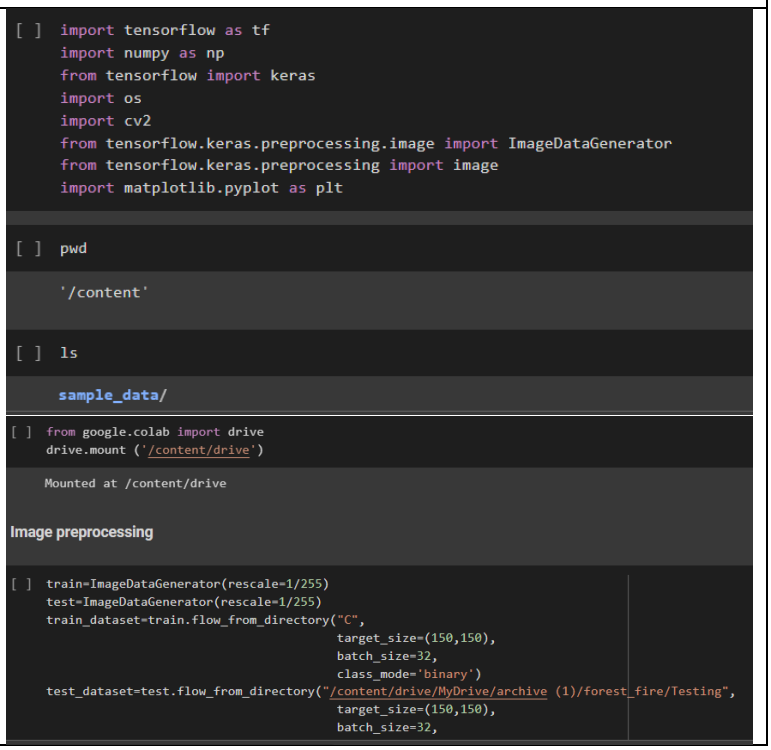
Project Development Phase
Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID49509
Project Name	Emerging methods for early detection of forest fires
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	

```
Found 1852 images belonging to 2 classes.
Found 68 images belonging to 2 classes.
```

```
[ ] test_dataset.class_indices
```

```
{'fire': 0, 'no fire': 1}
```

Model building

```
[ ] model=keras.Sequential()
    model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))
    model.add(keras.layers.MaxPool2D(2,2))
    model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))
    model.add(keras.layers.MaxPool2D(2,2))
    model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
    model.add(keras.layers.MaxPool2D(2,2))
    model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
    model.add(keras.layers.MaxPool2D(2,2))
    model.add(keras.layers.Flatten())
    model.add(keras.layers.Dense(512,activation='relu'))
```

```
[ ] model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

```
[ ] r=model.fit(train_dataset,epochs=7,validation_data=test_dataset)
```

```
Epoch 1/7
58/58 [==============================] - 257s 4s/step - loss: 0.3354 - accuracy: 0.8607 - val_loss: 0.3862 - val_accuracy: 0.8676
Epoch 2/7
58/58 [==============================] - 84s 1s/step - loss: 0.1499 - accuracy: 0.9460 - val_loss: 0.2213 - val_accuracy: 0.9265
Epoch 3/7
58/58 [==============================] - 79s 1s/step - loss: 0.1465 - accuracy: 0.9449 - val_loss: 0.2286 - val_accuracy: 0.8971
Epoch 4/7
58/58 [==============================] - 79s 1s/step - loss: 0.1017 - accuracy: 0.9687 - val_loss: 0.3571 - val_accuracy: 0.8676
Epoch 5/7
58/58 [==============================] - 83s 1s/step - loss: 0.1269 - accuracy: 0.9590 - val_loss: 0.2900 - val_accuracy: 0.9265
Epoch 6/7
58/58 [==============================] - 80s 1s/step - loss: 0.0826 - accuracy: 0.9725 - val_loss: 0.2073 - val_accuracy: 0.8824
Epoch 7/7
58/58 [==============================] - 79s 1s/step - loss: 0.1157 - accuracy: 0.9649 - val_loss: 0.2886 - val_accuracy: 0.8824
```

```
[ ] predictions=model.predict(test_dataset)
    predictions=np.round(predictions)
```

```
3/3 [==============================] - 1s 210ms/step
```

```
[ ] predictions
```

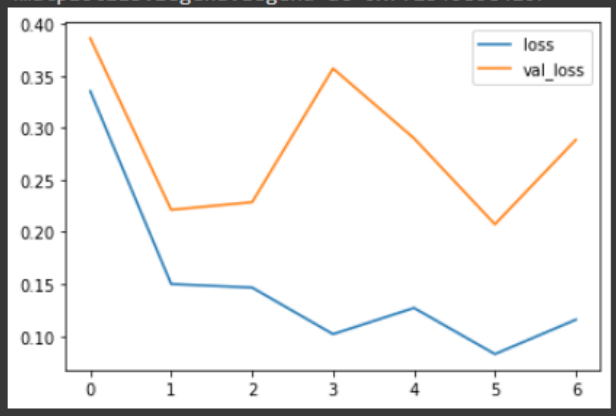
```
array([[1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [0.],
       [0.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [0.],
       [0.],
       [1.]], dtype=float32)
```

```
[ ] print(len(predictions))
```

```
68
```

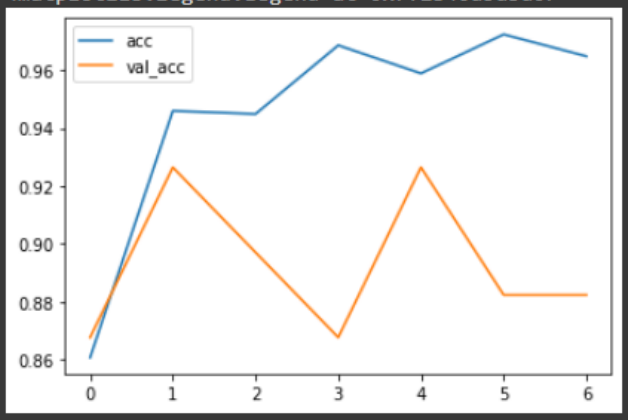
```
import matplotlib.pyplot as plt
plt.plot(r.history['loss'],label='loss')
plt.plot(r.history['val_loss'],label='val_loss')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f2b40e6c410>
```



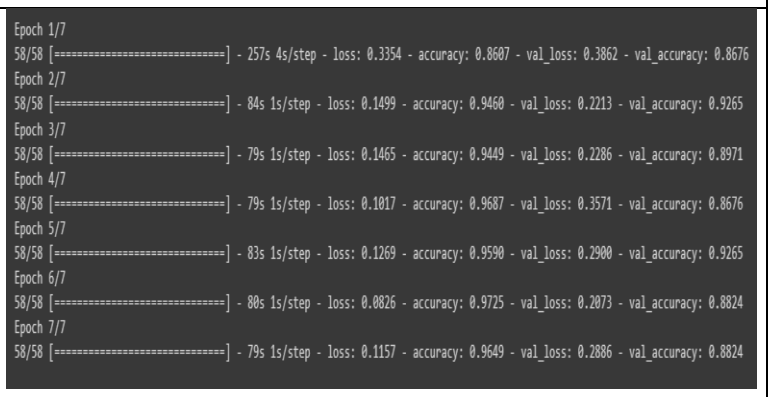

```
plt.plot(r.history['accuracy'],label='acc')
plt.plot(r.history['val_accuracy'],label='val_acc')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f2b40d5d3d0>
```



Testing the model

```
[ ]  def predictImage(filename):
       img1=image.load_img(filename,target_size=(150,150))
       plt.imshow(img1)
       Y=image.img_to_array(img1)
       X=np.expand_dims(Y,axis=0)
       val=model.predict(X)
       print(val)
       if val==1:
         plt.xlabel("No fire",fontsize=30)
       elif val==0:
         plt.xlabel("fire",fontsize=30)
```

			
2.	Accuracy	Training Accuracy - 96.49 Validation Accuracy -88.24	
3.	Confidence Score (Only Yolo Projects)	Class Detected - 2 classes Confidence Score - 80