

## EXPERIMENT-8

**AIM:** To fine tune a pre-trained CNN architecture and evaluate its performance on a dataset.

### **CODE and OUTPUT:**

*Dataset Link:*

<https://drive.google.com/file/d/1owuiGKC9L34daUkfJgfijM6j6OSHGCIS/view?usp=sharing>

<https://drive.google.com/file/d/1B0XAeCkBT1hzIleiGFUFvZd5jmw81tUI/view?usp=sharing>

```
# importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from glob import glob
from keras.models import Model
from keras.layers import Flatten, Dense
from keras.applications import VGG19
from keras.preprocessing.image import img_to_array, ImageDataGenerator
```

```
!unzip images.zip
```

```
traindf=pd.read_csv("emergency_train.csv",dtype=str)
traindf.head(10)
```

	image_names	emergency_or_not
0	1503.jpg	0
1	1420.jpg	0
2	1764.jpg	0
3	1356.jpg	0
4	1117.jpg	0
5	45.jpg	1
6	2313.jpg	0
7	1657.jpg	0
8	294.jpg	1
9	999.jpg	0

```
train_datagen = ImageDataGenerator(  
    rescale = 1./255,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True)
```

```
test_datagen = ImageDataGenerator(rescale=1./255.,validation_split=0.10)
```

```
valid_datagen=ImageDataGenerator(rescale=1./255.,validation_split=0.15)
```

```
train_generator=train_datagen.flow_from_dataframe(  
    dataframe=train_df,  
    directory="images",  
    x_col='image_names',  
    y_col='emergency_or_not',  
    subset="training",  
    batch_size=16,  
    seed=42,  
    shuffle=True,  
    class_mode="binary",  
    target_size=(224, 224))
```

Found 1646 validated image filenames belonging to 2 classes.

```
test_generator=test_datagen.flow_from_dataframe(  
    dataframe=train_df,  
    directory="images",  
    x_col='image_names',  
    y_col='emergency_or_not',  
    subset="validation",  
    batch_size=16,  
    seed=42,  
    shuffle=True,  
    class_mode="binary",  
    target_size=(224, 224))
```

Found 164 validated image filenames belonging to 2 classes.

```
valid_generator=valid_datagen.flow_from_dataframe(  
    dataframe=train_df,  
    directory="images",  
    x_col='image_names',  
    y_col='emergency_or_not',  
    subset="validation",  
    batch_size=16,  
    seed=42,  
    shuffle=True,  
    class_mode="binary",  
    target_size=(224, 224))
```

Found 246 validated image filenames belonging to 2 classes.

```
# loading the weights of VGG16 without the top layer. These weights are trained on Imagenet dataset.
vgg = VGG16(include_top=False, weights='imagenet', input_shape=(224, 224, 3))

# this will exclude the initial layers from training phase as there are already been trained.
for layer in vgg.layers:
    layer.trainable = False

x = Flatten()(vgg.output)
x = Dense(1, activation = 'sigmoid')(x) # adding the output layer with softmax function as this is a multi label classification problem.

model = Model(inputs = vgg.input, outputs = x)
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
```

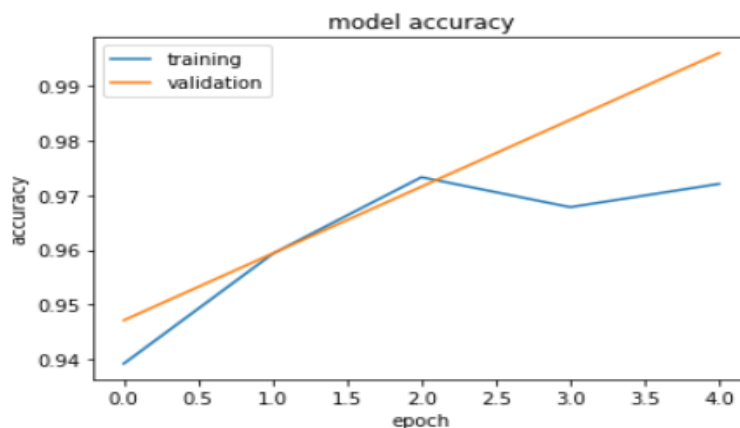
```
train_generator.class_indices
```

```
{'0': 0, '1': 1}
```

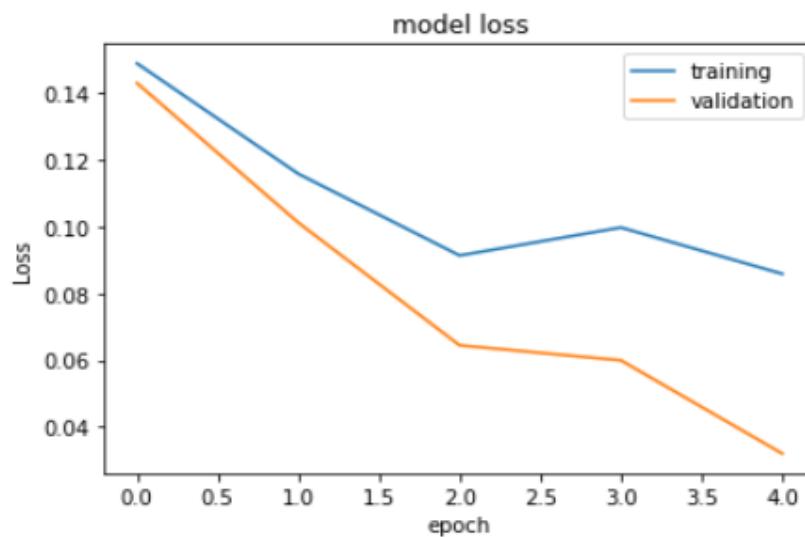
```
history = model.fit_generator(train_generator, epochs = 5, validation_data = valid_generator)
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: `Model.fit_generator` is deprecated and will be
warnings.warn("`Model.fit_generator` is deprecated and ")
Epoch 1/5
103/103 [=====] - 22s 218ms/step - loss: 0.1490 - accuracy: 0.9392 - val_loss: 0.1431 - val_accuracy: 0.9472
Epoch 2/5
103/103 [=====] - 22s 214ms/step - loss: 0.1159 - accuracy: 0.9593 - val_loss: 0.1012 - val_accuracy: 0.9593
Epoch 3/5
103/103 [=====] - 22s 214ms/step - loss: 0.0914 - accuracy: 0.9733 - val_loss: 0.0644 - val_accuracy: 0.9715
Epoch 4/5
103/103 [=====] - 22s 214ms/step - loss: 0.0998 - accuracy: 0.9678 - val_loss: 0.0599 - val_accuracy: 0.9837
Epoch 5/5
103/103 [=====] - 22s 214ms/step - loss: 0.0859 - accuracy: 0.9721 - val_loss: 0.0319 - val_accuracy: 0.9959
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['training', 'validation'], loc='best')
plt.show()
```



```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('Loss')
plt.xlabel('epoch')
plt.legend(['training', 'validation'], loc='best')
plt.show()
```



```
scores=model.evaluate_generator(test_generator, steps=50)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:1877: UserWarning: `Model.evaluate\_generator` is deprecated and will be removed in a future version. Use `Model.evaluate` instead.  
 warnings.warn("`Model.evaluate\_generator` is deprecated and will be removed in a future version. Use `Model.evaluate` instead.")  
 WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps\_per\_epoch \* epochs` batches before the end of the last epoch. This may be caused by a too small dataset.  
 accuracy: 99.39%