In [1]:
```
# Importing Keras Library
import keras
# Importing ImageDataGenerator class from keras
from keras.preprocessing.image import ImageDataGenerator
```

In [3]:
```
# Define the parameters / arguments for ImageDataGenerator class
train_datagen= ImageDataGenerator(rescale=1./255,
                                  shear_range=0.2,
                                  rotation_range=180,
                                  horizontal_flip=True,
                                  zoom_range=0.2)
test_datagen= ImageDataGenerator(rescale=1./255)
```

In [5]:
```
#Applying ImageDataGenerator functionality to training_set
x_train=train_datagen.flow_from_directory('C:\\Users\\mohan\\Downloads\\trainset',
                                          target_size=(64,64),batch_size=32,
                                          class_mode='binary')
# Note if more than 2 categories class_mode='categorical'
# Note Change your Directory path before executing this Cell
```

```
Found 240 images belonging to 2 classes.
```

In [7]:
```
# Applying ImageDataGenerator functionality to testing_set
x_test=train_datagen.flow_from_directory('C:\\Users\\mohan\\Downloads\\testset',
                                         target_size=(64,64),batch_size=32,
                                         class_mode='binary')
# Note if more than 2 categories class_mode='categorical'
# Note Change your Directory path before executing this Cell
```

```
Found 60 images belonging to 2 classes.
```

In [8]:
```
print(x_train.class_indices)
```

```
{'Found Missing': 0, 'Normal': 1}
```

In [9]:
```
# Importing Model Building Libraries
# To define linear intialisation import sequential
from tensorflow.keras.models import Sequential
# To add Hidden Layers import Dense
from tensorflow.keras.layers import Dense
# To Create Convolution Layer import convolution2D
from tensorflow.keras.layers import Conv2D
# Import Max pooling layer to extra maximum features
from tensorflow.keras.layers import MaxPool2D
# Importing Flatten Layer
from tensorflow.keras.layers import Flatten
```

In [10]:
```
# Intialising the model
model=Sequential()
```

In [11]:
```
# Adding Convolutional Layer
model.add(Conv2D( 32,3,3,input_shape=(64,64,3),activation='relu'))


#1st param in conv2D = no of feature detectors(or say feature detector matrix) to form feature map
#2nd,3rd param = size of feat.Detect(or say feature detector matrix size ie,. 3 X 3 here )
#4th param = Expected image input shape(every img sould be of same size so here 64 X 64 and 3 means its an RGB img
#5th param= Activation fun
```

In [12]:
```
# Adding Max Pooling Layer
model.add(MaxPool2D(pool_size=(2,2))) # 2,2 size of matrix
```

In [13]:
```
# Adding Flatten Layer
model.add(Flatten())
```

In [14]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 21, 21, 32)        896

 max_pooling2d (MaxPooling2D  (None, 10, 10, 32)       0
 )

 flatten (Flatten)           (None, 3200)              0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

In [15]:
```
# Adding Hidden Layer
model.add(Dense(units=128,activation='relu',kernel_initializer='random_uniform'))
```

In [16]:
```
# Adding Output Layer
model.add(Dense(units=1,activation='sigmoid',kernel_initializer='random_uniform'))
```

In [17]:
```
# Configure the Learning Process
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

In [18]:
```
# Training the model
model.fit_generator(x_train,steps_per_epoch=8,
                    validation_data=x_test,epochs=128,
                    validation_steps=2)
```

```
C:\Users\mohan\AppData\Local\Temp\ipykernel_14696\3961316105.py:2: UserWarning: `Model.fit_generator` is deprec
ated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=8,

Epoch 1/128
8/8 [==============================] - 13s 1s/step - loss: 0.6492 - accuracy: 0.6542 - val_loss: 0.6533 - val_a
ccuracy: 0.6667
Epoch 2/128
8/8 [==============================] - 7s 890ms/step - loss: 0.6340 - accuracy: 0.6667 - val_loss: 0.6625 - val
_accuracy: 0.6667
Epoch 3/128
8/8 [==============================] - 7s 912ms/step - loss: 0.6345 - accuracy: 0.6667 - val_loss: 0.6585 - val
_accuracy: 0.6667
Epoch 4/128
8/8 [==============================] - 8s 967ms/step - loss: 0.6282 - accuracy: 0.6667 - val_loss: 0.6653 - val
_accuracy: 0.6667
Epoch 5/128
8/8 [==============================] - 8s 951ms/step - loss: 0.6327 - accuracy: 0.6667 - val_loss: 0.6713 - val
_accuracy: 0.6667
Epoch 6/128
```

In [19]:
```
# Saving the trained model with .h5 extension
model.save('Missing.h5')
```

In [20]:
```
# Importing the Model Libraries
import cv2
import numpy as np
import smtplib
from keras.preprocessing import image
import tensorflow as tf
import os
name = ["Found Missing","Normal"]
```

In [26]:
```
# Loading the Saved model
model = tf.keras.models.load_model('Missing.h5')
```

In [35]:
```
# Giving Random Image Path
from PIL import Image

img = Image.open(r"C:\Users\mohan\Downloads\testset\Found Missing\gettyimages-1158031528-612x612.jpg")
img = img.resize((64, 64))  # Resizing the image
x = np.array(img)
x = np.expand_dims(x, axis=0)
```

In [38]:
```
# Classes of Prediction
pred = model.predict(x)
pred_classes = pred.argmax(axis=-1)
```

```
1/1 [==============================] - 0s 58ms/step
```

In [39]:
```
pred[0][0]
```

Out[39]:
```
1.0
```

In [1]:

```
import cv2
import numpy as np
from keras.preprocessing import image
import tensorflow as tf
from twilio.rest import Client
from PIL import Image

model = tf.keras.models.load_model('Missing.h5')
name = ["Found Missing", "Normal"]

# Load the image
img = Image.open(r"C:\Users\mohan\Downloads\testset\Found Missing\gettyimages-1158031528-612x612.jpg")
img = img.resize((64, 64))  # Resize the image
x = np.array(img)
x = np.expand_dims(x, axis=0)

# Predict the image
pred = model.predict(x)
pred_class = np.argmax(pred, axis=1)[0]
print(pred_class)

if pred_class == 0:
    from twilio.rest import Client
    account_sid = 'ACc36f587b05c6cae6b4e87a0e72dbc9ed'
    auth_token = '6be1180990b5c13b870147323b7303fc'
    client = Client(account_sid, auth_token)
    message = client.messages.create(
    to='+916303031647',
        from_='+14175282474',
    body='Found the Missing at 17.3984° N, 78.5583° E'
    )
    print(message.sid)
    print("Found Missing")
    print('SMS Sent')
else:
    print("Normal")
```

```
1/1 [==============================] - 0s 437ms/step
0
SM8d7f1921b97010a2256b2c6ee1303a16
Found Missing
SMS Sent
```

In [43]:
```
pip install twilio
```

```
Requirement already satisfied: twilio in c:\users\mohan\anaconda3\lib\site-packages (8.4.0)
Requirement already satisfied: pytz in c:\users\mohan\anaconda3\lib\site-packages (from twilio) (2022.1)
Requirement already satisfied: requests>=2.0.0 in c:\users\mohan\anaconda3\lib\site-packages (from twilio) (2.28.
1)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in c:\users\mohan\anaconda3\lib\site-packages (from twilio)
(2.4.0)
Requirement already satisfied: aiohttp-retry>=2.8.3 in c:\users\mohan\anaconda3\lib\site-packages (from twilio)
(2.8.3)
Requirement already satisfied: aiohttp>=3.8.4 in c:\users\mohan\anaconda3\lib\site-packages (from twilio) (3.8.4)
Requirement already satisfied: attrs>=17.3.0 in c:\users\mohan\anaconda3\lib\site-packages (from aiohttp>=3.8.4->
twilio) (21.4.0)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in c:\users\mohan\anaconda3\lib\site-packages (from ai
ohttp>=3.8.4->twilio) (4.0.2)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\mohan\anaconda3\lib\site-packages (from aiohttp>=
3.8.4->twilio) (6.0.4)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\mohan\anaconda3\lib\site-packages (from aiohttp>=3.
8.4->twilio) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\mohan\anaconda3\lib\site-packages (from aiohttp>=3.8.
4->twilio) (1.3.1)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\mohan\anaconda3\lib\site-packages (from aiohttp>=3.8.4-
>twilio) (1.9.2)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in c:\users\mohan\anaconda3\lib\site-packages (from a
iohttp>=3.8.4->twilio) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mohan\anaconda3\lib\site-packages (from requests>=
2.0.0->twilio) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\mohan\anaconda3\lib\site-packages (from requests
>=2.0.0->twilio) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\users\mohan\anaconda3\lib\site-packages (from requests>=2.0.0->
twilio) (3.3)
Note: you may need to restart the kernel to use updated packages.

WARNING: Ignoring invalid distribution -treamlit (c:\users\mohan\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -treamlit (c:\users\mohan\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -treamlit (c:\users\mohan\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -treamlit (c:\users\mohan\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -treamlit (c:\users\mohan\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -treamlit (c:\users\mohan\anaconda3\lib\site-packages)
```