

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

Mounted at /content/drive

import warnings
warnings.filterwarnings("ignore")

pip install keras_applications
```

```
Requirement already satisfied: keras_applications in /usr/local/lib/python3.7/
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dis
```

```
train_path = "drive/MyDrive/MiniProject/train"
valid_path = "drive/MyDrive/MiniProject/validation"
test_path = "drive/MyDrive/MiniProject/test"
```

```
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from keras.models import Model
from keras.layers import Dense, MaxPool2D, Conv2D
import keras
from keras.applications.resnet import preprocess_input
```

```
import pandas as pd
import numpy as np
import os
import shutil
import glob
import matplotlib.pyplot as plt
```

```
train_data_gen = ImageDataGenerator(preprocessing_function= preprocess_input,
                                     zoom_range= 0.2,
                                     horizontal_flip= True,
                                     shear_range= 0.2,

                                     )
```

```
train = train_data_gen.flow_from_directory(directory= train_path,
                                           target_size=(224,224))
```

```
Found 7800 images belonging to 2 classes.
```



```
test_data_gen = ImageDataGenerator(preprocessing_function= preprocess_input )

test = train_data_gen.flow_from_directory(directory= test_path ,
                                          target_size=(224,224),
                                          shuffle= False)
```

```
Found 800 images belonging to 2 classes.
```

```
# Covid +ve X-Ray is represented by 0 and Normal is represented by 1
```

```
class_type = {0:'Covid', 1 : 'Normal'}
```

```
# to visualize the images in the traing data denerator
```

```
t_img , label = train.next()
```

```
# function when called will prot the images
```

```
def plotImages(img_arr, label):
    """
    input  :- images array
    output :- plots the images
    """
```

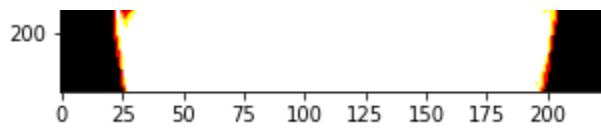
```
    for im, l in zip(img_arr,label) :
        plt.figure(figsize= (5,5))
        plt.imshow(im, cmap = 'gray')
        plt.title(im.shape)
        plt.axis = False
        plt.show()
```

```
# function call to plot the images
```

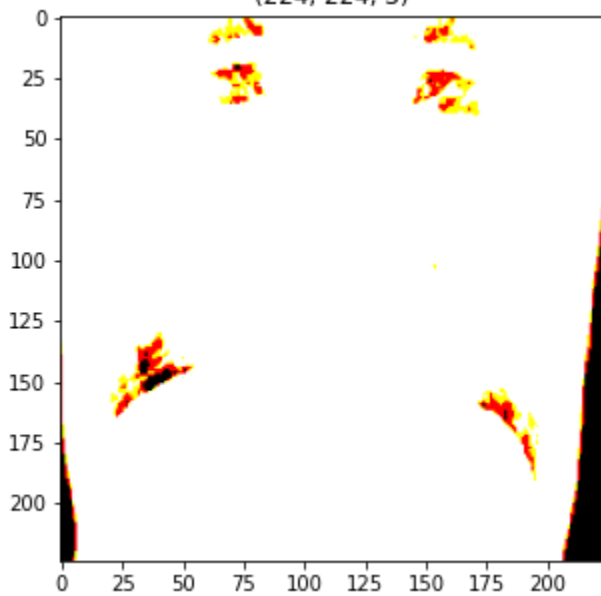
```
plotImages(t_img, label)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for fl
```

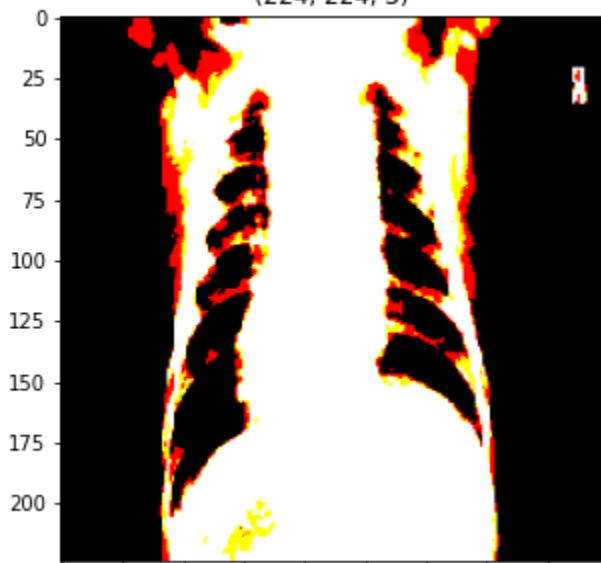


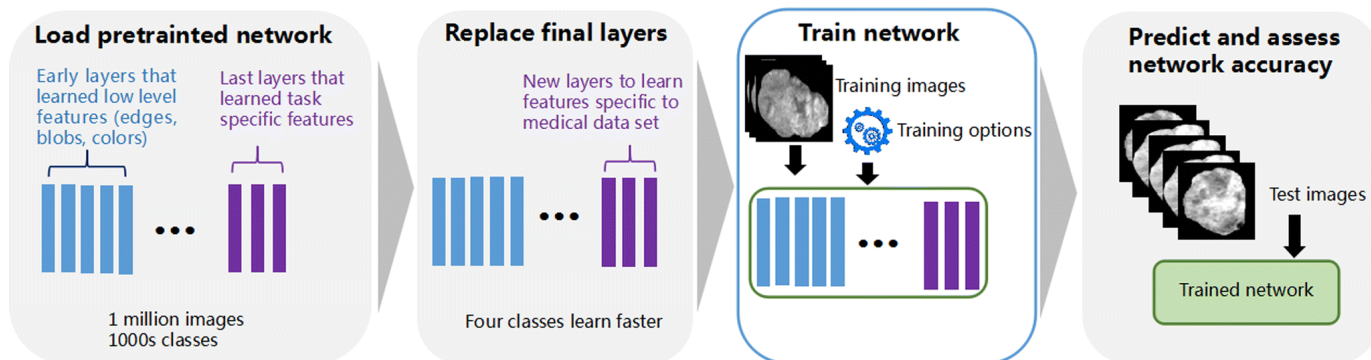


Clipping input data to the valid range for imshow with RGB data ([0..1] for fl
(224, 224, 3)



Clipping input data to the valid range for imshow with RGB data ([0..1] for fl
(224, 224, 3)



**C**

```

from tensorflow.keras.applications.resnet50 import ResNet50
from keras.layers import Flatten , Dense, Dropout , MaxPool2D

res = ResNet50( input_shape=(224,224,3), include_top= False, weights='imagenet') #

for layer in res.layers:
    layer.trainable = False    # Dont Train the parameters again

x = Flatten()(res.output)
x = Dense(units=2 , activation='sigmoid', name = 'predictions' )(x)

# creating our model.
model = Model(res.input, x)

```

conv2_block1_1_relu	(Activation)	(None, 56, 56, 64)	0	conv2_block1_1_relu
conv2_block1_2_conv	(Conv2D)	(None, 56, 56, 64)	36928	conv2_block1_2_conv
conv2_block1_2_bn	(BatchNormali	(None, 56, 56, 64)	256	conv2_block1_2_bn
conv2_block1_2_relu	(Activation)	(None, 56, 56, 64)	0	conv2_block1_2_relu
conv2_block1_0_conv	(Conv2D)	(None, 56, 56, 256)	16640	pool1_pool[0]
conv2_block1_3_conv	(Conv2D)	(None, 56, 56, 256)	16640	conv2_block1_3_conv
conv2_block1_0_bn	(BatchNormali	(None, 56, 56, 256)	1024	conv2_block1_0_bn
conv2_block1_3_bn	(BatchNormali	(None, 56, 56, 256)	1024	conv2_block1_3_bn
conv2_block1_add	(Add)	(None, 56, 56, 256)	0	conv2_block1_add
conv2_block1_out	(Activation)	(None, 56, 56, 256)	0	conv2_block1_out
conv2_block2_1_conv	(Conv2D)	(None, 56, 56, 64)	16448	conv2_block2_1_conv
conv2_block2_1_bn	(BatchNormali	(None, 56, 56, 64)	256	conv2_block2_1_bn
conv2_block2_1_relu	(Activation)	(None, 56, 56, 64)	0	conv2_block2_1_relu
conv2_block2_2_conv	(Conv2D)	(None, 56, 56, 64)	36928	conv2_block2_2_conv

```
Epoch 00003: val_accuracy improved from 0.85742 to 0.87500, saving model to be
Epoch 4/30
10/10 [=====] - 86s 9s/step - loss: 2.2661 - accuracy

Epoch 00004: val_accuracy improved from 0.87500 to 0.88672, saving model to be
Epoch 5/30
10/10 [=====] - 77s 8s/step - loss: 1.9272 - accuracy

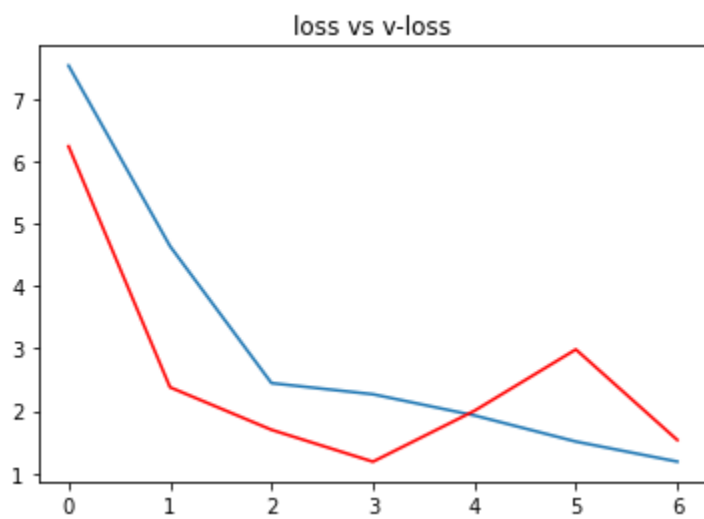
Epoch 00005: val_accuracy did not improve from 0.88672
Epoch 6/30
10/10 [=====] - 71s 7s/step - loss: 1.5086 - accuracy

Epoch 00006: val_accuracy did not improve from 0.88672
Epoch 7/30
10/10 [=====] - 67s 7s/step - loss: 1.1878 - accuracy

Epoch 00007: val_accuracy did not improve from 0.88672
Epoch 00007: early stopping
```

```
## load only the best model
```

```
plt.show()
```



```
# to display the image
plt.imshow(img[0]/255, cmap = "gray")
plt.title("input image")
plt.show()
```