In [15]:
```
import os
```

```
os.getcwd()
```

Out[15]:
```
'C:\\Users\\NIPUN.S\\Desktop'
```

In [3]:
```
os.chdir('C:\\Users\\NIPUN.S\\Desktop')
```

In [4]:
```
PATH = os.getcwd() print(PATH)
```

```
C:\Users\NIPUN.S\Desktop
```

In [5]:
```
data_path = PATH + '/data'
data_dir_list = os.listdir(data_path)
```

In [6]:
```
print(data_dir_list)
```

```
['Female', 'Male', 'Transgender']
```

In [7]:
```
img_rows=224
img_cols=224 num_channel=3
num_epoch=2

img_data_list=[]
classes_names_list=[]
```

In [8]:
```
import cv2

for dataset in data_dir_list:
    classes_names_list.append(dataset)     print ('Loading
images from {} folder\n'.format(dataset))
img_list=os.listdir(data_path+'/'+ dataset)     for img in
img_list:
        input_img=cv2.imread(data_path + '/'+ dataset + '/'+ img )
input_img_resize=cv2.resize(input_img,(img_rows, img_cols))
img_data_list.append(input_img_resize)
```

```
Loading images from Female folder

Loading images from Male folder

Loading images from Transgender folder
```

In [9]:
```
num_classes = len(classes_names_list)
```

In [10]:
```
import numpy as np img_data =
np.array(img_data_list) img_data =
img_data.astype('float32') img_data /=
255
```

In [11]:
```
print (img_data.shape)
```

```
img_data = img_data.reshape(img_data.shape[0], img_data.shape[1], img_data.shape[2]
```

```
(150, 224, 224, 3)
```

In [12]:
```
num_of_samples = img_data.shape[0]
input_shape = img_data[0].shape
```

In [13]:
```
classes = np.ones((num_of_samples,), dtype='int64')
classes[0:50]=0 classes[50:100]=1 classes[100:]=2
```

In [17]:
```
from keras.utils import to_categorical

# convert class labels to on-hot encoding classes
= to_categorical(classes, num_classes)
```

In [18]:
```
from sklearn.utils import shuffle

x, y = shuffle(img_data, classes, random_state=2)
```

In [19]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_sta
```

In [20]:
```
from keras.models import Sequential from
keras.layers import Dense, Dropout, Flatten from
keras.layers import Conv2D, MaxPooling2D
```

In [21]:
```
model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.5))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(64, activation='relu')) model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

In [22]:
```
model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=["accur
```

In [23]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
conv2d (Conv2D)             (None, 222, 222, 32)      896
conv2d_1 (Conv2D)           (None, 220, 220, 32)      9248
max_pooling2d (MaxPooling2  (None, 110, 110, 32)      0
D)
dropout (Dropout)           (None, 110, 110, 32)      0
conv2d_2 (Conv2D)           (None, 108, 108, 64)      18496
conv2d_3 (Conv2D)           (None, 106, 106, 64)      36928
max_pooling2d_1 (MaxPoolin  (None, 53, 53, 64)        0
g2D)
dropout_1 (Dropout)         (None, 53, 53, 64)        0
flatten (Flatten)           (None, 179776)            0
dense (Dense)               (None, 64)                11505728
dropout_2 (Dropout)         (None, 64)                0
dense_1 (Dense)             (None, 3)                 195
=================================================================
Total params: 11571491 (44.14 MB)
Trainable params: 11571491 (44.14 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [26]:
```
model.layers[1].get_weights()[1]
```

Out[26]:
```
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., 0., 0., 0., 0., 0., 0.],
      dtype=float32)
```

In [27]:
```
model.get_config()
```

Out[27]:
```
{'name': 'sequential',
 'layers': [{'module': 'keras.layers',
   'class_name': 'InputLayer',
   'config': {'batch_input_shape': (None, 224, 224, 3),
    'dtype': 'float32',
    'sparse': False,
    'ragged': False,
    'name': 'conv2d_input'},
   'registered_name': None},
  {'module': 'keras.layers',
   'class_name': 'Conv2D',
   'config': {'name': 'conv2d',
    'trainable': True,
    'dtype': 'float32',
    'batch_input_shape': (None, 224, 224, 3),
    'filters': 32,
    'kernel_size': (3, 3),
    'strides': (1, 1),
    'padding': 'valid',
    'data_format': 'channels_last',
    'dilation_rate': (1, 1),
    'groups': 1,
    'activation': 'relu',
    'use_bias': True,
    'kernel_initializer': {'module': 'keras.initializers',
     'class_name': 'GlorotUniform',
     'config': {'seed': None},
     'registered_name': None},
    'bias_initializer': {'module': 'keras.initializers',
     'class_name': 'Zeros',
     'config': {},
     'registered_name': None},
    'kernel_regularizer': None,
    'bias_regularizer': None,
    'activity_regularizer': None,
    'kernel_constraint': None,
    'bias_constraint': None},
   'registered_name': None,
   'build_config': {'input_shape': (None, 224, 224, 3)}},
  {'module': 'keras.layers',
   'class_name': 'Conv2D',
   'config': {'name': 'conv2d_1',
    'trainable': True,
    'dtype': 'float32',
    'filters': 32,
    'kernel_size': (3, 3),
    'strides': (1, 1),
    'padding': 'valid',
    'data_format': 'channels_last',
    'dilation_rate': (1, 1),
    'groups': 1,
    'activation': 'relu',
    'use_bias': True,
    'kernel_initializer': {'module': 'keras.initializers',
     'class_name': 'GlorotUniform',
     'config': {'seed': None},
```

```
     'registered_name': None},
    'bias_initializer': {'module': 'keras.initializers',
     'class_name': 'Zeros',
     'config': {},
     'registered_name': None},
    'kernel_regularizer': None,
    'bias_regularizer': None,
    'activity_regularizer': None,
    'kernel_constraint': None,
    'bias_constraint': None},
   'registered_name': None,
   'build_config': {'input_shape': (None, 222, 222, 32)}},
  {'module': 'keras.layers',
   'class_name': 'MaxPooling2D',
   'config': {'name': 'max_pooling2d',
    'trainable': True,
    'dtype': 'float32',
    'pool_size': (2, 2),
    'padding': 'valid',
    'strides': (2, 2),
    'data_format': 'channels_last'},
   'registered_name': None,
   'build_config': {'input_shape': (None, 220, 220, 32)}},
  {'module': 'keras.layers',
   'class_name': 'Dropout',
   'config': {'name': 'dropout',
    'trainable': True,
    'dtype': 'float32',
    'rate': 0.5,
    'noise_shape': None,
    'seed': None},
   'registered_name': None,
   'build_config': {'input_shape': (None, 110, 110, 32)}},
  {'module': 'keras.layers',
   'class_name': 'Conv2D',
   'config': {'name': 'conv2d_2',
    'trainable': True,
    'dtype': 'float32',
    'filters': 64,
    'kernel_size': (3, 3),
    'strides': (1, 1),
    'padding': 'valid',
    'data_format': 'channels_last',
    'dilation_rate': (1, 1),
    'groups': 1,
    'activation': 'relu',
    'use_bias': True,
    'kernel_initializer': {'module': 'keras.initializers',
     'class_name': 'GlorotUniform',
     'config': {'seed': None},
     'registered_name': None},
    'bias_initializer': {'module': 'keras.initializers',
     'class_name': 'Zeros',
     'config': {},
     'registered_name': None},
    'kernel_regularizer': None,
    'bias_regularizer': None,
    'activity_regularizer': None,
    'kernel_constraint': None,
    'bias_constraint': None},
   'registered_name': None,
   'build_config': {'input_shape': (None, 110, 110, 32)}},
  {'module': 'keras.layers',
```

```
 'class_name': 'Conv2D',
 'config': {'name': 'conv2d_3',
  'trainable': True,
  'dtype': 'float32',
  'filters': 64,
  'kernel_size': (3, 3),
  'strides': (1, 1),
  'padding': 'valid',
  'data_format': 'channels_last',
  'dilation_rate': (1, 1),
  'groups': 1,
  'activation': 'relu',
  'use_bias': True,
  'kernel_initializer': {'module': 'keras.initializers',
   'class_name': 'GlorotUniform',
   'config': {'seed': None},
   'registered_name': None},
  'bias_initializer': {'module': 'keras.initializers',
   'class_name': 'Zeros',
   'config': {},
   'registered_name': None},
  'kernel_regularizer': None,
  'bias_regularizer': None,
  'activity_regularizer': None,
  'kernel_constraint': None,
  'bias_constraint': None},
 'registered_name': None,
 'build_config': {'input_shape': (None, 108, 108, 64)}},
{'module': 'keras.layers',
 'class_name': 'MaxPooling2D',
 'config': {'name': 'max_pooling2d_1',
  'trainable': True,
  'dtype': 'float32',
  'pool_size': (2, 2),
  'padding': 'valid',
  'strides': (2, 2),
  'data_format': 'channels_last'},
 'registered_name': None,
 'build_config': {'input_shape': (None, 106, 106, 64)}},
{'module': 'keras.layers',
 'class_name': 'Dropout',
 'config': {'name': 'dropout_1',
  'trainable': True,
  'dtype': 'float32',
  'rate': 0.5,
  'noise_shape': None,
  'seed': None},
 'registered_name': None,
 'build_config': {'input_shape': (None, 53, 53, 64)}},
{'module': 'keras.layers',
 'class_name': 'Flatten',
 'config': {'name': 'flatten',
  'trainable': True,
  'dtype': 'float32',
  'data_format': 'channels_last'},
 'registered_name': None,
 'build_config': {'input_shape': (None, 53, 53, 64)}},
{'module': 'keras.layers',
 'class_name': 'Dense',
 'config': {'name': 'dense',
  'trainable': True,
  'dtype': 'float32',
  'units': 64,
```

```
   'activation': 'relu',
   'use_bias': True,
   'kernel_initializer': {'module': 'keras.initializers',
    'class_name': 'GlorotUniform',
    'config': {'seed': None},
    'registered_name': None},
   'bias_initializer': {'module': 'keras.initializers',
    'class_name': 'Zeros',
    'config': {},
    'registered_name': None},
   'kernel_regularizer': None,
   'bias_regularizer': None,
   'activity_regularizer': None,
   'kernel_constraint': None,
   'bias_constraint': None},
  'registered_name': None,
  'build_config': {'input_shape': (None, 179776)}},
 {'module': 'keras.layers',
  'class_name': 'Dropout',
  'config': {'name': 'dropout_2',
   'trainable': True,
   'dtype': 'float32',
   'rate': 0.5,
   'noise_shape': None,
   'seed': None},
  'registered_name': None,
  'build_config': {'input_shape': (None, 64)}},
 {'module': 'keras.layers',
  'class_name': 'Dense',
  'config': {'name': 'dense_1',
   'trainable': True,
   'dtype': 'float32',
   'units': 3,
   'activation': 'softmax',
   'use_bias': True,
   'kernel_initializer': {'module': 'keras.initializers',
    'class_name': 'GlorotUniform',
    'config': {'seed': None},
    'registered_name': None},
   'bias_initializer': {'module': 'keras.initializers',
    'class_name': 'Zeros',
    'config': {},
    'registered_name': None},
   'kernel_regularizer': None,
   'bias_regularizer': None,
   'activity_regularizer': None,
   'kernel_constraint': None,
   'bias_constraint': None},
  'registered_name': None,
  'build_config': {'input_shape': (None, 64)}}]}
```


In [28]:
```
model.layers[0].get_config()
```

Out[28]:

```
{'name': 'conv2d',
 'trainable': True,
 'dtype': 'float32',
 'batch_input_shape': (None, 224, 224, 3),
 'filters': 32,
 'kernel_size': (3, 3),
 'strides': (1, 1),
 'padding': 'valid',
 'data_format': 'channels_last',
 'dilation_rate': (1, 1),
 'groups': 1,
```

```
 'activation': 'relu',
 'use_bias': True,
 'kernel_initializer': {'module': 'keras.initializers',
  'class_name': 'GlorotUniform',
  'config': {'seed': None},
  'registered_name': None},
 'bias_initializer': {'module': 'keras.initializers',
  'class_name': 'Zeros',
  'config': {},
  'registered_name': None},
 'kernel_regularizer': None,
 'bias_regularizer': None,
 'activity_regularizer': None,
 'kernel_constraint': None,
 'bias_constraint': None}
```

In [29]:
```
model.layers[0].input_shape
```

Out[29]:
```
(None, 224, 224, 3)
```

In [30]:
```
model.layers[0].output_shape
```

Out[30]:
```
(None, 222, 222, 32)
```

In [31]:
```
model.layers[0].get_weights()
```

Out[31]:
```
[array([[[[ 1.30819187e-01,   1.22043490e-03, -7.46863037e-02,
            9.64652896e-02,  5.06858677e-02,  9.42830741e-02,
          1.82143748e-02, -3.86717319e-02,  2.76170373e-02,
            1.27847418e-01, -8.99328440e-02,   6.06965572e-02,
         -9.54713002e-02,  5.76431453e-02, -1.20869726e-01,
            6.62304014e-02,  2.63653696e-03,  8.78323019e-02,
         8.35230947e-03,  1.22183457e-01, -1.12507775e-01,           -
         4.37350422e-02,  2.69440711e-02,  3.98484766e-02,
           -1.00158662e-01, -3.20490673e-02, -1.21472083e-01,
            1.11380905e-01,   1.23215809e-01,   4.27543819e-02,
         1.28189489e-01, -6.70007020e-02],
          [ 1.33672968e-01, -6.09749332e-02,   1.26069829e-01,
         -9.94340032e-02,        4.88449633e-03,        1.07527941e-01,
         1.07485205e-02,  5.33378422e-02,  1.21159568e-01,           -
         1.08382322e-01, -1.30748570e-01,  3.52613032e-02,
           -3.37651521e-02, -3.42539549e-02,  7.18763918e-02,
           -7.27965757e-02,  7.77958781e-02,  8.65470618e-02,
           -1.13600381e-01,  -1.24182761e-01,  -7.77512789e-03,
         -1.23596638e-02,  6.76263273e-02,  7.70800263e-02,
           -1.71595067e-02,  8.35649222e-02,  5.80494255e-02,
            3.73375714e-02,   8.53247494e-02,   5.26038259e-02,
         5.90997487e-02, -9.32641700e-02],
          [-8.48410800e-02,   9.02998000e-02, -2.12717056e-03,
         -1.35182992e-01,       -3.89805809e-02,       -9.70700830e-02,
         7.90845305e-02, -1.03053078e-01, -5.40456176e-03,           -
         1.18464351e-01, -2.40975022e-02,  1.15814075e-01,           -
         1.16489165e-01,        1.00124031e-02,        2.00421661e-02,
         7.72698224e-03, -1.40529871e-02, -1.07965715e-01,           -
         9.31869745e-02, -9.42342728e-02, -1.15124829e-01,
           -9.94184613e-03,  1.67474002e-02, -1.35891959e-02,
            1.46867037e-02, -3.10033560e-04, -1.07543908e-01,
            9.23612863e-02, -8.57572854e-02, -9.98024717e-02,
           -7.30097294e-02,  7.82865286e-02]],

         [[-5.37678599e-03,  6.08397275e-02, -3.26765329e-02,
            2.09687650e-02,  1.02859989e-01,  4.05950248e-02,
            1.07155859e-01,   6.42398894e-02,  -1.35938004e-02,
         -5.94774336e-02,  9.73174274e-02,  9.99692380e-02,
            8.14161450e-02,  1.98506266e-02, -4.16404828e-02,
          1.02249444e-01,  7.86580592e-02, -1.32020667e-01,
```

```
            8.59857053e-02,   1.02153465e-01,   2.29799151e-02,
-5.84039092e-02, -7.31234848e-02,  6.52524978e-02,
            9.55528021e-03,   5.94959855e-02,   1.27431706e-01,
-5.28491661e-02,     -1.31900698e-01,          7.81397372e-02,
3.70036811e-02, -9.53932106e-03],
          [-1.27972901e-01, -4.24610972e-02, -8.02764446e-02,
            1.11551881e-01,   3.24679464e-02, -8.16035345e-02,
-4.06835973e-03,          1.35660812e-01,     -3.46463472e-02,
8.35370868e-02, -7.04363063e-02, -1.35932133e-01,            -
7.38175139e-02, -7.73624554e-02, -1.06234126e-01,            -
6.14659041e-02,  1.31410345e-01, -2.57683471e-02,
            1.36102989e-01,  1.26619145e-01, -6.09185323e-02,
            8.56596231e-03,   7.11592585e-02,   7.08939731e-02,
-2.20364407e-02,  5.53735942e-02,  6.36639148e-02,
           -9.36392844e-02, -1.14142716e-01, -4.57812846e-03,
           -7.44935125e-02, -1.17317878e-01],
          [ 8.30929428e-02,   1.97455287e-03, -2.75602490e-02,
4.95726019e-02,  1.37953237e-01, -1.23242564e-01,            -
5.44604883e-02,        -2.28779167e-02,        -5.10753021e-02,
3.64711285e-02, -7.95621127e-02,  1.61508024e-02,            -
1.35315672e-01,  1.32620335e-05,  2.41552144e-02,
      -3.84518653e-02,  4.55383658e-02,  1.36576518e-01,            -
       3.26694548e-02,  1.14940107e-03, -1.24851614e-01,
            3.56513560e-02, -3.30646783e-02, -6.56478554e-02,
            1.27239093e-01,   1.11678332e-01,   6.95177615e-02,
-5.15577197e-03,  1.09116197e-01,  2.39838213e-02,
            6.84559345e-03, -5.10090292e-02]],

         [[ 8.50358754e-02, -9.78583843e-02, -9.44300592e-02,
            6.58917427e-02,  5.49657047e-02,  6.81413859e-02,
1.12296745e-01, -2.50239372e-02,  1.37898907e-01,            -
7.64615536e-02,  1.10433906e-01,  2.38248855e-02,
           -4.89845872e-02,  7.55269229e-02,  1.09704882e-02,
           -1.07556283e-02,  -4.60848808e-02,  -9.66913849e-02,
-5.52865639e-02,          2.97851861e-03,      -6.15463555e-02,
6.59817010e-02,  2.39406377e-02,  1.40332431e-02,            -
1.35644093e-01,  3.32365930e-03,  6.81077838e-02,
            7.90141523e-03,   7.14982152e-02,  -6.23375922e-02,
1.21363863e-01, -1.27166927e-01],
          [-1.20702475e-01,  -5.41642308e-04,   1.35911703e-02,
-7.95833766e-03, -9.78359729e-02, -6.59059957e-02,
            6.40614778e-02,  5.62935472e-02, -1.33384511e-01,
3.29811275e-02, -9.40106511e-02,  4.46021557e-04,            -
4.72603068e-02,  8.84273797e-02, -2.98490152e-02,
           -1.26761347e-01,  8.42005908e-02, -1.01455085e-01,
           -1.02850258e-01,   2.34548301e-02,  -4.66170162e-02,
-1.18522495e-02, -1.01882093e-01, -5.26607037e-04,
           -2.40594000e-02,  6.45424128e-02, -1.28963411e-01,
           -6.86733574e-02,   1.06841847e-01,  -3.91412526e-02,
1.08504340e-01, -3.11708450e-03],
          [ 1.02367207e-01,  -3.39139476e-02,  -1.05001539e-01,
-1.16228297e-01, -2.51945853e-02,  2.88722366e-02,
           -1.11115038e-01,  -1.44730508e-02,   1.28280923e-01,
1.04357064e-01, -1.00739539e-01, -1.38555244e-02,            -
1.36477172e-01,           4.60929424e-02,      -1.33480653e-02,
1.23500749e-01,  1.16238445e-02,  2.51748711e-02,            -
1.30668908e-01,  1.12245455e-01,  5.01802266e-02,
            6.02132827e-02,  1.11756474e-02,  2.83515453e-03,
1.40876025e-02,  1.10789999e-01, -4.38624471e-02,            -
1.35977522e-01,  1.23252317e-01,  4.02958840e-02,
            1.36501893e-01, -8.24709386e-02]]],
```

```
       [[[ 8.33144784e-03, -7.53190964e-02,  4.32455838e-02,
-1.21371984e-01,  1.28943428e-01,  1.25752345e-01,
     -2.70372257e-02, -1.84454620e-02, -8.57987702e-02,           -
       4.34091836e-02, -7.87398815e-02, -1.34969369e-01,
            7.17023611e-02, -1.43782198e-02, -7.00893998e-03,
            1.31089285e-01,  -1.17380917e-02,   1.19927868e-01,
-1.17476411e-01, -3.58280316e-02,  4.92153168e-02,           -
6.73083961e-02,      -2.45267153e-02,           5.86271286e-02,
5.28918505e-02, -1.19348861e-01,  4.61010635e-02,            -
4.08916622e-02, -6.73450530e-02, -4.12170887e-02,
          -1.31915972e-01,  1.37957141e-01],
         [-3.83659527e-02, -5.17656729e-02, -4.04730812e-02,
           9.00882185e-02,   2.42765248e-03,   1.23901084e-01,
8.25500786e-02, -1.16028637e-02, -6.05915934e-02,
           6.21235818e-02,  1.06382757e-01, -1.24255776e-01,
-6.48259521e-02,  6.57276809e-02,  8.00132751e-02,
1.16062984e-01, -6.95421472e-02,  9.49426889e-02,           -
5.30944392e-02,  4.11695391e-02,  2.72054076e-02,
          -7.39001036e-02, -2.85670385e-02,  6.74469620e-02,
           7.41897374e-02,  -8.49346966e-02,  -2.91728526e-02,
-5.24559841e-02,        3.33524942e-02,        5.41649014e-02,
6.95945024e-02, -5.14476150e-02],
         [-1.25965148e-01, -5.94906509e-03, -1.31894276e-01,
           3.68194580e-02,  -1.91002041e-02,   1.08492881e-01,
-2.12569535e-02, -1.28346831e-02, -4.87565249e-02,           -
1.34887442e-01, -1.18832737e-02,  1.31267861e-01,
           6.17816895e-02,   5.51704168e-02,  -1.02370799e-01,
5.74259162e-02, -6.91189542e-02,  5.32531738e-02,            -
1.15171656e-01, -1.12094156e-01, -8.08677152e-02,
          -5.74301854e-02,  3.70087475e-02, -9.59591568e-03,
          -3.52002010e-02, -2.28073001e-02, -1.27249718e-01,
          -1.19077072e-01,   3.10950875e-02,  -2.55053043e-02,
-9.31710154e-02, -5.41308820e-02]],

        [[ 3.86233628e-03,  9.75410491e-02, -8.17558840e-02,
          -2.43505388e-02, -9.10353959e-02, -2.13001221e-02,
          -9.29766819e-02, -7.22346231e-02, -9.80517343e-02,
          -6.84883669e-02,  -1.08946905e-01,   1.14139810e-01,
1.11295000e-01,  3.27976197e-02, -7.02884272e-02,            -
1.70773715e-02, -1.18919551e-01, -1.85481608e-02,
          -3.11844572e-02, -4.41456288e-02,  1.54016316e-02,
           2.16643214e-02, -6.98965341e-02, -4.98455465e-02,
           2.31613219e-02,  1.16773471e-01,  1.00822464e-01,
          -1.04886606e-01,   3.56287956e-02,   8.31734985e-02,
7.63450414e-02, -1.37875453e-02],
         [-6.67971596e-02, -3.57084200e-02, -9.89135504e-02,
-7.37003386e-02, -6.64416552e-02, -1.30669102e-01,           -
5.55152893e-02, -2.15095282e-03, -1.15423046e-01,
           5.44280410e-02,  -2.39495933e-02,  -7.72581846e-02,
5.36980480e-02,      -1.23922743e-01,           9.89472419e-02,
1.04657784e-01,         6.90995753e-02,         9.84918028e-02,
8.63605887e-02,  4.88334596e-02,  2.48384178e-02,
           1.10884935e-01,   1.01799935e-01,  -8.89457613e-02,
-6.73575625e-02,  1.25060678e-02,  8.47292393e-02,
           1.01295799e-01,  -6.33146688e-02,   5.00430465e-02,
1.03451729e-01, -6.84394091e-02],
         [-6.12630174e-02,   1.31791458e-01,   1.17298946e-01,
-1.28734723e-01, -5.71272969e-02,  1.68211162e-02,
          -1.17480658e-01, -7.18560889e-02, -3.82867157e-02,
           1.19630143e-01,   7.60036707e-03,   7.60897994e-04,
-1.01738960e-01, -1.92440972e-02, -4.08100933e-02,
           1.11065447e-01,  -1.04460686e-01,   3.35304737e-02,
9.80753601e-02,        -9.40033495e-02,        -9.17088240e-02,
1.09560356e-01, -7.70313293e-02,  1.29158899e-01,
```

```
                1.29645094e-01,   3.21409404e-02, -8.09575468e-02,
       -4.80965823e-02,  1.78153813e-03, -1.04954749e-01,
                4.68791723e-02, -3.23190168e-02]],

              [[-8.37074667e-02,   3.49629521e-02, -6.13553748e-02,
       -3.56505737e-02, -7.45745227e-02,  1.12140134e-01,
               -1.16531774e-01, -1.16942599e-01,  7.00550377e-02,
                3.97466123e-02, -6.45633414e-02, -1.83123276e-02,
                1.08811557e-02, -5.66418543e-02,   6.43687695e-02,
       -1.04365394e-01, -1.04231596e-01,  6.16329908e-02,
               -1.11175582e-01, -2.11159885e-03,  1.02617949e-01,
                3.60996574e-02,   9.02796388e-02, -3.69216502e-03,
       -6.00083843e-02, -6.54212385e-02, -1.14379570e-01,
                1.19219109e-01, -8.99784863e-02,   8.84356350e-02,
       3.48278880e-02,  3.87982875e-02],
              [-7.90220797e-02,  1.36925295e-01,  1.34403661e-01,
       1.34074643e-01, -8.01123828e-02,  3.02472264e-02,           -
       9.71472114e-02,  1.04495242e-01,  3.56927961e-02,
            -5.50083518e-02,  9.62279737e-03, -5.54749370e-03,           -
             1.28506720e-02,  1.06068447e-01, -1.35102972e-01,
                1.26301095e-01, -6.89555183e-02, -2.15728432e-02,
                8.17199051e-03,   1.15445599e-01,   6.67931139e-02,
       -6.47784397e-02, -4.86363247e-02, -1.10276259e-01,
                6.54414743e-02, -6.08585775e-03,  1.17242143e-01,
                1.24757782e-01,  1.36911288e-01, -5.41399941e-02,
                4.79549021e-02, -2.55420581e-02],
              [ 8.92922580e-02,  7.69195259e-02, -4.62576747e-03,
                1.06217340e-01, -9.00135934e-02,  3.48832905e-02,
       8.93289745e-02,  3.14858258e-02, -5.48993424e-02,            6.79262877e-02,
                   -8.56303349e-02,  5.74491471e-02,
                9.75201875e-02,  -9.09922868e-02,  -2.47511864e-02,
       -6.66643158e-02, -1.22798778e-01, -2.32971907e-02,
               -1.22994527e-01, -1.26594976e-01, -9.44965929e-02,
                9.22931433e-02,  1.28762886e-01, -4.44497690e-02,
                1.27703771e-01,  2.88475901e-02,  1.93018466e-02,
               -1.26518905e-02,  9.70387310e-02, -9.02553499e-02,
                9.99018699e-02,  3.09829414e-03]]],


             [[[-2.84674019e-02, -7.03926757e-02,  6.85966909e-02,
                1.07938662e-01, -1.01399705e-01, -8.46358165e-02,
                1.05795503e-01,   8.29964876e-04, -1.28323048e-01,
       -9.40097943e-02,  2.10227966e-02, -2.37013921e-02,
                6.04754686e-02,   9.66637433e-02, -5.55374026e-02,
       -4.53710034e-02,  1.07127264e-01, -2.30444446e-02,
               -1.21845849e-01,  1.11670807e-01,  1.31758526e-01,
               -2.26723552e-02, -1.01113170e-01,   8.97675753e-04,
       -3.22848633e-02, -7.89360330e-02, -3.54686752e-02,
               -1.00068733e-01, -6.76346198e-02,   1.13459274e-01,
       1.21519715e-02, -3.43387201e-02],
              [-4.94593382e-03, -3.82247195e-02, -6.65623471e-02,
                9.98582840e-02, -3.56005207e-02, -5.24656251e-02,
             2.62061805e-02, -1.19155645e-03, -1.04632944e-01,
                2.38979310e-02, -1.00589029e-01,   6.33828491e-02,
       -7.94185847e-02,  4.06016111e-02,  1.31346092e-01,
                6.86239302e-02, -6.58497587e-02,  2.89816707e-02,
       4.97007370e-02,  6.45098388e-02,  2.50271559e-02,            1.33060887e-01,
        -1.26560196e-01,  7.65852481e-02,            8.82069170e-02,  2.58898735e-
                                03,  1.93957537e-02,
                2.69680917e-02,   5.61858863e-02, -2.63125300e-02,
       6.60366267e-02, -7.77476057e-02],
              [-1.25112385e-01, -4.07883450e-02,  1.12182945e-02,
                3.41919661e-02,   7.37981349e-02, -4.46839854e-02,
       8.30855370e-02, -3.26494128e-02, -1.03777543e-01,
```

```
                1.57117844e-02,  -6.97409287e-02,  -1.09386444e-03,
        -1.54286623e-04,         1.10398248e-01,         6.52337074e-02,
        5.25189042e-02, -2.30587274e-02, -1.89485103e-02,            -
        1.13122046e-01,  2.79443115e-02, -9.79645699e-02,
                -1.35541096e-01, -4.06207889e-02,  9.56631750e-02,
        7.75039494e-02,  1.36214346e-02,  1.01154998e-01,            -
        5.01533672e-02,  3.18863839e-02,  7.53821731e-02,
                1.55251473e-02,  8.32796097e-04]],

              [[-1.36137396e-01,   1.66207850e-02, -4.87460271e-02,
        -5.41725382e-02, -6.19836301e-02,  9.32126492e-02,
                3.37193906e-02,   1.15900621e-01, -6.70127124e-02,
        -1.12623602e-01,  4.50576544e-02, -3.99583578e-03,
                 2.21438408e-02,  4.33555096e-02, -6.64980188e-02,
    5.49696386e-03, -1.23869829e-01,  1.25903636e-02,            1.14461735e-01,
    7.93761760e-02, -3.17240655e-02,            9.14947391e-02,  1.73824728e-02,
                               -9.37475115e-02,
                9.59576368e-02,   5.40639907e-02,   3.95075828e-02,
        -7.01157749e-03,       -8.89982283e-03,       -1.23231970e-01,
        1.25155613e-01,  9.08647478e-02],
                [-7.57284015e-02,   6.46099746e-02,   7.26129562e-02,
        1.11082897e-01,  6.22446686e-02,  5.29126376e-02,            -
        4.56726104e-02, -2.78609991e-02, -1.05460107e-01,            -
        1.03258438e-01, -2.92267501e-02, -4.92113382e-02,
                4.50957566e-02,  1.31158218e-01, -4.57847491e-02,
                1.62467957e-02,  7.82876760e-02,  7.53044784e-02,
                1.16573587e-01,   8.56697857e-02,   2.88355350e-03,
        -8.09486657e-02,  2.90552080e-02, -1.85715184e-02,
                -5.75065911e-02,  1.06900707e-01,  8.35144520e-02,
                -1.24362454e-01, -3.91086489e-02,   4.48624790e-03,
        1.31689310e-02, -6.67003840e-02],
                [-4.55742255e-02, -6.06607571e-02,   4.11621034e-02,
        -1.54887885e-02, -2.73401439e-02,  3.36507112e-02,
                -1.02885790e-01,  9.87377614e-02,  1.28935829e-01,
                -7.78409913e-02, -1.16754316e-01, -1.34708866e-01,
                 4.96626198e-02,  9.08815563e-02,  4.87059504e-02,
              1.64322704e-02, -1.27197802e-03,  9.90230888e-02,
                1.21909156e-01,   1.19912341e-01,   9.75430310e-02,
        -2.97101513e-02,  4.67547476e-02,  5.54799736e-02,
                1.33366540e-01,   2.69794166e-02,   4.25442457e-02,
        -1.10234506e-01, -3.48773375e-02, -1.31131679e-01,
                3.94605249e-02,  9.81076062e-02]],

              [[  2.51200497e-02,  -9.36341211e-02,  -4.67206314e-02,
        -4.84213233e-02, -1.37962103e-01,  9.71088111e-02,
                -2.49306858e-03,  1.02378920e-01,  4.15574312e-02,
                -8.97044837e-02,  2.28849500e-02, -8.98732543e-02,
                 4.82911766e-02,   5.16867638e-02,  -2.36672685e-02,
        -2.83210427e-02,           1.02374345e-01,      -8.42012092e-02,
        1.06836975e-01, -9.02373791e-02,  9.55490470e-02,            -
        5.55169955e-02, -4.47488725e-02, -4.59655300e-02,            -
        2.35739872e-02, -8.87843966e-02, -6.12370223e-02,
                1.03283525e-02,   1.10865891e-01,  -4.44041193e-03,
        9.32968110e-02,  9.51604098e-02],
                [-8.48768651e-03,   3.98727953e-02,   4.08169627e-02,
        -6.41046464e-03,  3.61417383e-02, -2.22952366e-02,
                -1.34651244e-01,  1.11154258e-01,  8.96360129e-02,
        6.47932738e-02, -7.03260303e-03,  1.12902969e-02,            -
        3.26249003e-03, -1.21226519e-01,  5.60993850e-02,
                1.19097516e-01,  -7.12423399e-02,  -5.67770898e-02,
        -4.76877093e-02,      -2.87257135e-03,           5.34258634e-02,
        1.19040176e-01, -1.13292485e-02,  4.04617339e-02,            -
        1.82392448e-02, -8.60814899e-02, -3.59152928e-02,
```

```
           -1.19342253e-01, -6.22440055e-02,  5.48570305e-02,
        1.73793435e-02, -1.23483032e-01],
         [ 1.27635166e-01, -7.56311640e-02,  9.82901901e-02,
        -6.63041100e-02, -1.12987161e-02, -6.27360344e-02,           -
        1.52379274e-04,  7.98870772e-02,  7.55665749e-02,
           -4.70480397e-02,  1.34023651e-01,  7.25637078e-02,
           -4.29966226e-02, -5.07288128e-02,  7.67967850e-02,
        7.98142999e-02, -9.20170769e-02,  1.44783556e-02,           -
        1.12329334e-01, -6.77904561e-02,  1.00001186e-01,
           -9.77781266e-02,  5.82818240e-02,  4.04172689e-02,
            9.14824009e-02, -1.24828368e-02,  7.80842602e-02,
            4.70110774e-02, -1.37754902e-01, -2.65953094e-02,
           -5.36578074e-02, -6.76176101e-02]]]], dtype=float32),
         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.,  0.,  0.,  0.,  0.],
        dtype=float32)]
```

In [34]:
```
print(np.shape(model.layers[0].get_weights()[0]))
print(np.shape(model.layers[0].get_weights()[1]))
```

```
(3, 3, 3, 32)
(32,)
```

In [35]:
```
model.layers[0].trainable
```

```
True
```

Out[35]:

In [36]:
```
hist = model.fit(X_train, y_train, batch_size=16, epochs=num_epoch, verbose=1, vali
```

```
Epoch 1/2 8/8 [==============================] - 104s 7s/step - loss: 4.6745 -
accuracy: 0.3
667 - val_loss: 1.1070 - val_accuracy: 0.2667
Epoch 2/2 8/8 [==============================] - 27s 3s/step - loss: 1.0851 - accuracy:
0.36
67 - val_loss: 1.0996 - val_accuracy: 0.3000
```

In [37]:
```
score = model.evaluate(X_test, y_test, batch_size=16)
print('Test Loss:', score[0]) print('Test Accuracy:',
score[1])
```

```
2/2 [==============================] - 1s 569ms/step - loss: 1.0996 - accuracy: 0.
3000
Test Loss: 1.099566102027893
Test Accuracy: 0.30000001192092896
```

In [38]:
```
test_image = X_test[0:1]
print (test_image.shape)
```

```
(1, 224, 224, 3)
```

In [40]:
```
print(model.predict(test_image))
print(y_test[0:1])
```

```
1/1 [==============================] - 0s 284ms/step
[[0.32515705 0.33962205 0.33522096]]
[[1. 0. 0.]]
```

In [41]:
```
from sklearn.metrics import confusion_matrix
```

```
Y_pred = model.predict(X_test)
print(Y_pred)
```

```
1/1 [==============================] - 2s 2s/step
[[0.32515705 0.33962208 0.33522096]
 [0.32575732 0.3389225  0.33532017]
```

```
 [0.32684988 0.33662334 0.33652678]
 [0.32487446 0.33956906 0.33555648]
 [0.32396433 0.34011862 0.3359171 ]  [0.32830226
0.33522165 0.33647612]
 [0.32608843 0.33837917 0.3355324 ]
 [0.32512024 0.33929604 0.33558375]
 [0.32410952 0.3395924  0.3362981 ]  [0.32851028
0.33478174 0.33670795]
 [0.32334495 0.34220192 0.33445305]
 [0.32566318 0.337592   0.33674482]
 [0.32820114 0.33640984 0.33538905]  [0.32399088
0.3405988  0.33541036]
 [0.32456142 0.34017688 0.33526167]
 [0.3271584  0.33748674 0.3353548 ]
 [0.32787985 0.3359873  0.33613288]  [0.32551143
0.34032318 0.3341654 ]
 [0.32363418 0.339866   0.33649987]
 [0.32096425 0.34418055 0.33485523]
 [0.3250435  0.33941478 0.33554173]  [0.31945288
0.35371897 0.32682812]
 [0.3276528  0.33677667 0.33557048]
 [0.32068586 0.3432814  0.33603275]
 [0.32110372 0.34355694 0.3353393 ]  [0.32937962
0.33416253 0.33645782]
 [0.3257028  0.3385384  0.33575878]
 [0.32405204 0.33967412 0.33627385]
 [0.32425565 0.34146863 0.3342757 ]
 [0.3251048  0.340002   0.3348932 ]]
```

In [42]:
```
y_pred = np.argmax(Y_pred, axis=1)
print(y_pred)
```

```
[1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1]
```

In [43]:
```
print(confusion_matrix(np.argmax(y_test,axis=1), y_pred))
```

```
[[ 0 11  1]
 [ 0  7  1]
 [ 0  8  2]]
```

In [ ]: