

DetectionofLungInfection

May 22, 2022

```
[ ]: !unzip Dataset_Detection_of_Lung_Infection.zip
```

Archive: Dataset_Detection_of_Lung_Infection.zip

```
  creating: data/test/  
  creating: data/test/healthy/  
inflating: data/test/healthy/0101.jpeg  
inflating: data/test/healthy/0102.jpeg  
inflating: data/test/healthy/0103.jpeg  
inflating: data/test/healthy/0105.jpeg  
inflating: data/test/healthy/0106.jpeg  
inflating: data/test/healthy/0107.jpeg  
inflating: data/test/healthy/0108.jpeg  
inflating: data/test/healthy/0109.jpeg  
inflating: data/test/healthy/0110.jpeg  
inflating: data/test/healthy/0111.jpeg  
inflating: data/test/healthy/0112.jpeg  
inflating: data/test/healthy/0114.jpeg  
inflating: data/test/healthy/0115.jpeg  
inflating: data/test/healthy/0116.jpeg  
inflating: data/test/healthy/0117.jpeg  
inflating: data/test/healthy/0118.jpeg  
inflating: data/test/healthy/0119.jpeg  
inflating: data/test/healthy/0120.jpeg  
inflating: data/test/healthy/0121.jpeg  
inflating: data/test/healthy/0122.jpeg  
  creating: data/test/Type 1 disease/  
inflating: data/test/Type 1 disease/0100.jpeg  
inflating: data/test/Type 1 disease/0102.jpeg  
inflating: data/test/Type 1 disease/0105.png  
inflating: data/test/Type 1 disease/0106.jpeg  
inflating: data/test/Type 1 disease/0108.jpeg  
inflating: data/test/Type 1 disease/0111.jpg  
inflating: data/test/Type 1 disease/0112.jpg  
inflating: data/test/Type 1 disease/0113.jpg  
inflating: data/test/Type 1 disease/0115.jpeg  
inflating: data/test/Type 1 disease/0118.jpeg  
inflating: data/test/Type 1 disease/0119.jpeg  
inflating: data/test/Type 1 disease/0120.jpg
```

```

inflating: data/test/Type 1 disease/094.png
inflating: data/test/Type 1 disease/096.png
inflating: data/test/Type 1 disease/098.jpeg
inflating: data/test/Type 1
disease/auntminnie-2020_01_31_20_24_2322_2020_01_31_x-ray_coronavirus_US.jpg
inflating: data/test/Type 1
disease/auntminnie-a-2020_01_28_23_51_6665_2020_01_28_Vietnam_coronavirus.jpeg
inflating: data/test/Type 1
disease/auntminnie-b-2020_01_28_23_51_6665_2020_01_28_Vietnam_coronavirus.jpeg
inflating: data/test/Type 1
disease/auntminnie-c-2020_01_28_23_51_6665_2020_01_28_Vietnam_coronavirus.jpeg
inflating: data/test/Type 1
disease/auntminnie-d-2020_01_28_23_51_6665_2020_01_28_Vietnam_coronavirus.jpeg
inflating: data/test/Type 1 disease/COVID-00003b.jpg
inflating: data/test/Type 1 disease/COVID-00012.jpg
inflating: data/test/Type 1 disease/COVID-00022.jpg
inflating: data/test/Type 1 disease/COVID-00033.jpg
inflating: data/test/Type 1 disease/COVID-00037.jpg
inflating: data/test/Type 1 disease/radiopaedia-2019-novel-coronavirus-
infected-pneumonia.jpg
creating: data/test/Type 2 disease/
inflating: data/test/Type 2 disease/0101.jpeg
inflating: data/test/Type 2 disease/0102.jpeg
inflating: data/test/Type 2 disease/0103.jpeg
inflating: data/test/Type 2 disease/0104.jpeg
inflating: data/test/Type 2 disease/0105.jpeg
inflating: data/test/Type 2 disease/0106.jpeg
inflating: data/test/Type 2 disease/0107.jpeg
inflating: data/test/Type 2 disease/0108.jpeg
inflating: data/test/Type 2 disease/0109.jpeg
inflating: data/test/Type 2 disease/0110.jpeg
inflating: data/test/Type 2 disease/0111.jpeg
inflating: data/test/Type 2 disease/0112.jpeg
inflating: data/test/Type 2 disease/0113.jpeg
inflating: data/test/Type 2 disease/0114.jpeg
inflating: data/test/Type 2 disease/0115.jpeg
inflating: data/test/Type 2 disease/0116.jpeg
inflating: data/test/Type 2 disease/0117.jpeg
inflating: data/test/Type 2 disease/0118.jpeg
inflating: data/test/Type 2 disease/0119.jpeg
inflating: data/test/Type 2 disease/0120.jpeg
creating: data/train/
creating: data/train/Healthy/
inflating: data/train/Healthy/01.jpeg
inflating: data/train/Healthy/010.jpeg
inflating: data/train/Healthy/011.jpeg
inflating: data/train/Healthy/012.jpeg
inflating: data/train/Healthy/013.jpeg

```

inflating: data/train/Healthy/014.jpeg
inflating: data/train/Healthy/015.jpeg
inflating: data/train/Healthy/016.jpeg
inflating: data/train/Healthy/017.jpeg
inflating: data/train/Healthy/018.jpeg
inflating: data/train/Healthy/019.jpeg
inflating: data/train/Healthy/02.jpeg
inflating: data/train/Healthy/020.jpeg
inflating: data/train/Healthy/021.jpeg
inflating: data/train/Healthy/022.jpeg
inflating: data/train/Healthy/023.jpeg
inflating: data/train/Healthy/024.jpeg
inflating: data/train/Healthy/025.jpeg
inflating: data/train/Healthy/03.jpeg
inflating: data/train/Healthy/04.jpeg
inflating: data/train/Healthy/05.jpeg
inflating: data/train/Healthy/050.jpeg
inflating: data/train/Healthy/051.jpeg
inflating: data/train/Healthy/052.jpeg
inflating: data/train/Healthy/053.jpeg
inflating: data/train/Healthy/054.jpeg
inflating: data/train/Healthy/055.jpeg
inflating: data/train/Healthy/056.jpeg
inflating: data/train/Healthy/057.jpeg
inflating: data/train/Healthy/058.jpeg
inflating: data/train/Healthy/059.jpeg
inflating: data/train/Healthy/06.jpeg
inflating: data/train/Healthy/060.jpeg
inflating: data/train/Healthy/061.jpeg
inflating: data/train/Healthy/062.jpeg
inflating: data/train/Healthy/063.jpeg
inflating: data/train/Healthy/064.jpeg
inflating: data/train/Healthy/065.jpeg
inflating: data/train/Healthy/066.jpeg
inflating: data/train/Healthy/067.jpeg
inflating: data/train/Healthy/068.jpeg
inflating: data/train/Healthy/069.jpeg
inflating: data/train/Healthy/07.jpeg
inflating: data/train/Healthy/070.jpeg
inflating: data/train/Healthy/071.jpeg
inflating: data/train/Healthy/072.jpeg
inflating: data/train/Healthy/073.jpeg
inflating: data/train/Healthy/074.jpeg
inflating: data/train/Healthy/075.jpeg
inflating: data/train/Healthy/076.jpeg
inflating: data/train/Healthy/077.jpeg
inflating: data/train/Healthy/079.jpeg
inflating: data/train/Healthy/08.jpeg

inflating: data/train/Healthy/080.jpeg
inflating: data/train/Healthy/081.jpeg
inflating: data/train/Healthy/082.jpeg
inflating: data/train/Healthy/083.jpeg
inflating: data/train/Healthy/084.jpeg
inflating: data/train/Healthy/085.jpeg
inflating: data/train/Healthy/086.jpeg
inflating: data/train/Healthy/087.jpeg
inflating: data/train/Healthy/088.jpeg
inflating: data/train/Healthy/09.jpeg
inflating: data/train/Healthy/091.jpeg
inflating: data/train/Healthy/092.jpeg
inflating: data/train/Healthy/093.jpeg
inflating: data/train/Healthy/094.jpeg
inflating: data/train/Healthy/095.jpeg
inflating: data/train/Healthy/096.jpeg
inflating: data/train/Healthy/097.jpeg
creating: data/train/Type 1 disease/
inflating: data/train/Type 1 disease/01.jpeg
inflating: data/train/Type 1 disease/010.png
inflating: data/train/Type 1 disease/012.jpeg
inflating: data/train/Type 1 disease/015.jpg
inflating: data/train/Type 1 disease/019.png
inflating: data/train/Type 1 disease/02.jpeg
inflating: data/train/Type 1 disease/020.jpg
inflating: data/train/Type 1 disease/021.jpeg
inflating: data/train/Type 1 disease/022.jpeg
inflating: data/train/Type 1 disease/024.jpeg
inflating: data/train/Type 1 disease/025.jpeg
inflating: data/train/Type 1 disease/026.jpeg
inflating: data/train/Type 1 disease/027.jpeg
inflating: data/train/Type 1 disease/03.jpeg
inflating: data/train/Type 1 disease/031.jpeg
inflating: data/train/Type 1 disease/032.jpeg
inflating: data/train/Type 1 disease/033.png
inflating: data/train/Type 1 disease/039.jpeg
inflating: data/train/Type 1 disease/04.png
inflating: data/train/Type 1 disease/040.jpeg
inflating: data/train/Type 1 disease/041.jpeg
inflating: data/train/Type 1 disease/042.jpeg
inflating: data/train/Type 1 disease/043.jpeg
inflating: data/train/Type 1 disease/044.jpeg
inflating: data/train/Type 1 disease/045.jpeg
inflating: data/train/Type 1 disease/046.jpeg
inflating: data/train/Type 1 disease/047.jpeg
inflating: data/train/Type 1 disease/048.jpeg
inflating: data/train/Type 1 disease/049.jpeg
inflating: data/train/Type 1 disease/050.jpeg

inflating: data/train/Type 1 disease/051.jpeg
inflating: data/train/Type 1 disease/052.jpeg
inflating: data/train/Type 1 disease/053.jpeg
inflating: data/train/Type 1 disease/054.jpeg
inflating: data/train/Type 1 disease/055.jpeg
inflating: data/train/Type 1 disease/056.jpg
inflating: data/train/Type 1 disease/057.jpeg
inflating: data/train/Type 1 disease/058.jpeg
inflating: data/train/Type 1 disease/059.jpeg
inflating: data/train/Type 1 disease/06.jpeg
inflating: data/train/Type 1 disease/060.jpeg
inflating: data/train/Type 1 disease/061.jpg
inflating: data/train/Type 1 disease/062.jpeg
inflating: data/train/Type 1 disease/064.jpg
inflating: data/train/Type 1 disease/065.jpeg
inflating: data/train/Type 1 disease/067.jpg
inflating: data/train/Type 1 disease/068.jpg
inflating: data/train/Type 1 disease/069.jpg
inflating: data/train/Type 1 disease/07.jpg
inflating: data/train/Type 1 disease/071.jpg
inflating: data/train/Type 1 disease/072.jpeg
inflating: data/train/Type 1 disease/073.jpg
inflating: data/train/Type 1 disease/074.jpg
inflating: data/train/Type 1 disease/076.jpg
inflating: data/train/Type 1 disease/078.jpeg
inflating: data/train/Type 1 disease/079.jpeg
inflating: data/train/Type 1 disease/08.jpeg
inflating: data/train/Type 1 disease/080.jpg
inflating: data/train/Type 1 disease/081.jpeg
inflating: data/train/Type 1 disease/082.jpg
inflating: data/train/Type 1 disease/083.jpeg
inflating: data/train/Type 1 disease/084.jpeg
inflating: data/train/Type 1 disease/085.jpeg
inflating: data/train/Type 1 disease/086.jpg
inflating: data/train/Type 1 disease/088.jpeg
inflating: data/train/Type 1 disease/089.jpg
inflating: data/train/Type 1 disease/09.png
inflating: data/train/Type 1 disease/090.jpeg
inflating: data/train/Type 1 disease/091.jpg
inflating: data/train/Type 1 disease/092.png
inflating: data/train/Type 1 disease/COVID-00001.jpg
inflating: data/train/Type 1 disease/COVID-00002.jpg
inflating: data/train/Type 1 disease/COVID-00003a.jpg
inflating: data/train/Type 1 disease/COVID-00003b.jpg
inflating: data/train/Type 1 disease/COVID-00004.jpg
inflating: data/train/Type 1 disease/COVID-00005.jpg
inflating: data/train/Type 1 disease/COVID-00006.jpg
inflating: data/train/Type 1 disease/COVID-00007.jpg

inflating: data/train/Type 1 disease/COVID-00008.jpg
inflating: data/train/Type 1 disease/COVID-00009.jpg
inflating: data/train/Type 1 disease/COVID-00010.jpg
inflating: data/train/Type 1 disease/COVID-00011.jpg
inflating: data/train/Type 1 disease/COVID-00012.jpg
inflating: data/train/Type 1 disease/COVID-00013a.jpg
inflating: data/train/Type 1 disease/COVID-00013b.jpg
inflating: data/train/Type 1 disease/COVID-00014.jpg
inflating: data/train/Type 1 disease/COVID-00015a.png
inflating: data/train/Type 1 disease/COVID-00015b.png
inflating: data/train/Type 1 disease/COVID-00016.jpg
inflating: data/train/Type 1 disease/COVID-00017.jpg
inflating: data/train/Type 1 disease/COVID-00018.jpg
inflating: data/train/Type 1 disease/COVID-00019.jpg
inflating: data/train/Type 1 disease/COVID-00020.jpg
inflating: data/train/Type 1 disease/COVID-00021.jpg
inflating: data/train/Type 1 disease/COVID-00022.jpg
inflating: data/train/Type 1 disease/COVID-00023.jpg
inflating: data/train/Type 1 disease/COVID-00024.jpg
inflating: data/train/Type 1 disease/COVID-00025.jpg
inflating: data/train/Type 1 disease/COVID-00026.jpg
inflating: data/train/Type 1 disease/COVID-00027.jpg
inflating: data/train/Type 1 disease/COVID-00028.jpg
inflating: data/train/Type 1 disease/COVID-00029.jpg
inflating: data/train/Type 1 disease/COVID-00030.jpg
inflating: data/train/Type 1 disease/COVID-00031.jpg
inflating: data/train/Type 1 disease/COVID-00032.jpg
inflating: data/train/Type 1 disease/COVID-00033.jpg
inflating: data/train/Type 1 disease/COVID-00034.jpg
inflating: data/train/Type 1 disease/COVID-00035.jpg
inflating: data/train/Type 1 disease/COVID-00036.jpg
inflating: data/train/Type 1 disease/COVID-00037.jpg
inflating: data/train/Type 1 disease/COVID-00038.jpg
creating: data/train/Type 2 disease/
inflating: data/train/Type 2 disease/01.jpeg
inflating: data/train/Type 2 disease/010.jpeg
inflating: data/train/Type 2 disease/011.jpeg
inflating: data/train/Type 2 disease/012.jpeg
inflating: data/train/Type 2 disease/013.jpeg
inflating: data/train/Type 2 disease/016.jpeg
inflating: data/train/Type 2 disease/018.jpeg
inflating: data/train/Type 2 disease/019.jpeg
inflating: data/train/Type 2 disease/02.jpeg
inflating: data/train/Type 2 disease/020.jpeg
inflating: data/train/Type 2 disease/021.jpeg
inflating: data/train/Type 2 disease/022.jpeg
inflating: data/train/Type 2 disease/023.jpeg
inflating: data/train/Type 2 disease/024.jpeg


```

inflating: data/train/Type 2 disease/081.jpeg
inflating: data/train/Type 2 disease/082.jpeg
inflating: data/train/Type 2 disease/083.jpeg
inflating: data/train/Type 2 disease/084.jpeg
inflating: data/train/Type 2 disease/09.jpeg
inflating: data/train/Type 2 disease/094.jpeg
inflating: data/train/Type 2 disease/095.jpeg
inflating: data/train/Type 2 disease/096.jpeg

```

```

[ ]: from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.layers import Dense, Dropout,
    Flatten, Conv2D, BatchNormalization, Activation, MaxPooling2D , MaxPool2D
from tensorflow.keras.models import Model, Sequential

from tensorflow.keras.applications import MobileNet , DenseNet121
    ↪ # transfer learning

from sklearn.metrics import classification_report

```

```

[ ]: import os

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from tensorflow.keras.preprocessing.image import load_img, img_to_array

```

```

[ ]: from tensorflow.keras.preprocessing.image import ImageDataGenerator

```

```

[ ]: train_folder_path = r'data/train'
test_folder_path = r'data/test'

```

Check diectories in train folder

```

[ ]: os.listdir(train_folder_path)

```

```

[ ]: ['Type 1 disease', 'Healthy', 'Type 2 disease']

```

Check the way to access image

```

[ ]: os.listdir(r"data/train/Type 1 disease")[2]

```

```

[ ]: '024.jpeg'

```

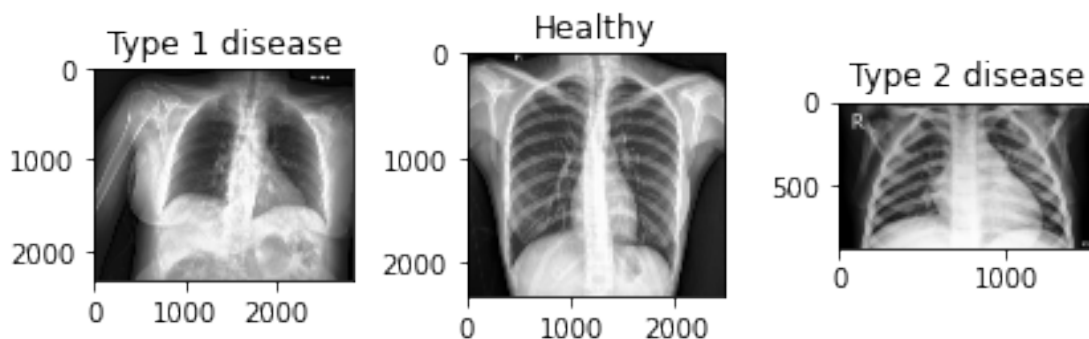
Plot the sample images for all the classes-----

```
[ ]: plot_no = 1
for each_folder in os.listdir(train_folder_path):
    complete_path_of_folder = os.path.join(train_folder_path ,each_folder)
    print(complete_path_of_folder)
    #making image path
    image_path = os.path.join(complete_path_of_folder, os.listdir(
    ↪complete_path_of_folder)[1])
    #loading image
    image = load_img(image_path)
    plt.subplot(1,3,plot_no)
    plt.imshow(image ,cmap = "gray")
    plt.title(each_folder)

    plot_no += 1

plt.tight_layout()
plt.show()
```

data/train/Type 1 disease
data/train/Healthy
data/train/Type 2 disease



Plot the distribution of images across the classes

```
[ ]: train_folder_path = r'data/train'
os.listdir(train_folder_path)
```

```
[ ]: ['Type 1 disease', 'Healthy', 'Type 2 disease']
```

```
[ ]: data_info_dataframe = pd.DataFrame({'Class' : None
    , 'Number of Images' : None} , index = range(3))
```

```
[ ]: data_info_dataframe
```

```
[ ]:      Class Number of Images
      0  None          None
      1  None          None
      2  None          None
```

For Training Data

```
[ ]: p = 0
      for each_folder in os.listdir(train_folder_path):
          data_info_dataframe.at[p , "Class"] = each_folder
          data_info_dataframe.at[p,"Number of Images"]=len(os.listdir(os.path.
          ↳join(train_folder_path, each_folder)))
          p = p+1
```

```
[ ]: data_info_dataframe
```

```
[ ]:      Class Number of Images
      0  Type 1 disease      111
      1      Healthy        70
      2  Type 2 disease      70
```

Cross checking

```
[ ]: len(os.listdir(r"data/train/Type 1 disease"))
```

```
[ ]: 111
```

```
[ ]: len(os.listdir(r"data/train/Healthy"))
```

```
[ ]: 70
```

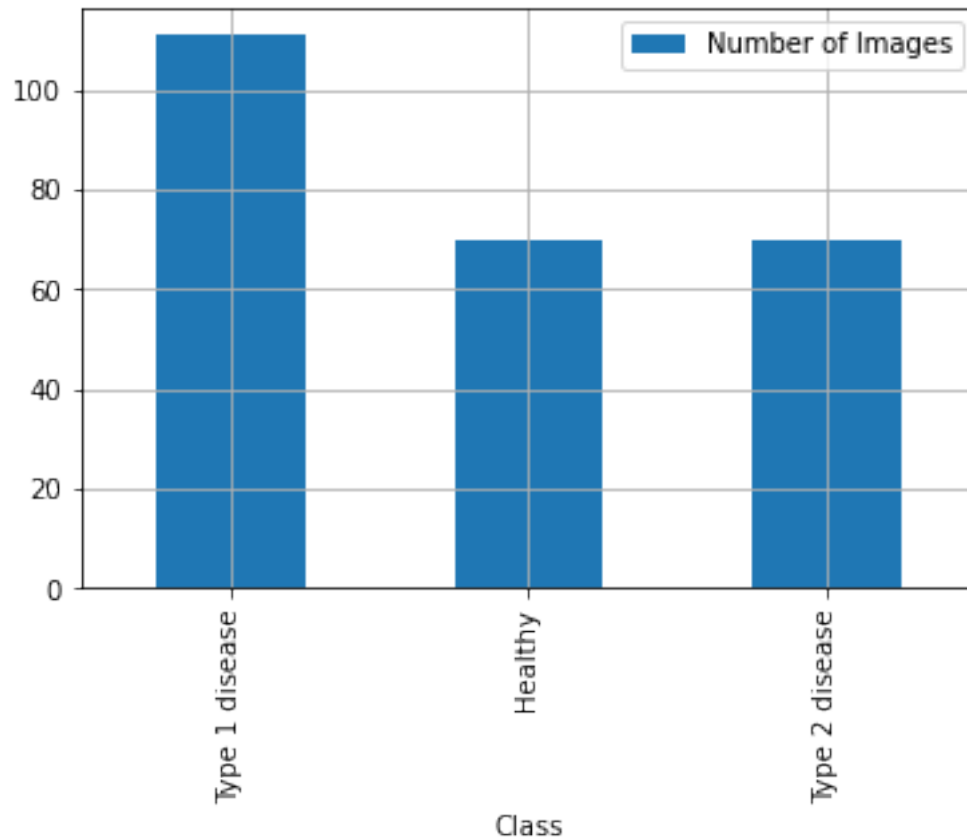
```
[ ]: data_info_dataframe.set_index("Class" , inplace = True)
      data_info_dataframe
```

```
[ ]:      Number of Images
      Class
      Type 1 disease      111
      Healthy            70
      Type 2 disease      70
```

```
[ ]: data_info_dataframe.plot(kind = 'bar')

      plt.grid(True)
      plt.xlabel("Class")
      plt.figure(figsize=(20,20))
```

```
[ ]: <Figure size 1440x1440 with 0 Axes>
```



<Figure size 1440x1440 with 0 Axes>

Plot the distribution of images across the classes

For Test Data-----

```
[ ]: test_data_info_dataframe = pd.DataFrame({'Class' : None , 'Number of Images' : None}, index = range(3))
```

```
[ ]: b = 0
for test_folder in os.listdir(test_folder_path):
    test_data_info_dataframe.at[b , 'Number of Images'] = len(os.listdir(os.path.join(test_folder_path, test_folder)))
    test_data_info_dataframe.at[b , 'Class'] = test_folder
    b = b + 1
```

```
[ ]: test_data_info_dataframe
```

```
[ ]:      Class Number of Images
0  Type 1 disease             26
```

```
1      healthy      20
2 Type 2 disease    20
```

```
[ ]: len(os.listdir(r"data/test/Type 1 disease"))
```

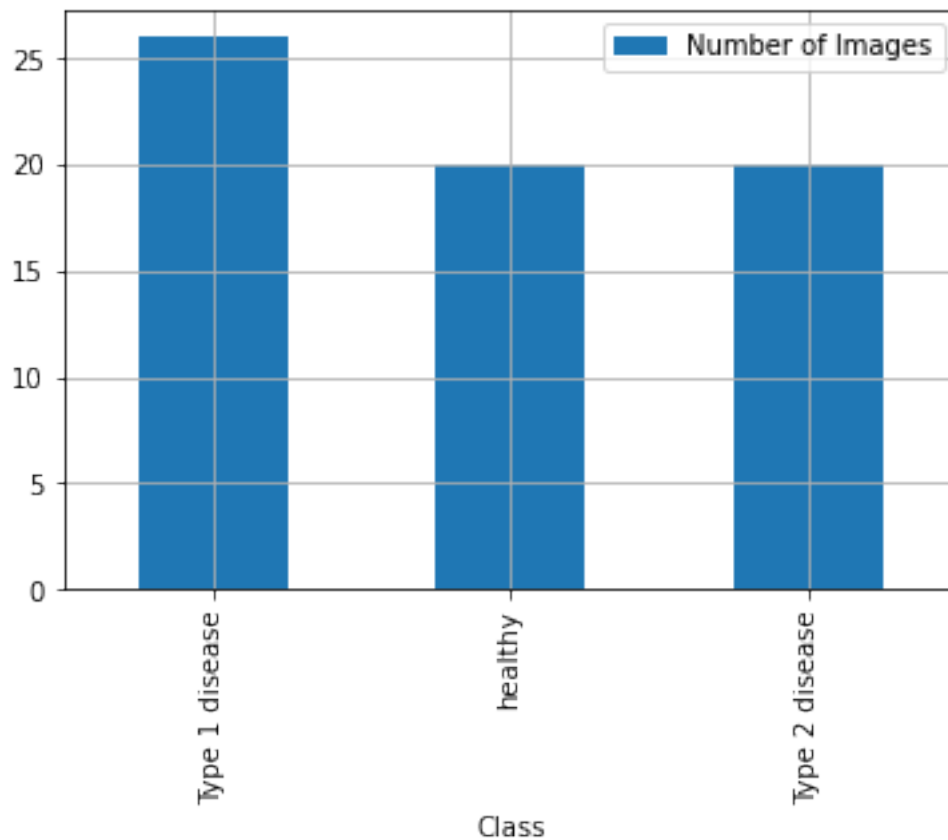
```
[ ]: 26
```

```
[ ]: test_data_info_dataframe.set_index("Class" , inplace = True)
```

```
[ ]: test_data_info_dataframe
```

```
[ ]:      Number of Images
Class
Type 1 disease      26
healthy             20
Type 2 disease      20
```

```
[ ]: test_data_info_dataframe.plot(kind = 'bar')
plt.xlabel("Class")
plt.grid(True)
```



Build a data augmentation for train data to create new data with translation, rescale and flip, and rotation transformations

FOR TRAINING-----

```
[ ]: traindataGeneratorObj = ImageDataGenerator(rescale = 1.0/225.0 ,
        width_shift_range = 0.1 ,
        height_shift_range = 0.1 ,
        rotation_range = 10 ,
        horizontal_flip = True)

[ ]: train_obj_gen = traindataGeneratorObj.flow_from_directory(train_folder_path ,
        class_mode='categorical',
        target_size = ( 100 ,100) ,
        batch_size=7,
        shuffle=True ,
        color_mode='grayscale')
```

Found 251 images belonging to 3 classes.

```
[ ]: train_obj_gen.class_indices
```

```
[ ]: {'Healthy': 0, 'Type 1 disease': 1, 'Type 2 disease': 2}
```

```
[ ]:
```

FOR TESTING-----

```
[ ]: testdataGeneratorObj = ImageDataGenerator(rescale = 1.0/255.0)

[ ]: test_obj_gen = testdataGeneratorObj.flow_from_directory(test_folder_path ,
        class_mode='categorical',
        target_size = ( 100 ,100) ,
        batch_size=7,
        shuffle=True ,
        color_mode='grayscale')
```

Found 66 images belonging to 3 classes.

```
[ ]: test_obj_gen.class_indices
```

```
[ ]: {'Healthy': 0, 'Type 1 disease': 1, 'Type 2 disease': 2}
```

Add convolutional layers with different filters, max pool layers, dropout layers, and batch normalization layers

```
[ ]: model1 = Sequential()
```

Use Relu as an activation function

```
[ ]: model1.add( Conv2D(164 ,(3,3) , input_shape = (100, 100 , 1 )))  
model1.add(Activation ('relu'))  
model1.add(BatchNormalization())  
model1.add(MaxPooling2D(pool_size=(3,3)))  
model1.add(Dropout(0.1))
```

```
[ ]: model1.add(Conv2D(128,(3,3)))  
model1.add(BatchNormalization())  
model1.add(Activation('relu'))  
model1.add(MaxPooling2D(pool_size= (3,3)))  
model1.add(Dropout(0.1))
```

```
[ ]: model1.add(Conv2D(512,(3,3)))  
model1.add(BatchNormalization())  
model1.add(Activation('relu'))  
model1.add(MaxPooling2D(pool_size= (3,3)))  
model1.add(Dropout(0.1))
```

```
[ ]: model1.add(Flatten())
```

```
[ ]: model1.add(Dense(2048))  
model1.add(BatchNormalization())  
model1.add(Activation('relu'))  
model1.add(Dropout(0.1))
```

```
[ ]: model1.add(Dense(3, activation = 'softmax')) # 7 --> total number of classes  
  
print(model1.summary())
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 98, 98, 164)	1640
activation (Activation)	(None, 98, 98, 164)	0
batch_normalization (Batch Normalization)	(None, 98, 98, 164)	656
max_pooling2d (MaxPooling2D)	(None, 32, 32, 164)	0
dropout (Dropout)	(None, 32, 32, 164)	0

conv2d_1 (Conv2D)	(None, 30, 30, 128)	189056
batch_normalization_1 (Batch Normalization)	(None, 30, 30, 128)	512
activation_1 (Activation)	(None, 30, 30, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 128)	0
dropout_1 (Dropout)	(None, 10, 10, 128)	0
conv2d_2 (Conv2D)	(None, 8, 8, 512)	590336
batch_normalization_2 (Batch Normalization)	(None, 8, 8, 512)	2048
activation_2 (Activation)	(None, 8, 8, 512)	0
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 512)	0
dropout_2 (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 2048)	4196352
batch_normalization_3 (Batch Normalization)	(None, 2048)	8192
activation_3 (Activation)	(None, 2048)	0
dropout_3 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 3)	6147

=====

Total params: 4,994,939
Trainable params: 4,989,235
Non-trainable params: 5,704

None

Take the loss function as categorical cross-entropy
Take rmsprop as an optimizer

```
[ ]: model1.compile(optimizer = 'rmsprop'
                    , loss = 'categorical_crossentropy' , metrics = ['accuracy'])
```

```
[ ]: EarlyStoppingObj = EarlyStopping(monitor = 'val_loss'
                                     ,min_delta = 0.1 ,mode = 'min' ,   patience = 2
                                     ↪,restore_best_weights = True)
```

Use early stopping with the patience of two epochs and monitor the validation loss or accuracy

Try with ten numbers epoch

```
[ ]: history = model1.fit(train_obj_gen ,
                          steps_per_epoch = train_obj_gen.n //train_obj_gen.batch_size ,
                          epochs = 10 ,
                          validation_data = test_obj_gen ,
                          validation_steps = test_obj_gen.n // test_obj_gen .batch_size ,
                          callbacks = [EarlyStoppingObj ,]
                          )
```

Epoch 1/10

```
35/35 [=====] - 14s 397ms/step - loss: 0.8977 -
accuracy: 0.7951 - val_loss: 8.5095 - val_accuracy: 0.3016
```

Epoch 2/10

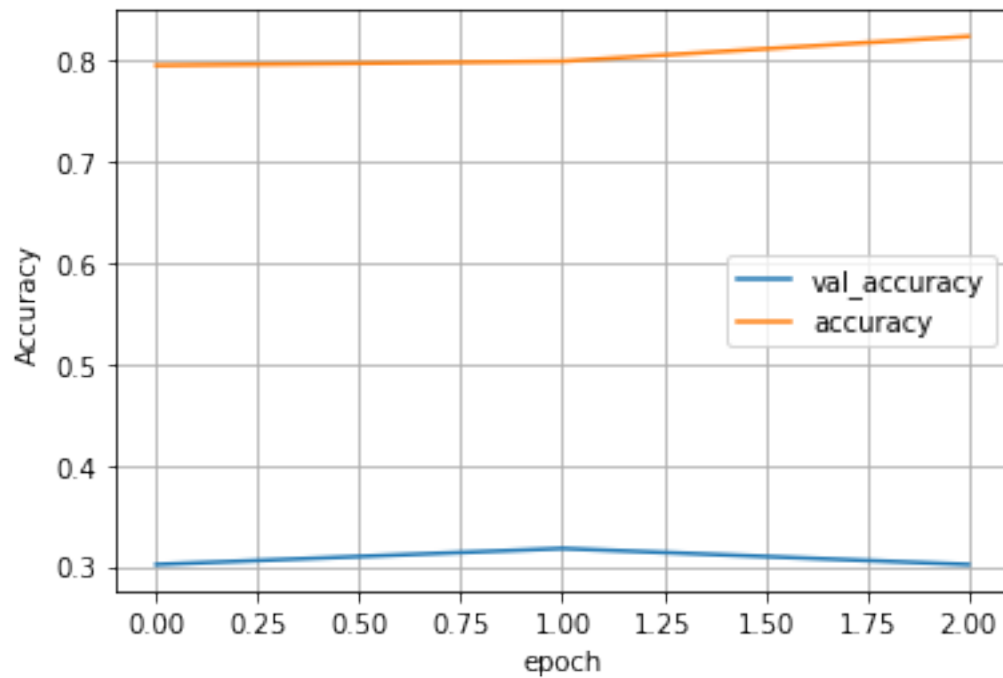
```
35/35 [=====] - 13s 383ms/step - loss: 0.8035 -
accuracy: 0.7992 - val_loss: 10.3642 - val_accuracy: 0.3175
```

Epoch 3/10

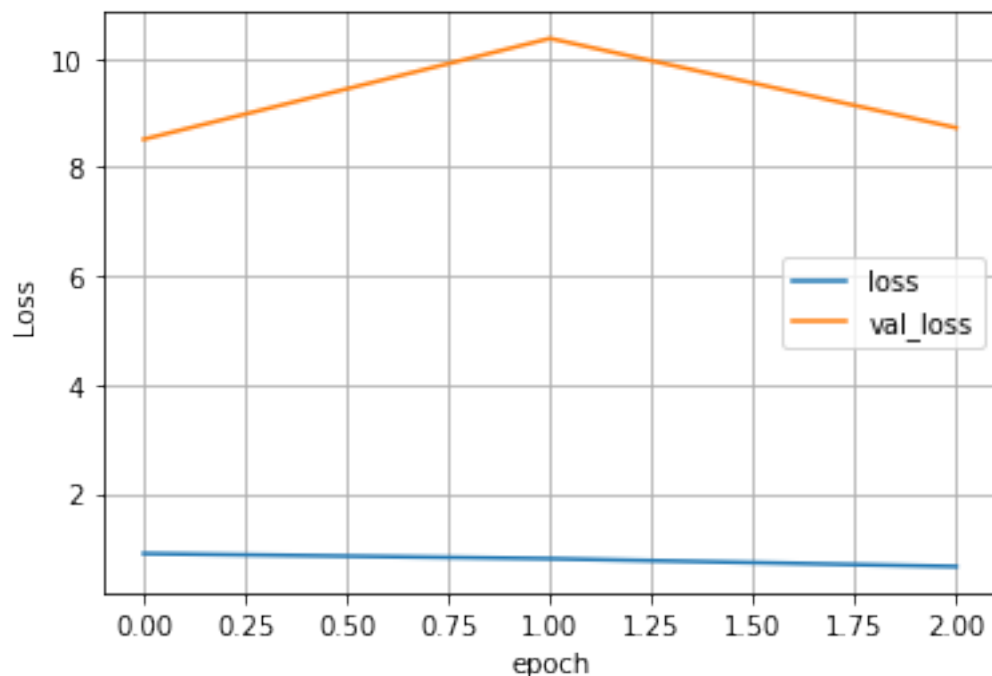
```
35/35 [=====] - 14s 401ms/step - loss: 0.6592 -
accuracy: 0.8238 - val_loss: 8.7217 - val_accuracy: 0.3016
```

Plot the training and validation accuracy, and the loss-----

```
[ ]: plt.plot(history.history['val_accuracy'] ,label = 'val_accuracy')
plt.plot(history.history['accuracy'] , label = 'accuracy')
plt.xlabel('epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```

```
[ ]: plt.plot(history.history['loss'], label = 'loss')
plt.plot(history.history['val_loss'], label = 'val_loss')
plt.xlabel('epoch')
plt.ylabel('Loss')
plt.grid(True)
plt.legend()
plt.show()
```



```
[ ]: prediction = model1.predict(test_obj_gen )
```

```
[ ]: prediction
```

```
[ ]: array([[2.90877379e-05, 9.99967933e-01, 2.98682471e-06],
 [5.02848416e-05, 9.99942064e-01, 7.62153104e-06],
 [3.22023516e-05, 9.99964237e-01, 3.51922563e-06],
 [4.60961273e-05, 9.99950051e-01, 3.83411725e-06],
 [4.16243438e-06, 9.99994755e-01, 1.04722903e-06],
 [7.73056036e-06, 9.99991655e-01, 6.49931053e-07],
 [4.55360896e-05, 9.99949574e-01, 4.86530962e-06],
 [1.95024786e-05, 9.99976873e-01, 3.57094927e-06],
 [7.72795102e-05, 9.99914885e-01, 7.86280089e-06],
 [6.95389972e-05, 9.99901652e-01, 2.88196388e-05],
 [1.05316094e-05, 9.99987364e-01, 2.10293388e-06],
 [5.02612711e-05, 9.99929070e-01, 2.06253371e-05],
 [3.33056632e-05, 9.99956965e-01, 9.75947842e-06],
 [1.52758148e-05, 9.99978423e-01, 6.35606330e-06],
 [4.98339477e-06, 9.99993205e-01, 1.81796020e-06],
 [3.27773887e-05, 9.99964356e-01, 2.84261614e-06],
 [4.98145964e-05, 9.99944925e-01, 5.27606426e-06],
 [4.39173209e-06, 9.99994159e-01, 1.38132157e-06],
 [3.93410301e-05, 9.99943018e-01, 1.75852547e-05],
 [7.30024694e-05, 9.99923110e-01, 3.98343718e-06],
 [2.91448146e-06, 9.99995708e-01, 1.39862755e-06],
```

```
[4.16859511e-05, 9.99953985e-01, 4.26556790e-06],
[1.43291766e-06, 9.99996424e-01, 2.11633596e-06],
[1.00244424e-05, 9.99989033e-01, 9.33395256e-07],
[6.67995801e-06, 9.99992490e-01, 8.32484204e-07],
[1.82717777e-05, 9.99976754e-01, 4.97574274e-06],
[4.89239437e-06, 9.99993563e-01, 1.52791711e-06],
[6.65901325e-06, 9.99991536e-01, 1.75445530e-06],
[3.42212552e-05, 9.99960661e-01, 5.09176289e-06],
[4.25879698e-05, 9.99952674e-01, 4.78412449e-06],
[1.00997504e-05, 9.99988914e-01, 9.03019213e-07],
[2.97632487e-05, 9.99964952e-01, 5.23590916e-06],
[2.28831359e-05, 9.99974966e-01, 2.09994960e-06],
[4.33337300e-06, 9.99993920e-01, 1.75322862e-06],
[4.51131473e-06, 9.99994159e-01, 1.25439669e-06],
[6.49022377e-06, 9.99992132e-01, 1.44674289e-06],
[3.37461570e-05, 9.99960184e-01, 6.08004893e-06],
[4.98164336e-05, 9.99942660e-01, 7.52946698e-06],
[8.80618245e-06, 9.99990463e-01, 7.34085404e-07],
[1.77675938e-05, 9.99979854e-01, 2.43827617e-06],
[1.05606850e-05, 9.99988198e-01, 1.20952291e-06],
[2.49274945e-06, 9.99995351e-01, 2.13781163e-06],
[8.02102932e-06, 9.99990940e-01, 1.12150121e-06],
[6.20545688e-05, 9.99919653e-01, 1.81956348e-05],
[5.76693456e-05, 9.99934435e-01, 7.88026591e-06],
[7.35011508e-06, 9.99991894e-01, 6.99026714e-07],
[1.40470565e-05, 9.99980927e-01, 4.97523206e-06],
[5.04428426e-05, 9.99943495e-01, 6.05671767e-06],
[1.59393749e-05, 9.99981403e-01, 2.67408586e-06],
[2.16755834e-05, 9.99973774e-01, 4.50596872e-06],
[9.46578257e-06, 9.99989629e-01, 9.68706445e-07],
[2.50806806e-06, 9.99995351e-01, 2.13058206e-06],
[1.55142698e-05, 9.99979258e-01, 5.24785173e-06],
[3.94721028e-05, 9.99953270e-01, 7.27204269e-06],
[1.91126655e-05, 9.99976754e-01, 4.23126312e-06],
[5.06648721e-05, 9.99939084e-01, 1.02427439e-05],
[2.09122700e-05, 9.99971986e-01, 7.13718919e-06],
[5.16385171e-06, 9.99993682e-01, 1.19398658e-06],
[1.07749574e-05, 9.99984741e-01, 4.51394044e-06],
[4.03456652e-05, 9.99948621e-01, 1.11241097e-05],
[4.51131473e-06, 9.99994159e-01, 1.25439669e-06],
[5.54073904e-06, 9.99993682e-01, 8.76387162e-07],
[2.82395722e-05, 9.99965072e-01, 6.71587668e-06],
[4.39819632e-05, 9.99946237e-01, 9.71729423e-06],
[2.18896821e-05, 9.99973178e-01, 4.85993360e-06],
[6.32512965e-05, 9.99926686e-01, 1.00334964e-05]], dtype=float32)
```

```
[ ]: pred_labels = np.argmax(prediction , axis = -1)
```

```
[ ]: test_obj_gen.classes
```

[illegible]

```
[ ]: len(test_obj_gen.classes)
```

[]: 66

```
[ ]: len(prediction)
```

[]: 66

Observe the precision, recall the F1-score for all classes for both grayscale and color models, and determine if the model's classes are good-----

```
[ ]: print(classification_report(test_obj_gen.classes, pred_labels ,
                                zero_division = 0))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	26
1	0.30	1.00	0.47	20
2	0.00	0.00	0.00	20
accuracy			0.30	66
macro avg	0.10	0.33	0.16	66
weighted avg	0.09	0.30	0.14	66

Transfer learning using mobile net----->

```
[ ]: import tensorflow as tf
tf.__version__
```

```
[ ]: '2.8.0'
```

Prepare data for the pre-trained mobile net model, with color mode as RGB

[illegible]

Target size required for mobilenet = (224, 224, 3)

```
[ ]: trainGenerator_Mobilenet = ImageDataGenerator_frommobilenet.  
    ↳flow_from_directory(train_folder_path ,  
                        target_size = (224 ,224) ,  
                        color_mode = "rgb",  
                        batch_size = 7 ,  
                        class_mode = 'categorical',  
                        shuffle = True  
                        )
```

Found 251 images belonging to 3 classes.

```
[ ]: trainGenerator_Mobilenet.class_indices
```

```
[ ]: {'Healthy': 0, 'Type 1 disease': 1, 'Type 2 disease': 2}
```

```
[ ]: test_data_generator_mobilenet = ImageDataGenerator(rescale = 1/255 )
```

```
[ ]: testGenerator_Mobilenet = test_data_generator_mobilenet.  
    ↳flow_from_directory(test_folder_path,  
                        target_size = (224 ,224) ,  
                        color_mode = 'rgb',  
                        batch_size = 7 ,  
                        class_mode = 'categorical' ,  
                        shuffle = True  
                        )
```

Found 66 images belonging to 3 classes.

```
[ ]: testGenerator_Mobilenet.class_indices
```

```
[ ]: {'Type 1 disease': 0, 'Type 2 disease': 1, 'healthy': 2}
```

Create an instance of a mobile net pre-trained model

```
[ ]: MobileNetLayer = tf.keras.applications.mobilenet.MobileNet(  
    input_shape=(224,224,3),  
    include_top= False,  
    weights='imagenet',  
    )
```

```
[ ]: MobileNetLayer.output
```

```
[ ]: <KerasTensor: shape=(None, 7, 7, 1024) dtype=float32 (created by layer  
    'conv_pw_13_relu')>
```

Add dense layer, dropout layer, batch normalization layer on the pre-trained model

```
[ ]: outputAfterTransferMobilenet = Flatten()(MobileNetLayer.output)

[ ]: outputAfterTransferMobilenet

[ ]: <KerasTensor: shape=(None, 50176) dtype=float32 (created by layer 'flatten_4')>

[ ]: outputAfterTransferMobilenet = Dense(500 ,activation = ↵
    ↪'relu')(outputAfterTransferMobilenet)

[ ]: outputAfterTransferMobilenet = Dropout(0.1)(outputAfterTransferMobilenet)

[ ]: outputAfterTransferMobilenet = ↵
    ↪BatchNormalization()(outputAfterTransferMobilenet)

[ ]: outputAfterTransferMobilenet

[ ]: <KerasTensor: shape=(None, 50176) dtype=float32 (created by layer
    'batch_normalization_5')>

[ ]: outputAfterTransferMobilenet = Dense(500 ,activation = ↵
    ↪'relu')(outputAfterTransferMobilenet)
outputAfterTransferMobilenet = Dropout(0.1)(outputAfterTransferMobilenet)
outputAfterTransferMobilenet = ↵
    ↪BatchNormalization()(outputAfterTransferMobilenet)

outputAfterTransferMobilenet = Dense(300 ,activation = ↵
    ↪'relu')(outputAfterTransferMobilenet )
outputAfterTransferMobilenet = Dropout(0.1)(outputAfterTransferMobilenet)
outputAfterTransferMobilenet = ↵
    ↪BatchNormalization()(outputAfterTransferMobilenet)
```

Create a final output layer with a SoftMax activation function

```
[ ]: predictions_output = Dense(3, activation = ↵
    ↪'softmax')(outputAfterTransferMobilenet)

[ ]: mobileNetModel = Model(inputs = MobileNetLayer.input , outputs = ↵
    ↪predictions_output)

[ ]: mobileNetModel.compile(optimizer = 'rmsprop'
    ,loss = 'categorical_crossentropy' , metrics = ['accuracy'])

[ ]: mobileNetModel.summary()
```

Model: "model_2"

Layer (type)	Output Shape	Param #
--------------	--------------	---------

=====		
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
conv1 (Conv2D)	(None, 112, 112, 32)	864
conv1_bn (BatchNormalizatio n)	(None, 112, 112, 32)	128
conv1_relu (ReLU)	(None, 112, 112, 32)	0
conv_dw_1 (DepthwiseConv2D)	(None, 112, 112, 32)	288
conv_dw_1_bn (BatchNormaliz ation)	(None, 112, 112, 32)	128
conv_dw_1_relu (ReLU)	(None, 112, 112, 32)	0
conv_pw_1 (Conv2D)	(None, 112, 112, 64)	2048
conv_pw_1_bn (BatchNormaliz ation)	(None, 112, 112, 64)	256
conv_pw_1_relu (ReLU)	(None, 112, 112, 64)	0
conv_pad_2 (ZeroPadding2D)	(None, 113, 113, 64)	0
conv_dw_2 (DepthwiseConv2D)	(None, 56, 56, 64)	576
conv_dw_2_bn (BatchNormaliz ation)	(None, 56, 56, 64)	256
conv_dw_2_relu (ReLU)	(None, 56, 56, 64)	0
conv_pw_2 (Conv2D)	(None, 56, 56, 128)	8192
conv_pw_2_bn (BatchNormaliz ation)	(None, 56, 56, 128)	512
conv_pw_2_relu (ReLU)	(None, 56, 56, 128)	0
conv_dw_3 (DepthwiseConv2D)	(None, 56, 56, 128)	1152
conv_dw_3_bn (BatchNormaliz ation)	(None, 56, 56, 128)	512
conv_dw_3_relu (ReLU)	(None, 56, 56, 128)	0
conv_pw_3 (Conv2D)	(None, 56, 56, 128)	16384

conv_pw_3_bn (BatchNormaliz ation)	(None, 56, 56, 128)	512
conv_pw_3_relu (ReLU)	(None, 56, 56, 128)	0
conv_pad_4 (ZeroPadding2D)	(None, 57, 57, 128)	0
conv_dw_4 (DepthwiseConv2D)	(None, 28, 28, 128)	1152
conv_dw_4_bn (BatchNormaliz ation)	(None, 28, 28, 128)	512
conv_dw_4_relu (ReLU)	(None, 28, 28, 128)	0
conv_pw_4 (Conv2D)	(None, 28, 28, 256)	32768
conv_pw_4_bn (BatchNormaliz ation)	(None, 28, 28, 256)	1024
conv_pw_4_relu (ReLU)	(None, 28, 28, 256)	0
conv_dw_5 (DepthwiseConv2D)	(None, 28, 28, 256)	2304
conv_dw_5_bn (BatchNormaliz ation)	(None, 28, 28, 256)	1024
conv_dw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65536
conv_pw_5_bn (BatchNormaliz ation)	(None, 28, 28, 256)	1024
conv_pw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pad_6 (ZeroPadding2D)	(None, 29, 29, 256)	0
conv_dw_6 (DepthwiseConv2D)	(None, 14, 14, 256)	2304
conv_dw_6_bn (BatchNormaliz ation)	(None, 14, 14, 256)	1024
conv_dw_6_relu (ReLU)	(None, 14, 14, 256)	0
conv_pw_6 (Conv2D)	(None, 14, 14, 512)	131072
conv_pw_6_bn (BatchNormaliz	(None, 14, 14, 512)	2048

ation)

conv_pw_6_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_7 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_7_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_dw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_7 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_7_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_pw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_8 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_8_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_dw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_8 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_8_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_pw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_9 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_9_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_dw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_9 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_9_bn (BatchNormaliz ation)	(None, 14, 14, 512)	2048
conv_pw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_10 (DepthwiseConv2D)	(None, 14, 14, 512)	4608

conv_dw_10_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_dw_10_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_10 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_10_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_pw_10_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_11 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_11_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_dw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_11 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_11_bn (BatchNormalization)	(None, 14, 14, 512)	2048
conv_pw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pad_12 (ZeroPadding2D)	(None, 15, 15, 512)	0
conv_dw_12 (DepthwiseConv2D)	(None, 7, 7, 512)	4608
conv_dw_12_bn (BatchNormalization)	(None, 7, 7, 512)	2048
conv_dw_12_relu (ReLU)	(None, 7, 7, 512)	0
conv_pw_12 (Conv2D)	(None, 7, 7, 1024)	524288
conv_pw_12_bn (BatchNormalization)	(None, 7, 7, 1024)	4096
conv_pw_12_relu (ReLU)	(None, 7, 7, 1024)	0
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216

conv_dw_13_bn (BatchNormali zation)	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
conv_pw_13_bn (BatchNormali zation)	(None, 7, 7, 1024)	4096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
flatten_4 (Flatten)	(None, 50176)	0
dense_20 (Dense)	(None, 500)	25088500
dropout_8 (Dropout)	(None, 500)	0
batch_normalization_8 (Batc hNormalization)	(None, 500)	2000
dense_21 (Dense)	(None, 500)	250500
dropout_9 (Dropout)	(None, 500)	0
batch_normalization_9 (Batc hNormalization)	(None, 500)	2000
dense_22 (Dense)	(None, 300)	150300
dropout_10 (Dropout)	(None, 300)	0
batch_normalization_10 (Bat chNormalization)	(None, 300)	1200
dense_23 (Dense)	(None, 3)	903

```
=====
Total params: 28,724,267
Trainable params: 28,699,779
Non-trainable params: 24,488
-----
```

Use early stopping with the patience of two epoch and call back function for preventing overfitting

Try with ten numbers epoch

Train the model using a generator and test the accuracy of the test data at every epoch

```
[ ]: estop = EarlyStopping( monitor = 'val_loss', min_delta = 0.1 ,
                           patience = 5 , mode = 'min' , restore_best_weights = True)
```

```
[ ]: histor_mobilenet = mobileNetModel.fit(trainGenerator_Mobilenet ,epochs = 10 ,
      validation_data = testGenerator_Mobilenet ,
      validation_steps = testGenerator_Mobilenet.n // testGenerator_Mobilenet.
      ↪batch_size ,
      steps_per_epoch = trainGenerator_Mobilenet.n // trainGenerator_Mobilenet.
      ↪batch_size ,
      callbacks = [estop ,]

      )
```

Epoch 1/10

35/35 [=====] - 17s 484ms/step - loss: 0.2875 - accuracy: 0.9139 - val_loss: 6.6293 - val_accuracy: 0.0000e+00

Epoch 2/10

35/35 [=====] - 15s 433ms/step - loss: 0.1919 - accuracy: 0.9344 - val_loss: 9.7952 - val_accuracy: 0.0000e+00

Epoch 3/10

35/35 [=====] - 15s 436ms/step - loss: 0.2254 - accuracy: 0.9344 - val_loss: 6.9204 - val_accuracy: 0.0476

Epoch 4/10

35/35 [=====] - 15s 438ms/step - loss: 0.2034 - accuracy: 0.9303 - val_loss: 7.9154 - val_accuracy: 0.0000e+00

Epoch 5/10

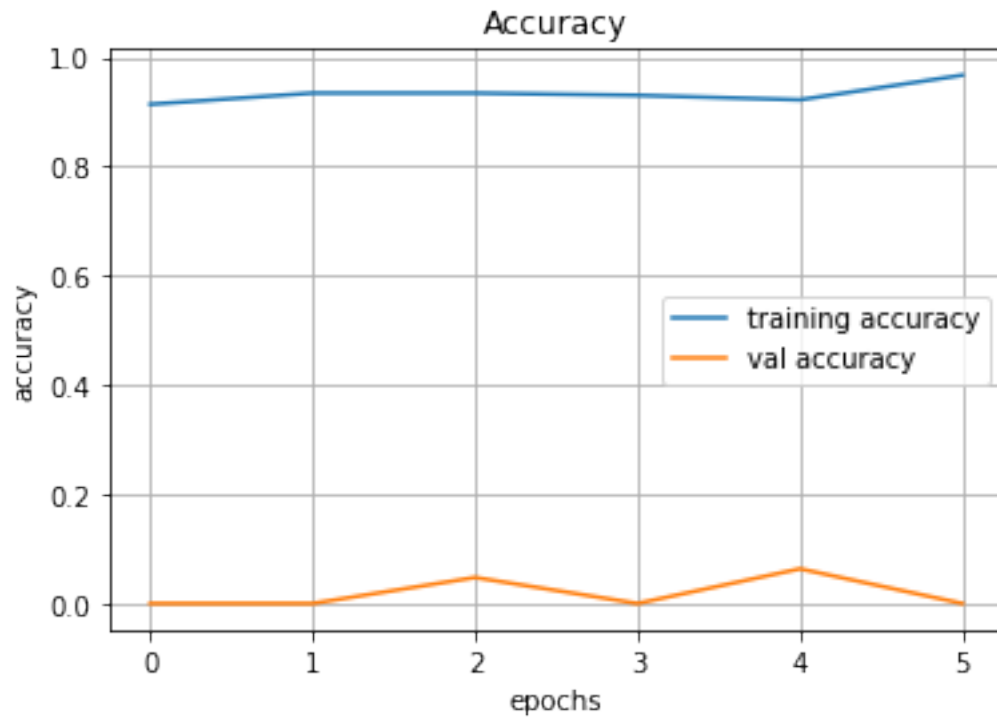
35/35 [=====] - 15s 435ms/step - loss: 0.2333 - accuracy: 0.9221 - val_loss: 9.2002 - val_accuracy: 0.0635

Epoch 6/10

35/35 [=====] - 15s 438ms/step - loss: 0.1443 - accuracy: 0.9672 - val_loss: 8.9770 - val_accuracy: 0.0000e+00

Plot the training and validation accuracy, and the loss

```
[ ]: plt.plot(histor_mobilenet.history['accuracy'] ,label = 'training accuracy')
      plt.plot(histor_mobilenet.history['val_accuracy'], label = 'val accuracy')
      plt.xlabel('epochs')
      plt.ylabel('accuracy')
      plt.grid()
      plt.title('Accuracy')
      plt.legend()
      plt.show()
```

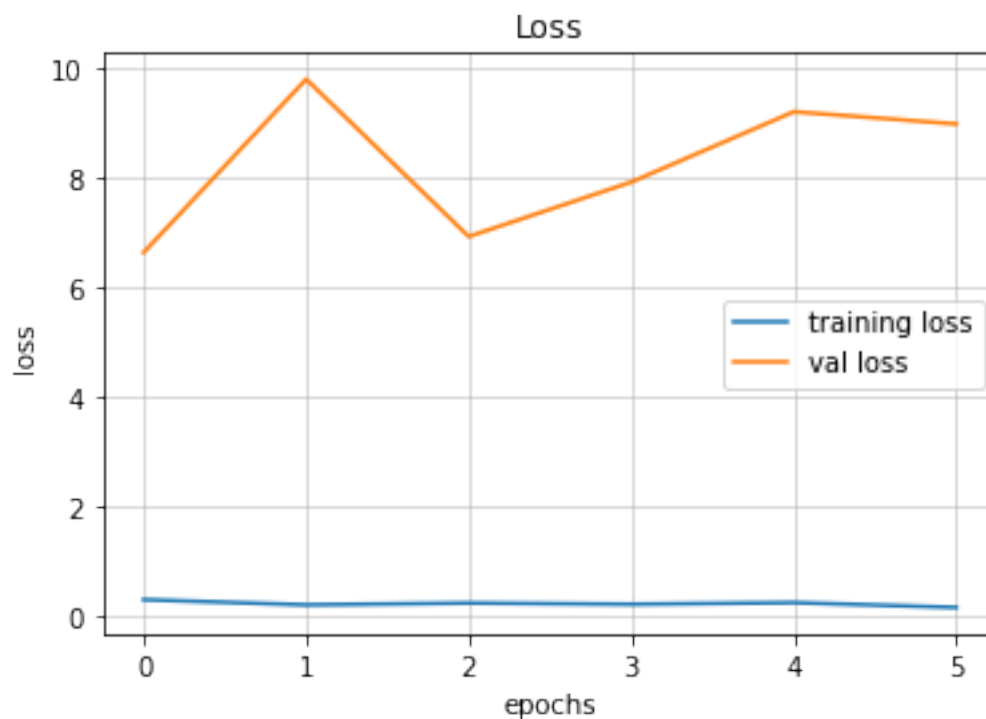


```
[ ]: plt.plot(histor_mobilenet.history['loss'], label = 'training loss')
plt.plot(histor_mobilenet.history['val_loss'], label = 'val loss')

plt.title('Loss')

plt.xlabel('epochs')
plt.ylabel('loss')

plt.grid(alpha = 0.6)
plt.legend()
plt.show()
```



```
[ ]: prediction_mobilenet = mobileNetModel.predict(testGenerator_Mobilenet)

[ ]: prdictionMobilenet = np.argmax(prediction_mobilenet , axis = -1)

[ ]: testGenerator_Mobilenet.classes

[ ]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
          1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
          dtype=int32)

[ ]: print(classification_report(testGenerator_Mobilenet.classes ,
                                ↪ prdictionMobilenet))
```

	precision	recall	f1-score	support
0	0.33	0.31	0.32	26
1	0.42	0.55	0.48	20
2	0.25	0.20	0.22	20
accuracy			0.35	66
macro avg	0.34	0.35	0.34	66
weighted avg	0.34	0.35	0.34	66

```
[ ]:
```

Transfer Learning using Densenet121:->.....

Prepare the dataset for the transfer learning algorithm using Densenet121 with the image size as 224x224x3

Freeze the top layers of the pre-trained model

```
[ ]: DenseNetModel = DenseNet121(include_top = False,
                                weights = 'imagenet', input_shape = (224,224,3))
```

```
[ ]: DenseNetModel.output
```

```
[ ]: <KerasTensor: shape=(None, 7, 7, 1024) dtype=float32 (created by layer 'relu')>
```

Add a dense layer at the end of the pre-trained model followed by a dropout layer and try various combinations to get an accuracy

```
[ ]: x = Flatten()(DenseNetModel.output)
x = Dense(728, activation = 'relu')(x)
x = Dropout(0.1)(x)
x = BatchNormalization()(x)
predictions = Dense(3, activation = 'softmax')(x)

#////////////////////////////////////
->///
```

```
[ ]: model_transfer1 = Model(inputs = DenseNetModel.input, outputs = predictions)

model_transfer1.compile(optimizer = 'Adam',
                        loss = 'categorical_crossentropy', metrics = ['accuracy'])

model_transfer1.summary()
```

Model: "model_4"

```
-----
-----
Layer (type)                Output Shape              Param #   Connected to
-----
=====
input_4 (InputLayer)        [(None, 224, 224, 3) 0   []
                                )]

zero_padding2d_2 (ZeroPadding2D) (None, 230, 230, 3) 0
['input_4[0][0]']
D)

conv1/conv (Conv2D)         (None, 112, 112, 64) 9408
```

```

['zero_padding2d_2[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]']
)

zero_padding2d_3 (ZeroPadding2D) (None, 114, 114, 64 0
['conv1/relu[0][0]']
D)
)

pool1 (MaxPooling2D) (None, 56, 56, 64) 0
['zero_padding2d_3[0][0]']

conv2_block1_0_bn (BatchNormal (None, 56, 56, 64) 256 ['pool1[0][0]']
ization)

conv2_block1_0_relu (Activatio (None, 56, 56, 64) 0
['conv2_block1_0_bn[0][0]']
n)

conv2_block1_1_conv (Conv2D) (None, 56, 56, 128) 8192
['conv2_block1_0_relu[0][0]']

conv2_block1_1_bn (BatchNormal (None, 56, 56, 128) 512
['conv2_block1_1_conv[0][0]']
ization)

conv2_block1_1_relu (Activatio (None, 56, 56, 128) 0
['conv2_block1_1_bn[0][0]']
n)

conv2_block1_2_conv (Conv2D) (None, 56, 56, 32) 36864
['conv2_block1_1_relu[0][0]']

conv2_block1_concat (Concatena (None, 56, 56, 96) 0 ['pool1[0][0]',
te)
['conv2_block1_2_conv[0][0]']

conv2_block2_0_bn (BatchNormal (None, 56, 56, 96) 384
['conv2_block1_concat[0][0]']
ization)

conv2_block2_0_relu (Activatio (None, 56, 56, 96) 0

```



```

['conv2_block2_0_bn[0][0]']
n)

conv2_block2_1_conv (Conv2D) (None, 56, 56, 128) 12288
['conv2_block2_0_relu[0][0]']

conv2_block2_1_bn (BatchNormal (None, 56, 56, 128) 512
['conv2_block2_1_conv[0][0]']
ization)

conv2_block2_1_relu (Activatio (None, 56, 56, 128) 0
['conv2_block2_1_bn[0][0]']
n)

conv2_block2_2_conv (Conv2D) (None, 56, 56, 32) 36864
['conv2_block2_1_relu[0][0]']

conv2_block2_concat (Concatena (None, 56, 56, 128) 0
['conv2_block1_concat[0][0]',
te)
'conv2_block2_2_conv[0][0]']

conv2_block3_0_bn (BatchNormal (None, 56, 56, 128) 512
['conv2_block2_concat[0][0]']
ization)

conv2_block3_0_relu (Activatio (None, 56, 56, 128) 0
['conv2_block3_0_bn[0][0]']
n)

conv2_block3_1_conv (Conv2D) (None, 56, 56, 128) 16384
['conv2_block3_0_relu[0][0]']

conv2_block3_1_bn (BatchNormal (None, 56, 56, 128) 512
['conv2_block3_1_conv[0][0]']
ization)

conv2_block3_1_relu (Activatio (None, 56, 56, 128) 0
['conv2_block3_1_bn[0][0]']
n)

conv2_block3_2_conv (Conv2D) (None, 56, 56, 32) 36864
['conv2_block3_1_relu[0][0]']

conv2_block3_concat (Concatena (None, 56, 56, 160) 0
['conv2_block2_concat[0][0]',
te)
'conv2_block3_2_conv[0][0]']

```

```

conv2_block4_0_bn (BatchNormal (None, 56, 56, 160) 640
['conv2_block3_concat[0][0]']
ization)

conv2_block4_0_relu (Activatio (None, 56, 56, 160) 0
['conv2_block4_0_bn[0][0]']
n)

conv2_block4_1_conv (Conv2D) (None, 56, 56, 128) 20480
['conv2_block4_0_relu[0][0]']

conv2_block4_1_bn (BatchNormal (None, 56, 56, 128) 512
['conv2_block4_1_conv[0][0]']
ization)

conv2_block4_1_relu (Activatio (None, 56, 56, 128) 0
['conv2_block4_1_bn[0][0]']
n)

conv2_block4_2_conv (Conv2D) (None, 56, 56, 32) 36864
['conv2_block4_1_relu[0][0]']

conv2_block4_concat (Concatena (None, 56, 56, 192) 0
['conv2_block3_concat[0][0]',
te)
'conv2_block4_2_conv[0][0]']

conv2_block5_0_bn (BatchNormal (None, 56, 56, 192) 768
['conv2_block4_concat[0][0]']
ization)

conv2_block5_0_relu (Activatio (None, 56, 56, 192) 0
['conv2_block5_0_bn[0][0]']
n)

conv2_block5_1_conv (Conv2D) (None, 56, 56, 128) 24576
['conv2_block5_0_relu[0][0]']

conv2_block5_1_bn (BatchNormal (None, 56, 56, 128) 512
['conv2_block5_1_conv[0][0]']
ization)

conv2_block5_1_relu (Activatio (None, 56, 56, 128) 0
['conv2_block5_1_bn[0][0]']
n)

conv2_block5_2_conv (Conv2D) (None, 56, 56, 32) 36864

```

```

['conv2_block5_1_relu[0][0]']

conv2_block5_concat (Concatenation) (None, 56, 56, 224) 0
['conv2_block4_concat[0][0]',
 te)
'conv2_block5_2_conv[0][0]']

conv2_block6_0_bn (BatchNormalization) (None, 56, 56, 224) 896
['conv2_block5_concat[0][0]']
ization)

conv2_block6_0_relu (Activation) (None, 56, 56, 224) 0
['conv2_block6_0_bn[0][0]']
n)

conv2_block6_1_conv (Conv2D) (None, 56, 56, 128) 28672
['conv2_block6_0_relu[0][0]']

conv2_block6_1_bn (BatchNormalization) (None, 56, 56, 128) 512
['conv2_block6_1_conv[0][0]']
ization)

conv2_block6_1_relu (Activation) (None, 56, 56, 128) 0
['conv2_block6_1_bn[0][0]']
n)

conv2_block6_2_conv (Conv2D) (None, 56, 56, 32) 36864
['conv2_block6_1_relu[0][0]']

conv2_block6_concat (Concatenation) (None, 56, 56, 256) 0
['conv2_block5_concat[0][0]',
 te)
'conv2_block6_2_conv[0][0]']

pool2_bn (BatchNormalization) (None, 56, 56, 256) 1024
['conv2_block6_concat[0][0]']

pool2_relu (Activation) (None, 56, 56, 256) 0
['pool2_bn[0][0]']

pool2_conv (Conv2D) (None, 56, 56, 128) 32768
['pool2_relu[0][0]']

pool2_pool (AveragePooling2D) (None, 28, 28, 128) 0
['pool2_conv[0][0]']

conv3_block1_0_bn (BatchNormalization) (None, 28, 28, 128) 512
['pool2_pool[0][0]']

```

```

ization)

conv3_block1_0_relu (Activation) (None, 28, 28, 128) 0
['conv3_block1_0_bn[0][0]']
n)

conv3_block1_1_conv (Conv2D) (None, 28, 28, 128) 16384
['conv3_block1_0_relu[0][0]']

conv3_block1_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block1_1_conv[0][0]']
ization)

conv3_block1_1_relu (Activation) (None, 28, 28, 128) 0
['conv3_block1_1_bn[0][0]']
n)

conv3_block1_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block1_1_relu[0][0]']

conv3_block1_concat (Concatenation) (None, 28, 28, 160) 0
['pool2_pool[0][0]',
te)
'conv3_block1_2_conv[0][0]']

conv3_block2_0_bn (BatchNormal (None, 28, 28, 160) 640
['conv3_block1_concat[0][0]']
ization)

conv3_block2_0_relu (Activation) (None, 28, 28, 160) 0
['conv3_block2_0_bn[0][0]']
n)

conv3_block2_1_conv (Conv2D) (None, 28, 28, 128) 20480
['conv3_block2_0_relu[0][0]']

conv3_block2_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block2_1_conv[0][0]']
ization)

conv3_block2_1_relu (Activation) (None, 28, 28, 128) 0
['conv3_block2_1_bn[0][0]']
n)

conv3_block2_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block2_1_relu[0][0]']

conv3_block2_concat (Concatenation) (None, 28, 28, 192) 0

```

```

['conv3_block1_concat[0][0]',
 te)
'conv3_block2_2_conv[0][0]']

conv3_block3_0_bn (BatchNormal (None, 28, 28, 192) 768
['conv3_block2_concat[0][0]']
ization)

conv3_block3_0_relu (Activatio (None, 28, 28, 192) 0
['conv3_block3_0_bn[0][0]']
n)

conv3_block3_1_conv (Conv2D) (None, 28, 28, 128) 24576
['conv3_block3_0_relu[0][0]']

conv3_block3_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block3_1_conv[0][0]']
ization)

conv3_block3_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block3_1_bn[0][0]']
n)

conv3_block3_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block3_1_relu[0][0]']

conv3_block3_concat (Concatena (None, 28, 28, 224) 0
['conv3_block2_concat[0][0]',
 te)
'conv3_block3_2_conv[0][0]']

conv3_block4_0_bn (BatchNormal (None, 28, 28, 224) 896
['conv3_block3_concat[0][0]']
ization)

conv3_block4_0_relu (Activatio (None, 28, 28, 224) 0
['conv3_block4_0_bn[0][0]']
n)

conv3_block4_1_conv (Conv2D) (None, 28, 28, 128) 28672
['conv3_block4_0_relu[0][0]']

conv3_block4_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block4_1_conv[0][0]']
ization)

conv3_block4_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block4_1_bn[0][0]']

```

```

n)

conv3_block4_2_conv (Conv2D)    (None, 28, 28, 32)    36864
['conv3_block4_1_relu[0][0]']

conv3_block4_concat (Concatena (None, 28, 28, 256)    0
['conv3_block3_concat[0][0]',
te)
'conv3_block4_2_conv[0][0]']

conv3_block5_0_bn (BatchNormal (None, 28, 28, 256)    1024
['conv3_block4_concat[0][0]']
ization)

conv3_block5_0_relu (Activatio (None, 28, 28, 256)    0
['conv3_block5_0_bn[0][0]']
n)

conv3_block5_1_conv (Conv2D)    (None, 28, 28, 128)    32768
['conv3_block5_0_relu[0][0]']

conv3_block5_1_bn (BatchNormal (None, 28, 28, 128)    512
['conv3_block5_1_conv[0][0]']
ization)

conv3_block5_1_relu (Activatio (None, 28, 28, 128)    0
['conv3_block5_1_bn[0][0]']
n)

conv3_block5_2_conv (Conv2D)    (None, 28, 28, 32)    36864
['conv3_block5_1_relu[0][0]']

conv3_block5_concat (Concatena (None, 28, 28, 288)    0
['conv3_block4_concat[0][0]',
te)
'conv3_block5_2_conv[0][0]']

conv3_block6_0_bn (BatchNormal (None, 28, 28, 288)    1152
['conv3_block5_concat[0][0]']
ization)

conv3_block6_0_relu (Activatio (None, 28, 28, 288)    0
['conv3_block6_0_bn[0][0]']
n)

conv3_block6_1_conv (Conv2D)    (None, 28, 28, 128)    36864
['conv3_block6_0_relu[0][0]']

```

```

conv3_block6_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block6_1_conv[0][0]']
ization)

conv3_block6_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block6_1_bn[0][0]']
n)

conv3_block6_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block6_1_relu[0][0]']

conv3_block6_concat (Concatena (None, 28, 28, 320) 0
['conv3_block5_concat[0][0]',
te)
'conv3_block6_2_conv[0][0]']

conv3_block7_0_bn (BatchNormal (None, 28, 28, 320) 1280
['conv3_block6_concat[0][0]']
ization)

conv3_block7_0_relu (Activatio (None, 28, 28, 320) 0
['conv3_block7_0_bn[0][0]']
n)

conv3_block7_1_conv (Conv2D) (None, 28, 28, 128) 40960
['conv3_block7_0_relu[0][0]']

conv3_block7_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block7_1_conv[0][0]']
ization)

conv3_block7_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block7_1_bn[0][0]']
n)

conv3_block7_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block7_1_relu[0][0]']

conv3_block7_concat (Concatena (None, 28, 28, 352) 0
['conv3_block6_concat[0][0]',
te)
'conv3_block7_2_conv[0][0]']

conv3_block8_0_bn (BatchNormal (None, 28, 28, 352) 1408
['conv3_block7_concat[0][0]']
ization)

conv3_block8_0_relu (Activatio (None, 28, 28, 352) 0

```

```

['conv3_block8_0_bn[0][0]']
n)

conv3_block8_1_conv (Conv2D) (None, 28, 28, 128) 45056
['conv3_block8_0_relu[0][0]']

conv3_block8_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block8_1_conv[0][0]']
ization)

conv3_block8_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block8_1_bn[0][0]']
n)

conv3_block8_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block8_1_relu[0][0]']

conv3_block8_concat (Concatena (None, 28, 28, 384) 0
['conv3_block7_concat[0][0]',
te)
'conv3_block8_2_conv[0][0]']

conv3_block9_0_bn (BatchNormal (None, 28, 28, 384) 1536
['conv3_block8_concat[0][0]']
ization)

conv3_block9_0_relu (Activatio (None, 28, 28, 384) 0
['conv3_block9_0_bn[0][0]']
n)

conv3_block9_1_conv (Conv2D) (None, 28, 28, 128) 49152
['conv3_block9_0_relu[0][0]']

conv3_block9_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block9_1_conv[0][0]']
ization)

conv3_block9_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block9_1_bn[0][0]']
n)

conv3_block9_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block9_1_relu[0][0]']

conv3_block9_concat (Concatena (None, 28, 28, 416) 0
['conv3_block8_concat[0][0]',
te)
'conv3_block9_2_conv[0][0]']

```



```

conv3_block10_0_bn (BatchNormaliza (None, 28, 28, 416) 1664
['conv3_block9_concat[0][0]']
lization)

conv3_block10_0_relu (Activation (None, 28, 28, 416) 0
['conv3_block10_0_bn[0][0]']
on)

conv3_block10_1_conv (Conv2D) (None, 28, 28, 128) 53248
['conv3_block10_0_relu[0][0]']

conv3_block10_1_bn (BatchNormaliza (None, 28, 28, 128) 512
['conv3_block10_1_conv[0][0]']
lization)

conv3_block10_1_relu (Activation (None, 28, 28, 128) 0
['conv3_block10_1_bn[0][0]']
on)

conv3_block10_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block10_1_relu[0][0]']

conv3_block10_concat (Concatenate) (None, 28, 28, 448) 0
['conv3_block9_concat[0][0]',
ate)
['conv3_block10_2_conv[0][0]']

conv3_block11_0_bn (BatchNormaliza (None, 28, 28, 448) 1792
['conv3_block10_concat[0][0]']
lization)

conv3_block11_0_relu (Activation (None, 28, 28, 448) 0
['conv3_block11_0_bn[0][0]']
on)

conv3_block11_1_conv (Conv2D) (None, 28, 28, 128) 57344
['conv3_block11_0_relu[0][0]']

conv3_block11_1_bn (BatchNormaliza (None, 28, 28, 128) 512
['conv3_block11_1_conv[0][0]']
lization)

conv3_block11_1_relu (Activation (None, 28, 28, 128) 0
['conv3_block11_1_bn[0][0]']
on)

conv3_block11_2_conv (Conv2D) (None, 28, 28, 32) 36864

```

```

['conv3_block11_1_relu[0][0]']

conv3_block11_concat (Concaten (None, 28, 28, 480) 0
['conv3_block10_concat[0][0]',
ate)
'conv3_block11_2_conv[0][0]']

conv3_block12_0_bn (BatchNorma (None, 28, 28, 480) 1920
['conv3_block11_concat[0][0]']
lization)

conv3_block12_0_relu (Activati (None, 28, 28, 480) 0
['conv3_block12_0_bn[0][0]']
on)

conv3_block12_1_conv (Conv2D) (None, 28, 28, 128) 61440
['conv3_block12_0_relu[0][0]']

conv3_block12_1_bn (BatchNorma (None, 28, 28, 128) 512
['conv3_block12_1_conv[0][0]']
lization)

conv3_block12_1_relu (Activati (None, 28, 28, 128) 0
['conv3_block12_1_bn[0][0]']
on)

conv3_block12_2_conv (Conv2D) (None, 28, 28, 32) 36864
['conv3_block12_1_relu[0][0]']

conv3_block12_concat (Concaten (None, 28, 28, 512) 0
['conv3_block11_concat[0][0]',
ate)
'conv3_block12_2_conv[0][0]']

pool3_bn (BatchNormalization) (None, 28, 28, 512) 2048
['conv3_block12_concat[0][0]']

pool3_relu (Activation) (None, 28, 28, 512) 0
['pool3_bn[0][0]']

pool3_conv (Conv2D) (None, 28, 28, 256) 131072
['pool3_relu[0][0]']

pool3_pool (AveragePooling2D) (None, 14, 14, 256) 0
['pool3_conv[0][0]']

conv4_block1_0_bn (BatchNormal (None, 14, 14, 256) 1024
['pool3_pool[0][0]']

```

```

ization)

conv4_block1_0_relu (Activation) (None, 14, 14, 256) 0
['conv4_block1_0_bn[0][0]']
n)

conv4_block1_1_conv (Conv2D) (None, 14, 14, 128) 32768
['conv4_block1_0_relu[0][0]']

conv4_block1_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block1_1_conv[0][0]']
ization)

conv4_block1_1_relu (Activation) (None, 14, 14, 128) 0
['conv4_block1_1_bn[0][0]']
n)

conv4_block1_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block1_1_relu[0][0]']

conv4_block1_concat (Concatenation) (None, 14, 14, 288) 0
['pool3_pool[0][0]',
te)
'conv4_block1_2_conv[0][0]']

conv4_block2_0_bn (BatchNormal (None, 14, 14, 288) 1152
['conv4_block1_concat[0][0]']
ization)

conv4_block2_0_relu (Activation) (None, 14, 14, 288) 0
['conv4_block2_0_bn[0][0]']
n)

conv4_block2_1_conv (Conv2D) (None, 14, 14, 128) 36864
['conv4_block2_0_relu[0][0]']

conv4_block2_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block2_1_conv[0][0]']
ization)

conv4_block2_1_relu (Activation) (None, 14, 14, 128) 0
['conv4_block2_1_bn[0][0]']
n)

conv4_block2_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block2_1_relu[0][0]']

conv4_block2_concat (Concatenation) (None, 14, 14, 320) 0

```

```

['conv4_block1_concat[0][0]',
 te)
'conv4_block2_2_conv[0][0]']

conv4_block3_0_bn (BatchNormal (None, 14, 14, 320) 1280
['conv4_block2_concat[0][0]']
ization)

conv4_block3_0_relu (Activatio (None, 14, 14, 320) 0
['conv4_block3_0_bn[0][0]']
n)

conv4_block3_1_conv (Conv2D) (None, 14, 14, 128) 40960
['conv4_block3_0_relu[0][0]']

conv4_block3_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block3_1_conv[0][0]']
ization)

conv4_block3_1_relu (Activatio (None, 14, 14, 128) 0
['conv4_block3_1_bn[0][0]']
n)

conv4_block3_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block3_1_relu[0][0]']

conv4_block3_concat (Concatena (None, 14, 14, 352) 0
['conv4_block2_concat[0][0]',
 te)
'conv4_block3_2_conv[0][0]']

conv4_block4_0_bn (BatchNormal (None, 14, 14, 352) 1408
['conv4_block3_concat[0][0]']
ization)

conv4_block4_0_relu (Activatio (None, 14, 14, 352) 0
['conv4_block4_0_bn[0][0]']
n)

conv4_block4_1_conv (Conv2D) (None, 14, 14, 128) 45056
['conv4_block4_0_relu[0][0]']

conv4_block4_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block4_1_conv[0][0]']
ization)

conv4_block4_1_relu (Activatio (None, 14, 14, 128) 0
['conv4_block4_1_bn[0][0]']

```

```

n)

conv4_block4_2_conv (Conv2D)    (None, 14, 14, 32)    36864
['conv4_block4_1_relu[0][0]']

conv4_block4_concat (Concatena (None, 14, 14, 384)  0
['conv4_block3_concat[0][0]',
te)
'conv4_block4_2_conv[0][0]']

conv4_block5_0_bn (BatchNormal (None, 14, 14, 384)  1536
['conv4_block4_concat[0][0]']
ization)

conv4_block5_0_relu (Activatio (None, 14, 14, 384)  0
['conv4_block5_0_bn[0][0]']
n)

conv4_block5_1_conv (Conv2D)    (None, 14, 14, 128)  49152
['conv4_block5_0_relu[0][0]']

conv4_block5_1_bn (BatchNormal (None, 14, 14, 128)  512
['conv4_block5_1_conv[0][0]']
ization)

conv4_block5_1_relu (Activatio (None, 14, 14, 128)  0
['conv4_block5_1_bn[0][0]']
n)

conv4_block5_2_conv (Conv2D)    (None, 14, 14, 32)    36864
['conv4_block5_1_relu[0][0]']

conv4_block5_concat (Concatena (None, 14, 14, 416)  0
['conv4_block4_concat[0][0]',
te)
'conv4_block5_2_conv[0][0]']

conv4_block6_0_bn (BatchNormal (None, 14, 14, 416)  1664
['conv4_block5_concat[0][0]']
ization)

conv4_block6_0_relu (Activatio (None, 14, 14, 416)  0
['conv4_block6_0_bn[0][0]']
n)

conv4_block6_1_conv (Conv2D)    (None, 14, 14, 128)  53248
['conv4_block6_0_relu[0][0]']

```

```

conv4_block6_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block6_1_conv[0][0]']
ization)

conv4_block6_1_relu (Activatio (None, 14, 14, 128) 0
['conv4_block6_1_bn[0][0]']
n)

conv4_block6_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block6_1_relu[0][0]']

conv4_block6_concat (Concatena (None, 14, 14, 448) 0
['conv4_block5_concat[0][0]',
te)
'conv4_block6_2_conv[0][0]']

conv4_block7_0_bn (BatchNormal (None, 14, 14, 448) 1792
['conv4_block6_concat[0][0]']
ization)

conv4_block7_0_relu (Activatio (None, 14, 14, 448) 0
['conv4_block7_0_bn[0][0]']
n)

conv4_block7_1_conv (Conv2D) (None, 14, 14, 128) 57344
['conv4_block7_0_relu[0][0]']

conv4_block7_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block7_1_conv[0][0]']
ization)

conv4_block7_1_relu (Activatio (None, 14, 14, 128) 0
['conv4_block7_1_bn[0][0]']
n)

conv4_block7_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block7_1_relu[0][0]']

conv4_block7_concat (Concatena (None, 14, 14, 480) 0
['conv4_block6_concat[0][0]',
te)
'conv4_block7_2_conv[0][0]']

conv4_block8_0_bn (BatchNormal (None, 14, 14, 480) 1920
['conv4_block7_concat[0][0]']
ization)

conv4_block8_0_relu (Activatio (None, 14, 14, 480) 0

```

```

['conv4_block8_0_bn[0][0]']
n)

conv4_block8_1_conv (Conv2D) (None, 14, 14, 128) 61440
['conv4_block8_0_relu[0][0]']

conv4_block8_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block8_1_conv[0][0]']
ization)

conv4_block8_1_relu (Activatio (None, 14, 14, 128) 0
['conv4_block8_1_bn[0][0]']
n)

conv4_block8_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block8_1_relu[0][0]']

conv4_block8_concat (Concatena (None, 14, 14, 512) 0
['conv4_block7_concat[0][0]',
te)
'conv4_block8_2_conv[0][0]']

conv4_block9_0_bn (BatchNormal (None, 14, 14, 512) 2048
['conv4_block8_concat[0][0]']
ization)

conv4_block9_0_relu (Activatio (None, 14, 14, 512) 0
['conv4_block9_0_bn[0][0]']
n)

conv4_block9_1_conv (Conv2D) (None, 14, 14, 128) 65536
['conv4_block9_0_relu[0][0]']

conv4_block9_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block9_1_conv[0][0]']
ization)

conv4_block9_1_relu (Activatio (None, 14, 14, 128) 0
['conv4_block9_1_bn[0][0]']
n)

conv4_block9_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block9_1_relu[0][0]']

conv4_block9_concat (Concatena (None, 14, 14, 544) 0
['conv4_block8_concat[0][0]',
te)
'conv4_block9_2_conv[0][0]']

```

```

conv4_block10_0_bn (BatchNorma (None, 14, 14, 544) 2176
['conv4_block9_concat[0][0]']
lization)

conv4_block10_0_relu (Activati (None, 14, 14, 544) 0
['conv4_block10_0_bn[0][0]']
on)

conv4_block10_1_conv (Conv2D) (None, 14, 14, 128) 69632
['conv4_block10_0_relu[0][0]']

conv4_block10_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block10_1_conv[0][0]']
lization)

conv4_block10_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block10_1_bn[0][0]']
on)

conv4_block10_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block10_1_relu[0][0]']

conv4_block10_concat (Concaten (None, 14, 14, 576) 0
['conv4_block9_concat[0][0]',
ate)
'conv4_block10_2_conv[0][0]']

conv4_block11_0_bn (BatchNorma (None, 14, 14, 576) 2304
['conv4_block10_concat[0][0]']
lization)

conv4_block11_0_relu (Activati (None, 14, 14, 576) 0
['conv4_block11_0_bn[0][0]']
on)

conv4_block11_1_conv (Conv2D) (None, 14, 14, 128) 73728
['conv4_block11_0_relu[0][0]']

conv4_block11_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block11_1_conv[0][0]']
lization)

conv4_block11_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block11_1_bn[0][0]']
on)

conv4_block11_2_conv (Conv2D) (None, 14, 14, 32) 36864

```



```

['conv4_block11_1_relu[0][0]']

conv4_block11_concat (Concaten (None, 14, 14, 608) 0
['conv4_block10_concat[0][0]',
ate)
'conv4_block11_2_conv[0][0]']

conv4_block12_0_bn (BatchNorma (None, 14, 14, 608) 2432
['conv4_block11_concat[0][0]']
lization)

conv4_block12_0_relu (Activati (None, 14, 14, 608) 0
['conv4_block12_0_bn[0][0]']
on)

conv4_block12_1_conv (Conv2D) (None, 14, 14, 128) 77824
['conv4_block12_0_relu[0][0]']

conv4_block12_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block12_1_conv[0][0]']
lization)

conv4_block12_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block12_1_bn[0][0]']
on)

conv4_block12_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block12_1_relu[0][0]']

conv4_block12_concat (Concaten (None, 14, 14, 640) 0
['conv4_block11_concat[0][0]',
ate)
'conv4_block12_2_conv[0][0]']

conv4_block13_0_bn (BatchNorma (None, 14, 14, 640) 2560
['conv4_block12_concat[0][0]']
lization)

conv4_block13_0_relu (Activati (None, 14, 14, 640) 0
['conv4_block13_0_bn[0][0]']
on)

conv4_block13_1_conv (Conv2D) (None, 14, 14, 128) 81920
['conv4_block13_0_relu[0][0]']

conv4_block13_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block13_1_conv[0][0]']
lization)

```

```

conv4_block13_1_relu (Activation) (None, 14, 14, 128) 0
['conv4_block13_1_bn[0][0]']
on)

conv4_block13_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block13_1_relu[0][0]']

conv4_block13_concat (Concatenation) (None, 14, 14, 672) 0
['conv4_block12_concat[0][0]',
ate)
'conv4_block13_2_conv[0][0]']

conv4_block14_0_bn (BatchNormalization) (None, 14, 14, 672) 2688
['conv4_block13_concat[0][0]']
lization)

conv4_block14_0_relu (Activation) (None, 14, 14, 672) 0
['conv4_block14_0_bn[0][0]']
on)

conv4_block14_1_conv (Conv2D) (None, 14, 14, 128) 86016
['conv4_block14_0_relu[0][0]']

conv4_block14_1_bn (BatchNormalization) (None, 14, 14, 128) 512
['conv4_block14_1_conv[0][0]']
lization)

conv4_block14_1_relu (Activation) (None, 14, 14, 128) 0
['conv4_block14_1_bn[0][0]']
on)

conv4_block14_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block14_1_relu[0][0]']

conv4_block14_concat (Concatenation) (None, 14, 14, 704) 0
['conv4_block13_concat[0][0]',
ate)
'conv4_block14_2_conv[0][0]']

conv4_block15_0_bn (BatchNormalization) (None, 14, 14, 704) 2816
['conv4_block14_concat[0][0]']
lization)

conv4_block15_0_relu (Activation) (None, 14, 14, 704) 0
['conv4_block15_0_bn[0][0]']
on)

```

```

conv4_block15_1_conv (Conv2D) (None, 14, 14, 128) 90112
['conv4_block15_0_relu[0][0]']

conv4_block15_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block15_1_conv[0][0]']
lization)

conv4_block15_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block15_1_bn[0][0]']
on)

conv4_block15_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block15_1_relu[0][0]']

conv4_block15_concat (Concaten (None, 14, 14, 736) 0
['conv4_block14_concat[0][0]',
ate)
'conv4_block15_2_conv[0][0]']

conv4_block16_0_bn (BatchNorma (None, 14, 14, 736) 2944
['conv4_block15_concat[0][0]']
lization)

conv4_block16_0_relu (Activati (None, 14, 14, 736) 0
['conv4_block16_0_bn[0][0]']
on)

conv4_block16_1_conv (Conv2D) (None, 14, 14, 128) 94208
['conv4_block16_0_relu[0][0]']

conv4_block16_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block16_1_conv[0][0]']
lization)

conv4_block16_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block16_1_bn[0][0]']
on)

conv4_block16_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block16_1_relu[0][0]']

conv4_block16_concat (Concaten (None, 14, 14, 768) 0
['conv4_block15_concat[0][0]',
ate)
'conv4_block16_2_conv[0][0]']

conv4_block17_0_bn (BatchNorma (None, 14, 14, 768) 3072
['conv4_block16_concat[0][0]']

```

```

lization)

conv4_block17_0_relu (Activation) (None, 14, 14, 768) 0
['conv4_block17_0_bn[0][0]']
on)

conv4_block17_1_conv (Conv2D) (None, 14, 14, 128) 98304
['conv4_block17_0_relu[0][0]']

conv4_block17_1_bn (BatchNormalization) (None, 14, 14, 128) 512
['conv4_block17_1_conv[0][0]']
lization)

conv4_block17_1_relu (Activation) (None, 14, 14, 128) 0
['conv4_block17_1_bn[0][0]']
on)

conv4_block17_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block17_1_relu[0][0]']

conv4_block17_concat (Concatenation) (None, 14, 14, 800) 0
['conv4_block16_concat[0][0]',
ate)
'conv4_block17_2_conv[0][0]']

conv4_block18_0_bn (BatchNormalization) (None, 14, 14, 800) 3200
['conv4_block17_concat[0][0]']
lization)

conv4_block18_0_relu (Activation) (None, 14, 14, 800) 0
['conv4_block18_0_bn[0][0]']
on)

conv4_block18_1_conv (Conv2D) (None, 14, 14, 128) 102400
['conv4_block18_0_relu[0][0]']

conv4_block18_1_bn (BatchNormalization) (None, 14, 14, 128) 512
['conv4_block18_1_conv[0][0]']
lization)

conv4_block18_1_relu (Activation) (None, 14, 14, 128) 0
['conv4_block18_1_bn[0][0]']
on)

conv4_block18_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block18_1_relu[0][0]']

conv4_block18_concat (Concatenation) (None, 14, 14, 832) 0

```

```

['conv4_block17_concat[0][0]',
 ate)
'conv4_block18_2_conv[0][0]']

conv4_block19_0_bn (BatchNorma (None, 14, 14, 832) 3328
['conv4_block18_concat[0][0]']
lization)

conv4_block19_0_relu (Activati (None, 14, 14, 832) 0
['conv4_block19_0_bn[0][0]']
on)

conv4_block19_1_conv (Conv2D) (None, 14, 14, 128) 106496
['conv4_block19_0_relu[0][0]']

conv4_block19_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block19_1_conv[0][0]']
lization)

conv4_block19_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block19_1_bn[0][0]']
on)

conv4_block19_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block19_1_relu[0][0]']

conv4_block19_concat (Concaten (None, 14, 14, 864) 0
['conv4_block18_concat[0][0]',
 ate)
'conv4_block19_2_conv[0][0]']

conv4_block20_0_bn (BatchNorma (None, 14, 14, 864) 3456
['conv4_block19_concat[0][0]']
lization)

conv4_block20_0_relu (Activati (None, 14, 14, 864) 0
['conv4_block20_0_bn[0][0]']
on)

conv4_block20_1_conv (Conv2D) (None, 14, 14, 128) 110592
['conv4_block20_0_relu[0][0]']

conv4_block20_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block20_1_conv[0][0]']
lization)

conv4_block20_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block20_1_bn[0][0]']

```

```

on)

conv4_block20_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block20_1_relu[0][0]']

conv4_block20_concat (Concaten (None, 14, 14, 896) 0
['conv4_block19_concat[0][0]',
ate)
'conv4_block20_2_conv[0][0]']

conv4_block21_0_bn (BatchNorma (None, 14, 14, 896) 3584
['conv4_block20_concat[0][0]']
lization)

conv4_block21_0_relu (Activati (None, 14, 14, 896) 0
['conv4_block21_0_bn[0][0]']
on)

conv4_block21_1_conv (Conv2D) (None, 14, 14, 128) 114688
['conv4_block21_0_relu[0][0]']

conv4_block21_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block21_1_conv[0][0]']
lization)

conv4_block21_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block21_1_bn[0][0]']
on)

conv4_block21_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block21_1_relu[0][0]']

conv4_block21_concat (Concaten (None, 14, 14, 928) 0
['conv4_block20_concat[0][0]',
ate)
'conv4_block21_2_conv[0][0]']

conv4_block22_0_bn (BatchNorma (None, 14, 14, 928) 3712
['conv4_block21_concat[0][0]']
lization)

conv4_block22_0_relu (Activati (None, 14, 14, 928) 0
['conv4_block22_0_bn[0][0]']
on)

conv4_block22_1_conv (Conv2D) (None, 14, 14, 128) 118784
['conv4_block22_0_relu[0][0]']

```

```

conv4_block22_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block22_1_conv[0][0]']
lization)

conv4_block22_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block22_1_bn[0][0]']
on)

conv4_block22_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block22_1_relu[0][0]']

conv4_block22_concat (Concaten (None, 14, 14, 960) 0
['conv4_block21_concat[0][0]',
ate)
'conv4_block22_2_conv[0][0]']

conv4_block23_0_bn (BatchNorma (None, 14, 14, 960) 3840
['conv4_block22_concat[0][0]']
lization)

conv4_block23_0_relu (Activati (None, 14, 14, 960) 0
['conv4_block23_0_bn[0][0]']
on)

conv4_block23_1_conv (Conv2D) (None, 14, 14, 128) 122880
['conv4_block23_0_relu[0][0]']

conv4_block23_1_bn (BatchNorma (None, 14, 14, 128) 512
['conv4_block23_1_conv[0][0]']
lization)

conv4_block23_1_relu (Activati (None, 14, 14, 128) 0
['conv4_block23_1_bn[0][0]']
on)

conv4_block23_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block23_1_relu[0][0]']

conv4_block23_concat (Concaten (None, 14, 14, 992) 0
['conv4_block22_concat[0][0]',
ate)
'conv4_block23_2_conv[0][0]']

conv4_block24_0_bn (BatchNorma (None, 14, 14, 992) 3968
['conv4_block23_concat[0][0]']
lization)

conv4_block24_0_relu (Activati (None, 14, 14, 992) 0

```

```

['conv4_block24_0_bn[0][0]']
on)

conv4_block24_1_conv (Conv2D) (None, 14, 14, 128) 126976
['conv4_block24_0_relu[0][0]']

conv4_block24_1_bn (BatchNormal (None, 14, 14, 128) 512
['conv4_block24_1_conv[0][0]']
lization)

conv4_block24_1_relu (Activation (None, 14, 14, 128) 0
['conv4_block24_1_bn[0][0]']
on)

conv4_block24_2_conv (Conv2D) (None, 14, 14, 32) 36864
['conv4_block24_1_relu[0][0]']

conv4_block24_concat (Concaten (None, 14, 14, 1024 0
['conv4_block23_concat[0][0]',
ate)
)
['conv4_block24_2_conv[0][0]']

pool4_bn (BatchNormalization) (None, 14, 14, 1024 4096
['conv4_block24_concat[0][0]']
)

pool4_relu (Activation) (None, 14, 14, 1024 0
['pool4_bn[0][0]']
)

pool4_conv (Conv2D) (None, 14, 14, 512) 524288
['pool4_relu[0][0]']

pool4_pool (AveragePooling2D) (None, 7, 7, 512) 0
['pool4_conv[0][0]']

conv5_block1_0_bn (BatchNormal (None, 7, 7, 512) 2048
['pool4_pool[0][0]']
ization)

conv5_block1_0_relu (Activation (None, 7, 7, 512) 0
['conv5_block1_0_bn[0][0]']
n)

conv5_block1_1_conv (Conv2D) (None, 7, 7, 128) 65536
['conv5_block1_0_relu[0][0]']

conv5_block1_1_bn (BatchNormal (None, 7, 7, 128) 512

```



```

['conv5_block1_1_conv[0][0]']
ization)

conv5_block1_1_relu (Activation) (None, 7, 7, 128) 0
['conv5_block1_1_bn[0][0]']
n)

conv5_block1_2_conv (Conv2D) (None, 7, 7, 32) 36864
['conv5_block1_1_relu[0][0]']

conv5_block1_concat (Concatenation) (None, 7, 7, 544) 0
['pool4_pool[0][0]',
te)
'conv5_block1_2_conv[0][0]']

conv5_block2_0_bn (BatchNormalization) (None, 7, 7, 544) 2176
['conv5_block1_concat[0][0]']
ization)

conv5_block2_0_relu (Activation) (None, 7, 7, 544) 0
['conv5_block2_0_bn[0][0]']
n)

conv5_block2_1_conv (Conv2D) (None, 7, 7, 128) 69632
['conv5_block2_0_relu[0][0]']

conv5_block2_1_bn (BatchNormalization) (None, 7, 7, 128) 512
['conv5_block2_1_conv[0][0]']
ization)

conv5_block2_1_relu (Activation) (None, 7, 7, 128) 0
['conv5_block2_1_bn[0][0]']
n)

conv5_block2_2_conv (Conv2D) (None, 7, 7, 32) 36864
['conv5_block2_1_relu[0][0]']

conv5_block2_concat (Concatenation) (None, 7, 7, 576) 0
['conv5_block1_concat[0][0]',
te)
'conv5_block2_2_conv[0][0]']

conv5_block3_0_bn (BatchNormalization) (None, 7, 7, 576) 2304
['conv5_block2_concat[0][0]']
ization)

conv5_block3_0_relu (Activation) (None, 7, 7, 576) 0
['conv5_block3_0_bn[0][0]']

```

n)

conv5_block3_1_conv (Conv2D)	(None, 7, 7, 128)	73728
['conv5_block3_0_relu[0][0]']		
conv5_block3_1_bn (BatchNormal	(None, 7, 7, 128)	512
['conv5_block3_1_conv[0][0]']		
ization)		
conv5_block3_1_relu (Activatio	(None, 7, 7, 128)	0
['conv5_block3_1_bn[0][0]']		
n)		
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 32)	36864
['conv5_block3_1_relu[0][0]']		
conv5_block3_concat (Concatena	(None, 7, 7, 608)	0
['conv5_block2_concat[0][0]',		
te)		
'conv5_block3_2_conv[0][0]']		
conv5_block4_0_bn (BatchNormal	(None, 7, 7, 608)	2432
['conv5_block3_concat[0][0]']		
ization)		
conv5_block4_0_relu (Activatio	(None, 7, 7, 608)	0
['conv5_block4_0_bn[0][0]']		
n)		
conv5_block4_1_conv (Conv2D)	(None, 7, 7, 128)	77824
['conv5_block4_0_relu[0][0]']		
conv5_block4_1_bn (BatchNormal	(None, 7, 7, 128)	512
['conv5_block4_1_conv[0][0]']		
ization)		
conv5_block4_1_relu (Activatio	(None, 7, 7, 128)	0
['conv5_block4_1_bn[0][0]']		
n)		
conv5_block4_2_conv (Conv2D)	(None, 7, 7, 32)	36864
['conv5_block4_1_relu[0][0]']		
conv5_block4_concat (Concatena	(None, 7, 7, 640)	0
['conv5_block3_concat[0][0]',		
te)		
'conv5_block4_2_conv[0][0]']		

conv5_block5_0_bn (BatchNormal ['conv5_block4_concat[0][0]' ization)	(None, 7, 7, 640)	2560
conv5_block5_0_relu (Activatio ['conv5_block5_0_bn[0][0]' n)	(None, 7, 7, 640)	0
conv5_block5_1_conv (Conv2D) ['conv5_block5_0_relu[0][0]']	(None, 7, 7, 128)	81920
conv5_block5_1_bn (BatchNormal ['conv5_block5_1_conv[0][0]' ization)	(None, 7, 7, 128)	512
conv5_block5_1_relu (Activatio ['conv5_block5_1_bn[0][0]' n)	(None, 7, 7, 128)	0
conv5_block5_2_conv (Conv2D) ['conv5_block5_1_relu[0][0]']	(None, 7, 7, 32)	36864
conv5_block5_concat (Concatena ['conv5_block4_concat[0][0]', te) 'conv5_block5_2_conv[0][0]']	(None, 7, 7, 672)	0
conv5_block6_0_bn (BatchNormal ['conv5_block5_concat[0][0]' ization)	(None, 7, 7, 672)	2688
conv5_block6_0_relu (Activatio ['conv5_block6_0_bn[0][0]' n)	(None, 7, 7, 672)	0
conv5_block6_1_conv (Conv2D) ['conv5_block6_0_relu[0][0]']	(None, 7, 7, 128)	86016
conv5_block6_1_bn (BatchNormal ['conv5_block6_1_conv[0][0]' ization)	(None, 7, 7, 128)	512
conv5_block6_1_relu (Activatio ['conv5_block6_1_bn[0][0]' n)	(None, 7, 7, 128)	0
conv5_block6_2_conv (Conv2D) ['conv5_block6_1_relu[0][0]']	(None, 7, 7, 32)	36864

conv5_block6_concat (Concatenation) ['conv5_block5_concat[0][0]', 'conv5_block6_2_conv[0][0]']	(None, 7, 7, 704)	0
conv5_block7_0_bn (BatchNormalization) ['conv5_block6_concat[0][0]']	(None, 7, 7, 704)	2816
conv5_block7_0_relu (Activation) ['conv5_block7_0_bn[0][0]']	(None, 7, 7, 704)	0
conv5_block7_1_conv (Conv2D) ['conv5_block7_0_relu[0][0]']	(None, 7, 7, 128)	90112
conv5_block7_1_bn (BatchNormalization) ['conv5_block7_1_conv[0][0]']	(None, 7, 7, 128)	512
conv5_block7_1_relu (Activation) ['conv5_block7_1_bn[0][0]']	(None, 7, 7, 128)	0
conv5_block7_2_conv (Conv2D) ['conv5_block7_1_relu[0][0]']	(None, 7, 7, 32)	36864
conv5_block7_concat (Concatenation) ['conv5_block6_concat[0][0]', 'conv5_block7_2_conv[0][0]']	(None, 7, 7, 736)	0
conv5_block8_0_bn (BatchNormalization) ['conv5_block7_concat[0][0]']	(None, 7, 7, 736)	2944
conv5_block8_0_relu (Activation) ['conv5_block8_0_bn[0][0]']	(None, 7, 7, 736)	0
conv5_block8_1_conv (Conv2D) ['conv5_block8_0_relu[0][0]']	(None, 7, 7, 128)	94208
conv5_block8_1_bn (BatchNormalization) ['conv5_block8_1_conv[0][0]']	(None, 7, 7, 128)	512

conv5_block8_1_relu (Activation)	(None, 7, 7, 128)	0
['conv5_block8_1_bn[0][0]']		
conv5_block8_2_conv (Conv2D)	(None, 7, 7, 32)	36864
['conv5_block8_1_relu[0][0]']		
conv5_block8_concat (Concatenation)	(None, 7, 7, 768)	0
['conv5_block7_concat[0][0]', te) 'conv5_block8_2_conv[0][0]']		
conv5_block9_0_bn (Batch Normalization)	(None, 7, 7, 768)	3072
['conv5_block8_concat[0][0]']		
conv5_block9_0_relu (Activation)	(None, 7, 7, 768)	0
['conv5_block9_0_bn[0][0]']		
conv5_block9_1_conv (Conv2D)	(None, 7, 7, 128)	98304
['conv5_block9_0_relu[0][0]']		
conv5_block9_1_bn (Batch Normalization)	(None, 7, 7, 128)	512
['conv5_block9_1_conv[0][0]']		
conv5_block9_1_relu (Activation)	(None, 7, 7, 128)	0
['conv5_block9_1_bn[0][0]']		
conv5_block9_2_conv (Conv2D)	(None, 7, 7, 32)	36864
['conv5_block9_1_relu[0][0]']		
conv5_block9_concat (Concatenation)	(None, 7, 7, 800)	0
['conv5_block8_concat[0][0]', te) 'conv5_block9_2_conv[0][0]']		
conv5_block10_0_bn (Batch Normalization)	(None, 7, 7, 800)	3200
['conv5_block9_concat[0][0]']		
conv5_block10_0_relu (Activation)	(None, 7, 7, 800)	0
['conv5_block10_0_bn[0][0]']		
conv5_block10_1_conv (Conv2D)	(None, 7, 7, 128)	102400

```

['conv5_block10_0_relu[0][0]']

conv5_block10_1_bn (BatchNorma (None, 7, 7, 128) 512
['conv5_block10_1_conv[0][0]']
lization)

conv5_block10_1_relu (Activati (None, 7, 7, 128) 0
['conv5_block10_1_bn[0][0]']
on)

conv5_block10_2_conv (Conv2D) (None, 7, 7, 32) 36864
['conv5_block10_1_relu[0][0]']

conv5_block10_concat (Concaten (None, 7, 7, 832) 0
['conv5_block9_concat[0][0]',
ate)
'conv5_block10_2_conv[0][0]']

conv5_block11_0_bn (BatchNorma (None, 7, 7, 832) 3328
['conv5_block10_concat[0][0]']
lization)

conv5_block11_0_relu (Activati (None, 7, 7, 832) 0
['conv5_block11_0_bn[0][0]']
on)

conv5_block11_1_conv (Conv2D) (None, 7, 7, 128) 106496
['conv5_block11_0_relu[0][0]']

conv5_block11_1_bn (BatchNorma (None, 7, 7, 128) 512
['conv5_block11_1_conv[0][0]']
lization)

conv5_block11_1_relu (Activati (None, 7, 7, 128) 0
['conv5_block11_1_bn[0][0]']
on)

conv5_block11_2_conv (Conv2D) (None, 7, 7, 32) 36864
['conv5_block11_1_relu[0][0]']

conv5_block11_concat (Concaten (None, 7, 7, 864) 0
['conv5_block10_concat[0][0]',
ate)
'conv5_block11_2_conv[0][0]']

conv5_block12_0_bn (BatchNorma (None, 7, 7, 864) 3456
['conv5_block11_concat[0][0]']
lization)

```

conv5_block12_0_relu (Activati ['conv5_block12_0_bn[0][0]' on)	(None, 7, 7, 864)	0
conv5_block12_1_conv (Conv2D) ['conv5_block12_0_relu[0][0]']	(None, 7, 7, 128)	110592
conv5_block12_1_bn (BatchNorma ['conv5_block12_1_conv[0][0]' lization)	(None, 7, 7, 128)	512
conv5_block12_1_relu (Activati ['conv5_block12_1_bn[0][0]' on)	(None, 7, 7, 128)	0
conv5_block12_2_conv (Conv2D) ['conv5_block12_1_relu[0][0]']	(None, 7, 7, 32)	36864
conv5_block12_concat (Concaten ['conv5_block11_concat[0][0]', ate) 'conv5_block12_2_conv[0][0]']	(None, 7, 7, 896)	0
conv5_block13_0_bn (BatchNorma ['conv5_block12_concat[0][0]' lization)	(None, 7, 7, 896)	3584
conv5_block13_0_relu (Activati ['conv5_block13_0_bn[0][0]' on)	(None, 7, 7, 896)	0
conv5_block13_1_conv (Conv2D) ['conv5_block13_0_relu[0][0]']	(None, 7, 7, 128)	114688
conv5_block13_1_bn (BatchNorma ['conv5_block13_1_conv[0][0]' lization)	(None, 7, 7, 128)	512
conv5_block13_1_relu (Activati ['conv5_block13_1_bn[0][0]' on)	(None, 7, 7, 128)	0
conv5_block13_2_conv (Conv2D) ['conv5_block13_1_relu[0][0]']	(None, 7, 7, 32)	36864
conv5_block13_concat (Concaten ['conv5_block12_concat[0][0]',	(None, 7, 7, 928)	0

```

    ate)
'conv5_block13_2_conv[0][0]']

conv5_block14_0_bn (BatchNorma (None, 7, 7, 928) 3712
['conv5_block13_concat[0][0]']
lization)

conv5_block14_0_relu (Activati (None, 7, 7, 928) 0
['conv5_block14_0_bn[0][0]']
on)

conv5_block14_1_conv (Conv2D) (None, 7, 7, 128) 118784
['conv5_block14_0_relu[0][0]']

conv5_block14_1_bn (BatchNorma (None, 7, 7, 128) 512
['conv5_block14_1_conv[0][0]']
lization)

conv5_block14_1_relu (Activati (None, 7, 7, 128) 0
['conv5_block14_1_bn[0][0]']
on)

conv5_block14_2_conv (Conv2D) (None, 7, 7, 32) 36864
['conv5_block14_1_relu[0][0]']

conv5_block14_concat (Concaten (None, 7, 7, 960) 0
['conv5_block13_concat[0][0]',
ate)
'conv5_block14_2_conv[0][0]']

conv5_block15_0_bn (BatchNorma (None, 7, 7, 960) 3840
['conv5_block14_concat[0][0]']
lization)

conv5_block15_0_relu (Activati (None, 7, 7, 960) 0
['conv5_block15_0_bn[0][0]']
on)

conv5_block15_1_conv (Conv2D) (None, 7, 7, 128) 122880
['conv5_block15_0_relu[0][0]']

conv5_block15_1_bn (BatchNorma (None, 7, 7, 128) 512
['conv5_block15_1_conv[0][0]']
lization)

conv5_block15_1_relu (Activati (None, 7, 7, 128) 0
['conv5_block15_1_bn[0][0]']
on)

```


conv5_block15_2_conv (Conv2D)	(None, 7, 7, 32)	36864	
['conv5_block15_1_relu[0][0]']			
conv5_block15_concat (Concaten	(None, 7, 7, 992)	0	
['conv5_block14_concat[0][0]', ate) 'conv5_block15_2_conv[0][0]']			
conv5_block16_0_bn (BatchNorma	(None, 7, 7, 992)	3968	
['conv5_block15_concat[0][0]'] lization)			
conv5_block16_0_relu (Activati	(None, 7, 7, 992)	0	
['conv5_block16_0_bn[0][0]'] on)			
conv5_block16_1_conv (Conv2D)	(None, 7, 7, 128)	126976	
['conv5_block16_0_relu[0][0]']			
conv5_block16_1_bn (BatchNorma	(None, 7, 7, 128)	512	
['conv5_block16_1_conv[0][0]'] lization)			
conv5_block16_1_relu (Activati	(None, 7, 7, 128)	0	
['conv5_block16_1_bn[0][0]'] on)			
conv5_block16_2_conv (Conv2D)	(None, 7, 7, 32)	36864	
['conv5_block16_1_relu[0][0]']			
conv5_block16_concat (Concaten	(None, 7, 7, 1024)	0	
['conv5_block15_concat[0][0]', ate) 'conv5_block16_2_conv[0][0]']			
bn (BatchNormalization)	(None, 7, 7, 1024)	4096	
['conv5_block16_concat[0][0]']			
relu (Activation)	(None, 7, 7, 1024)	0	['bn[0][0]']
flatten_6 (Flatten)	(None, 50176)	0	['relu[0][0]']
dense_29 (Dense)	(None, 728)	36528856	
['flatten_6[0][0]']			
dropout_14 (Dropout)	(None, 728)	0	
['dense_29[0][0]']			

```

batch_normalization_14 (BatchN (None, 728)          2912
['dropout_14[0][0]']
ormalization)

```

```

dense_30 (Dense)          (None, 3)          2187
['batch_normalization_14[0][0]']

```

```

=====
=====

```

```

Total params: 43,571,459
Trainable params: 43,486,355
Non-trainable params: 85,104

```

```

-----
-----

```

```
[ ]:
```

```

[ ]: ImageDataGenerator_fordenseNet = ImageDataGenerator(rescale = 1.0/255.0,
    width_shift_range = 0.1 ,
    height_shift_range = 0.1 ,
    rotation_range = 10 ,
    horizontal_flip = True
)
trainGenerator_Densenet = ImageDataGenerator_fordenseNet.
    ↳flow_from_directory(train_folder_path ,
        target_size = (224 ,224) ,
        color_mode = "rgb",
        batch_size = 7 ,
        class_mode = 'categorical',
        shuffle = True
    )
test_data_generator_Densenet = ImageDataGenerator(rescale = 1/255 )

testGenerator_Densenet = test_data_generator_Densenet.
    ↳flow_from_directory(test_folder_path ,
        target_size = (224 ,224) ,
        color_mode = "rgb",
        batch_size = 7 ,
        class_mode = 'categorical',
        shuffle = True
    )

```

Found 251 images belonging to 3 classes.

Found 66 images belonging to 3 classes.

Take loss function as categorical cross-entropy

Take Adam as an optimizer

Use early stopping to prevent overfitting

Try with 15 number of epoch and batch size with seven, also try various values to see the impact on results

Train the model using the generator and test the accuracy of the test data at every epoch

```
[ ]: estop = EarlyStopping( monitor = 'val_loss', min_delta = 0.1 ,
    patience = 2 , mode = 'min' , restore_best_weights = True)

[ ]: historyDensenet = model_transfer1.fit(trainGenerator_Densenet , epochs = 15 ,
    validation_data = testGenerator_Densenet ,
    validation_steps = testGenerator_Densenet.n // testGenerator_Densenet.
    ↪ batch_size ,
    steps_per_epoch = trainGenerator_Densenet.n // trainGenerator_Densenet.
    ↪ batch_size ,
    callbacks = [estop ,]
    )
```

Epoch 1/15

35/35 [=====] - 18s 512ms/step - loss: 0.1366 -
accuracy: 0.9426 - val_loss: 7.4786 - val_accuracy: 0.0159

Epoch 2/15

35/35 [=====] - 18s 502ms/step - loss: 0.1599 -
accuracy: 0.9303 - val_loss: 6.3801 - val_accuracy: 0.0635

Epoch 3/15

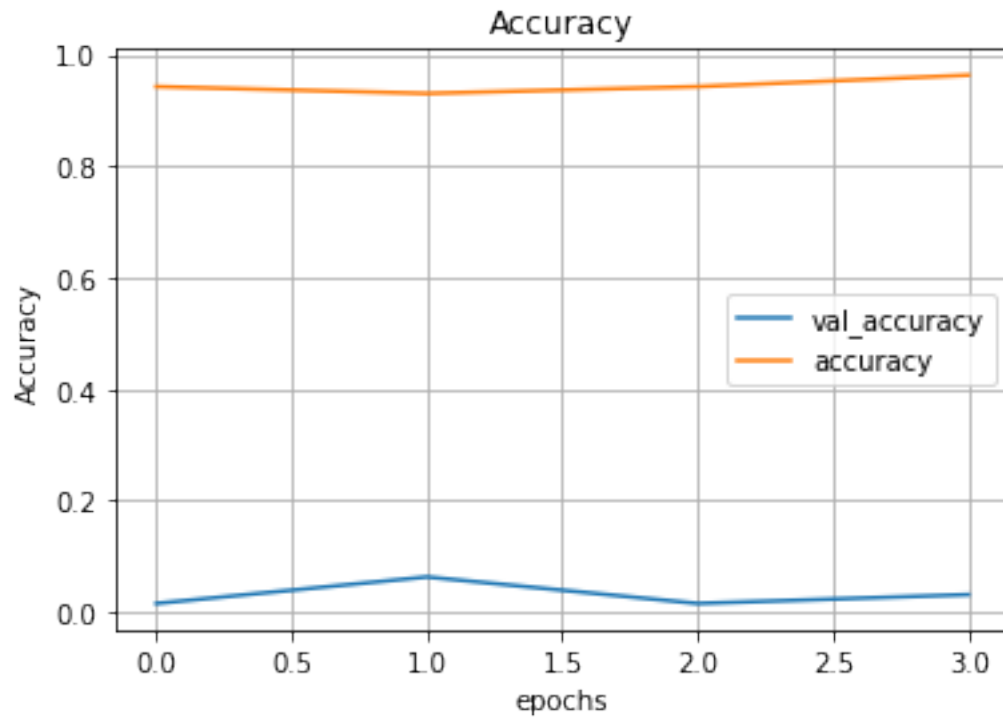
35/35 [=====] - 17s 490ms/step - loss: 0.2168 -
accuracy: 0.9426 - val_loss: 7.0555 - val_accuracy: 0.0159

Epoch 4/15

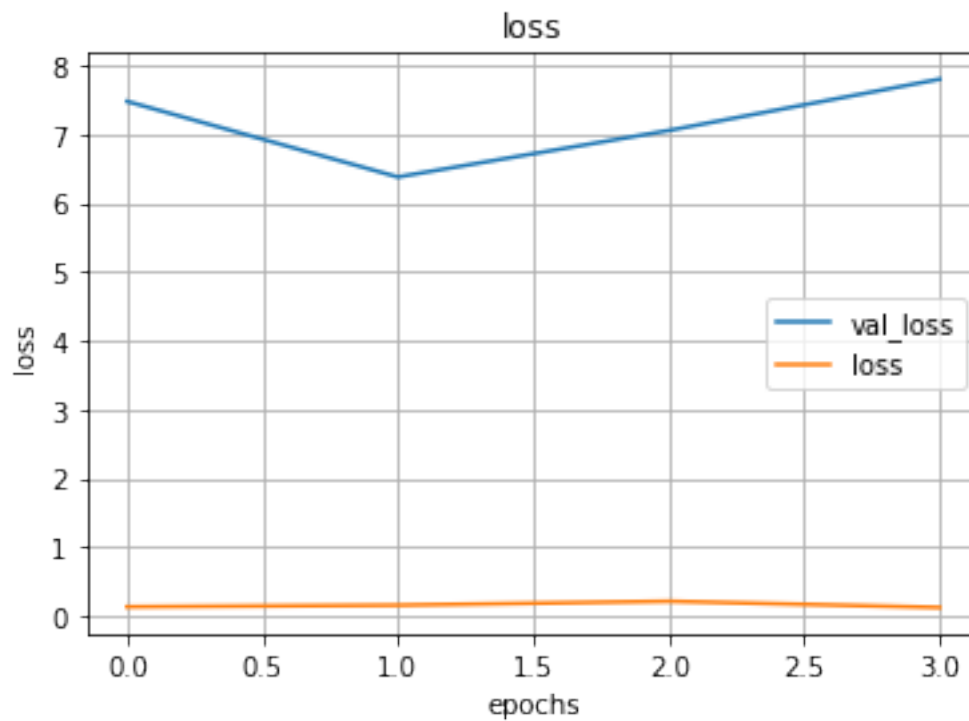
35/35 [=====] - 18s 504ms/step - loss: 0.1264 -
accuracy: 0.9631 - val_loss: 7.8010 - val_accuracy: 0.0317

Plot the training and validation accuracy, and the loss

```
[ ]: plt.plot(historyDensenet.history['val_accuracy'] , label = 'val_accuracy')
plt.plot(historyDensenet.history['accuracy'] , label = 'accuracy')
plt.xlabel('epochs')
plt.ylabel('Accuracy')
plt.title('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```



```
[ ]: plt.plot(historyDensenet.history['val_loss'], label = 'val_loss')
plt.plot(historyDensenet.history['loss'], label = 'loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.title('loss')
plt.legend()
plt.grid(True)
plt.show()
```



```
[ ]: pred_denseNet = model_transfer1.predict(testGenerator_Densenet )
```

```
pred_labels_DenseNet = np.argmax(pred_denseNet, axis = -1)
```

```
[ ]: print(classification_report(testGenerator_Densenet.classes
                                , pred_labels_DenseNet , zero_division = 0 ))
```

	precision	recall	f1-score	support
0	0.12	0.08	0.09	26
1	0.19	0.25	0.22	20
2	0.39	0.45	0.42	20
accuracy			0.24	66
macro avg	0.23	0.26	0.24	66
weighted avg	0.22	0.24	0.23	66

```
[ ]:
```

```
[ ]:
```