnCdzKm3)

In [ ]:
```
!unrar x "/gdrive/My Drive/Data_Final_main.rar" "/gdrive/My Drive/"
```

```
Streaming output truncated to the last 5000 lines.
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2042850402.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2043047892_2043047893.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2043520616.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2043524108.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2043561111_2043561117.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044153842.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044153843.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044164543.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044186248_6251.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044187167_7168.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044312329_2330.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044333689_3710.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044336342_2044336345.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044412864_2865.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044678856_8880.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044779176_2044779177.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044804002.tif    88  OK
Extracting  /gdrive/My Drive/Data_Final_main/val_test/presentation/2044895795_5796.tif    88  OK
```

In [ ]:
```
%tensorflow_version 2.x
```

In [ ]:
```
import pandas as pd
import shutil
import os
from tqdm import tqdm
import tensorflow as tf
#importing tensorflow
from tensorflow.keras.layers import Dense,Input,Conv2D,MaxPool2D,Activation,Dropout,Flatten
from tensorflow.keras.models import Model
import random as rn
from tensorflow.keras import applications
from tensorflow.keras.models import Sequential
import numpy as np
from tensorflow.keras.callbacks import TensorBoard
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping
import datetime
#importing tensorflow
from tensorflow.keras.layers import Dense,Input,Conv2D,MaxPool2D,Activation,Dropout,Flatten
from tensorflow.keras.models import Model
import random as rn
```

In [ ]:
```
tf.__version__
```

Out[6]:
```
'2.3.0'
```

In [ ]:
```
dir_path = '/gdrive/My Drive/Data_Final_main/train'
```

In [ ]:
```
tf.test.gpu_device_name()
```

Out[7]:
```
'/device:GPU:0'
```

In [ ]:
```
# Train folder classes count
for i in os.listdir(dir_path):

    print("No of Images in ",i," category is ",len(os.listdir(os.path.join(dir_path,i))))
```

```
No of Images in  news_article  category is  2101
No of Images in  presentation  category is  2106
No of Images in  questionnaire  category is  2106
No of Images in  resume  category is  2104
No of Images in  scientific_publication  category is  2085
No of Images in  scientific_report  category is  2099
No of Images in  specification  category is  2100
No of Images in  advertisement  category is  2094
No of Images in  budget  category is  2102
No of Images in  email  category is  2093
No of Images in  file_folder  category is  2103
No of Images in  form  category is  2094
No of Images in  handwritten  category is  2105
No of Images in  invoice  category is  2092
No of Images in  letter  category is  2431
No of Images in  memo  category is  2096
```

In [ ]:
```
dir_path_test = '/gdrive/My Drive/Data_Final_main/val_test'
```

In [ ]:
```
for i in os.listdir(dir_path_test):

    print("No of Images in ",i," category is ",len(os.listdir(os.path.join(dir_path_test,i))))
```

```
No of Images in  advertisement  category is  900
No of Images in  budget  category is  900
No of Images in  email  category is  900
No of Images in  file_folder  category is  900
No of Images in  form  category is  900
No of Images in  handwritten  category is  900
No of Images in  invoice  category is  900
No of Images in  letter  category is  900
No of Images in  memo  category is  900
No of Images in  news_article  category is  900
No of Images in  presentation  category is  900
No of Images in  questionnaire  category is  900
No of Images in  resume  category is  900
No of Images in  scientific_publication  category is  900
No of Images in  scientific_report  category is  900
No of Images in  specification  category is  900
```

In [ ]:
```
train_data_dir = dir_path
validation_data_dir = dir_path_test
```

In [ ]:
```
epochs = 20
batch_size = 32
#batch_size = 128
#img_width, img_height = 150, 150
img_width, img_height = 224,224

#VGG16 trained with 224,244 image size we using VGG16
```

In [ ]:
```
# prepare data augmentation configuration
train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)



test_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1. / 255)



train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(224,224),
    batch_size=batch_size,
    class_mode='categorical')


validation_generator = test_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(224,224),
    batch_size=batch_size,
    class_mode='categorical')
```

```
Found 33911 images belonging to 16 classes.
Found 14400 images belonging to 16 classes.
```

In [ ]:

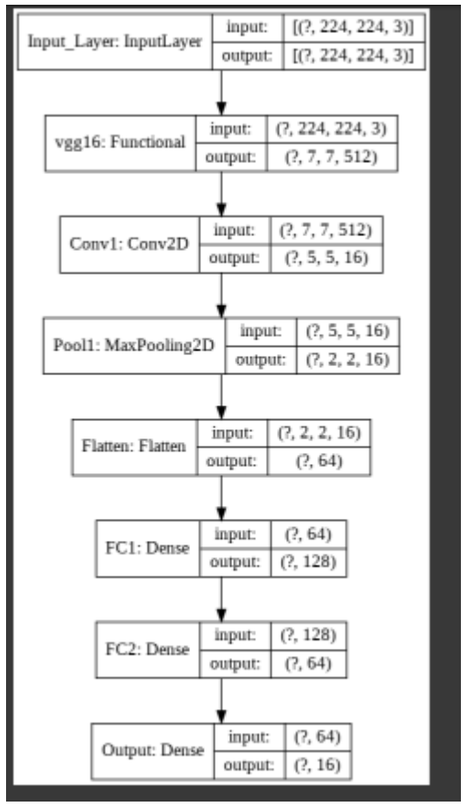
```
# Create Model
os.environ['PYTHONHASHSEED'] = '0'

##https://keras.io/getting-started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development
## Have to clear the session. If you are not clearing, Graph will create again and again and graph size will increses.
## Varibles will also set to some value from before session
tf.keras.backend.clear_session()

## Set the random seed values to regenerate the model.
np.random.seed(0)
rn.seed(0)



#Get back the convolutional part of a VGG network trained on ImageNet
model_vgg16_conv = applications.VGG16(weights='imagenet', include_top=False,input_shape=(150,150,3))


# Freezing No trainable layer
for layer in model_vgg16_conv.layers:
    layer.trainable = False
#model_vgg16_conv.summary()

#Input layer - Create your own input format (here 150,150,3)
input_layer = Input(shape=(224,224,3),name='Input_Layer')

#Use the generated model
output_vgg16_conv = model_vgg16_conv(input_layer)

#Conv Layer
Conv1 = Conv2D(filters=256,kernel_size=(3,3),strides=(1,1),padding='valid',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv1')(output_vgg16_conv)

#MaxPool Layer
Pool1 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='valid',data_format='channels_last',name='Pool1')(Conv1)


#Flatten
flatten = Flatten(data_format='channels_last',name='Flatten')(Pool1)

#FC layer
FC1 = Dense(units=128,activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=45),name='FC1')(flatten)
```

```
#FC layer
FC2 = Dense(units=64,activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=35),name='FC2')(FC1)

#output layer
Out = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.initializers.he_normal(seed=3),name='Output')(FC2)


model = Model(inputs=input_layer,outputs=Out)


model.summary()
```

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Input_Layer (InputLayer)     [(None, 150, 150, 3)]     0
_________________________________________________________________
vgg16 (Functional)           (None, 4, 4, 512)         14714688
_________________________________________________________________
Conv1 (Conv2D)               (None, 2, 2, 256)         1179904
_________________________________________________________________
Pool1 (MaxPooling2D)         (None, 1, 1, 256)         0
_________________________________________________________________
Flatten (Flatten)            (None, 256)               0
_________________________________________________________________
FC1 (Dense)                  (None, 128)               32896
_________________________________________________________________
FC2 (Dense)                  (None, 64)                8256
_________________________________________________________________
Output (Dense)               (None, 16)                1040
=================================================================
Total params: 15,936,784
Trainable params: 1,222,096
Non-trainable params: 14,714,688
_________________________________________________________________
```

In [6]:
```
from IPython.display import Image
Image(filename=r"C:\Users\User\Desktop\Applied_A\Program\DEEP_Learning\Callback\Callbacks\plot_image_task1.PNG")
```

Out[6]:



In [ ]:
```
# eARLY sTOOPING
earlystop = EarlyStopping(monitor='loss', patience=3, verbose=1)


rate_learning = tf.keras.callbacks.ReduceLROnPlateau(
    monitor='val_loss', factor=0.1, patience=2, verbose=0, mode='auto',
    min_delta=0.001, cooldown=0, min_lr=0
)


##Callbacks
#file path, it saves the model in the 'model_save' folder and we are naming model with epoch number
#and val acc to differtiate with other models
#you have to create model_save folder before running the code.
filepath="model_save/weights-{epoch:02d}.hdf5"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_loss',  verbose=1, save_best_only=True, mode='auto')
```

In [ ]:
```
# TensorBoard Creation
%load_ext tensorboard
folder_name = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```

In [ ]:
```
# Create log folder - TensorBoard
log_dir="/gdrive/My Drive/logs/fit/" + folder_name
tensorboard_callback =TensorBoard(log_dir=log_dir,histogram_freq=0, write_graph=True)
```

In [ ]:
```
folder_name
```

Out[61]:
```
'20200905-142122'
```

In [ ]:
```
#compiling
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),loss='categorical_crossentropy',metrics=['accuracy'])
```

In [ ]:
```
##fitting generator
model.fit(train_generator,steps_per_epoch=1060,epochs=20,
          validation_data=validation_generator,
validation_steps=450,
callbacks=[checkpoint,earlystop,rate_learning,tensorboard_callback])
```

```
Epoch 1/20
   1/1060 [..............................] - ETA: 0s - loss: 3.4569 - accuracy: 0.0000e+00WARNING:tensorflow:From /usr/local/l
ages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from tensorflow.python.eager.profiler) is deprecated and will be remo
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
1060/1060 [==============================] - ETA: 0s - loss: 2.0291 - accuracy: 0.3693
Epoch 00001: val_loss improved from inf to 1.71295, saving model to model_save/weights-01.hdf5
1060/1060 [==============================] - 32915s 31s/step - loss: 2.0291 - accuracy: 0.3693 - val_loss: 1.7129 - val_accura
Epoch 2/20
1060/1060 [==============================] - ETA: 0s - loss: 1.7080 - accuracy: 0.4657
Epoch 00002: val_loss improved from 1.71295 to 1.55061, saving model to model_save/weights-02.hdf5
1060/1060 [==============================] - 539s 508ms/step - loss: 1.7080 - accuracy: 0.4657 - val_loss: 1.5506 - val_accura
Epoch 3/20
1060/1060 [==============================] - ETA: 0s - loss: 1.6338 - accuracy: 0.4930
Epoch 00003: val_loss did not improve from 1.55061
1060/1060 [==============================] - 526s 496ms/step - loss: 1.6338 - accuracy: 0.4930 - val_loss: 1.6157 - val_accura
Epoch 4/20
1060/1060 [==============================] - ETA: 0s - loss: 1.6025 - accuracy: 0.5062
Epoch 00004: val_loss did not improve from 1.55061
1060/1060 [==============================] - 510s 481ms/step - loss: 1.6025 - accuracy: 0.5062 - val_loss: 1.6116 - val_accura
Epoch 5/20
1060/1060 [==============================] - ETA: 0s - loss: 1.4310 - accuracy: 0.5534
Epoch 00005: val_loss improved from 1.55061 to 1.39720, saving model to model_save/weights-05.hdf5
1060/1060 [==============================] - 491s 463ms/step - loss: 1.4310 - accuracy: 0.5534 - val_loss: 1.3972 - val_accura
Epoch 6/20
1060/1060 [==============================] - ETA: 0s - loss: 1.3909 - accuracy: 0.5670
Epoch 00006: val_loss improved from 1.39720 to 1.39255, saving model to model_save/weights-06.hdf5
1060/1060 [==============================] - 491s 463ms/step - loss: 1.3909 - accuracy: 0.5670 - val_loss: 1.3926 - val_accura
Epoch 7/20
 146/1060 [===>..........................] - ETA: 5:41 - loss: 1.3615 - accuracy: 0.5749
```

In [ ]:
```
%load_ext tensorboard
```

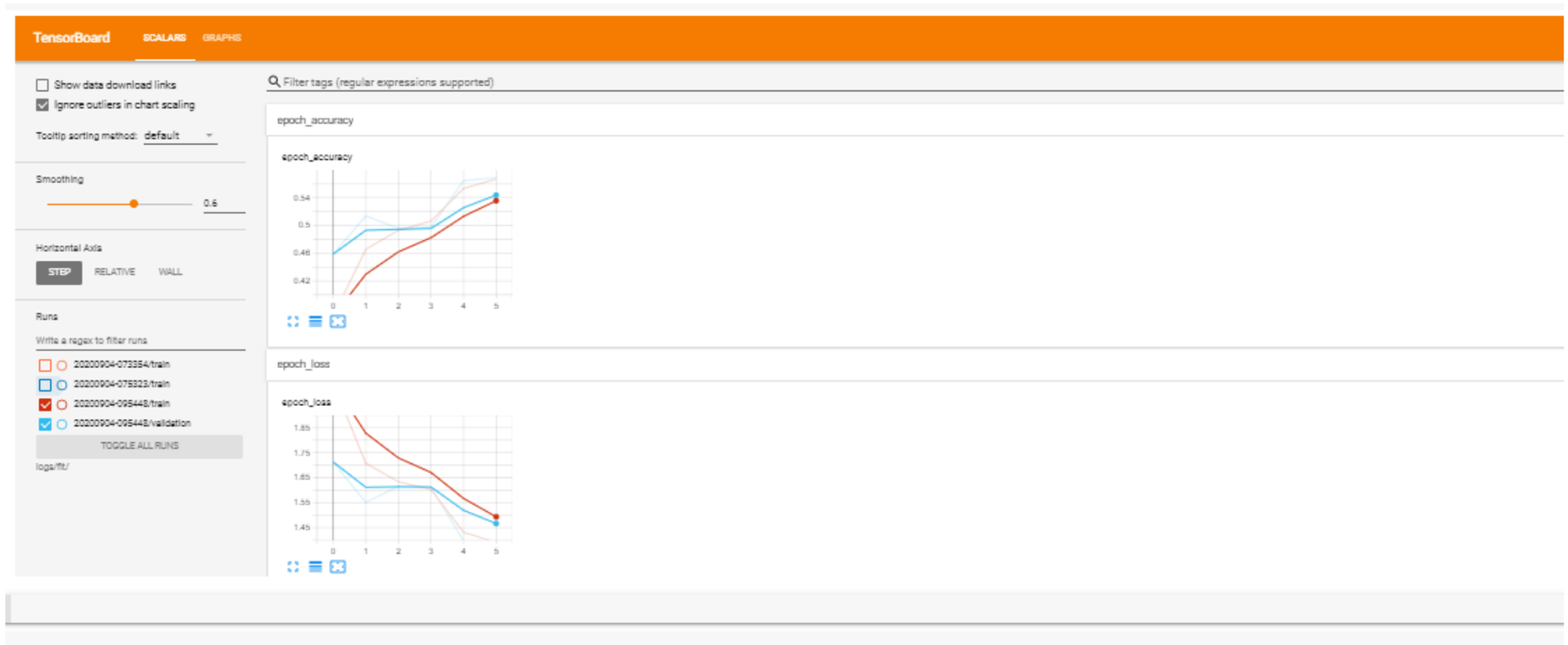
```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```

In [ ]:
```
os.chdir('/gdrive/My Drive')
```

In [ ]:
```
%tensorboard --logdir logs/fit/
```

In [ ]:
```
#Model 1 - results
from IPython.display import Image
Image(filename='Transfer_model1.PNG')
```

Out[24]:



Note - In Google colab, unfortunately session got timeout at 7th epoch, but model improved in each epoch , i got 0.58 accuracy

In Deeplearning we have n number of state of art best model like Xception, VGG16,VGG19 which gave good result for image c oncept of Transfer model is we going to use that architecture, in our task we took VGG16 which gave good result for image 1000 categorical model, we going to using this architecture for our image prediction problem.

```
    include_top  = False  - we are not going to use 1000 categorical we going to remove top layer and replace by dense lay
16 catgorical
    layer.trainable = False -  layer are trained, if we give system should need high level of configuration
```

Takeaway Transfer model - use the model which already gave good result for Image classification probelm

In [ ]: