Face Musk Detection Using OpenCV

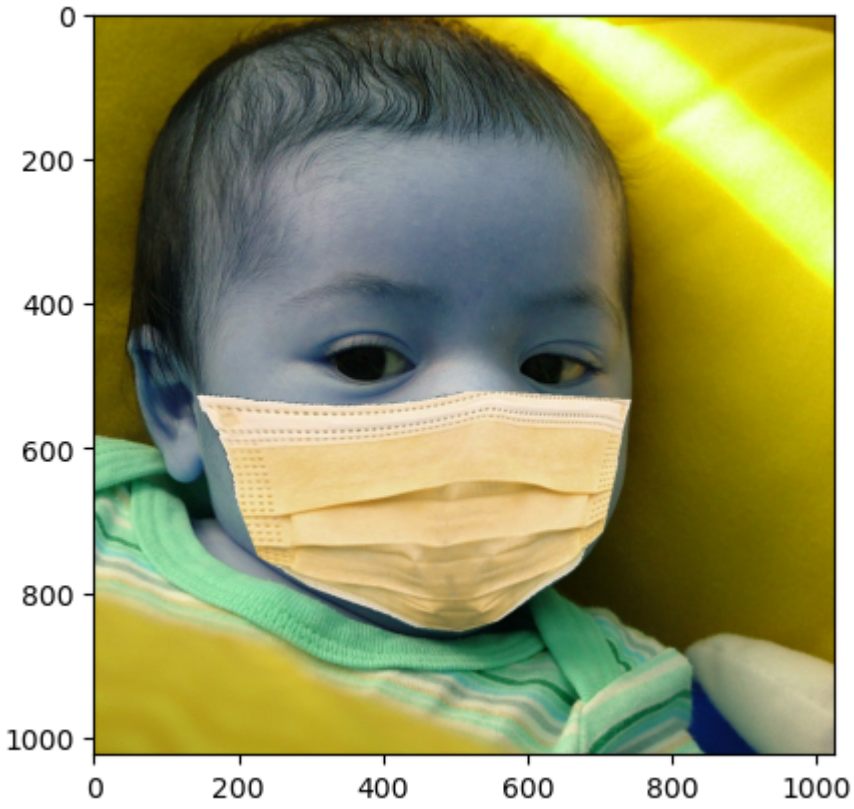
Import Librarys

In [1]:
```
import tensorflow as tf

import cv2
import os

import matplotlib.pyplot as plt

import numpy as np
```

In [2]:
```
#BGR because cv2 always uses BGR
img_array = cv2.imread("Dataset/With_Mask/00000_Mask.jpg")
```

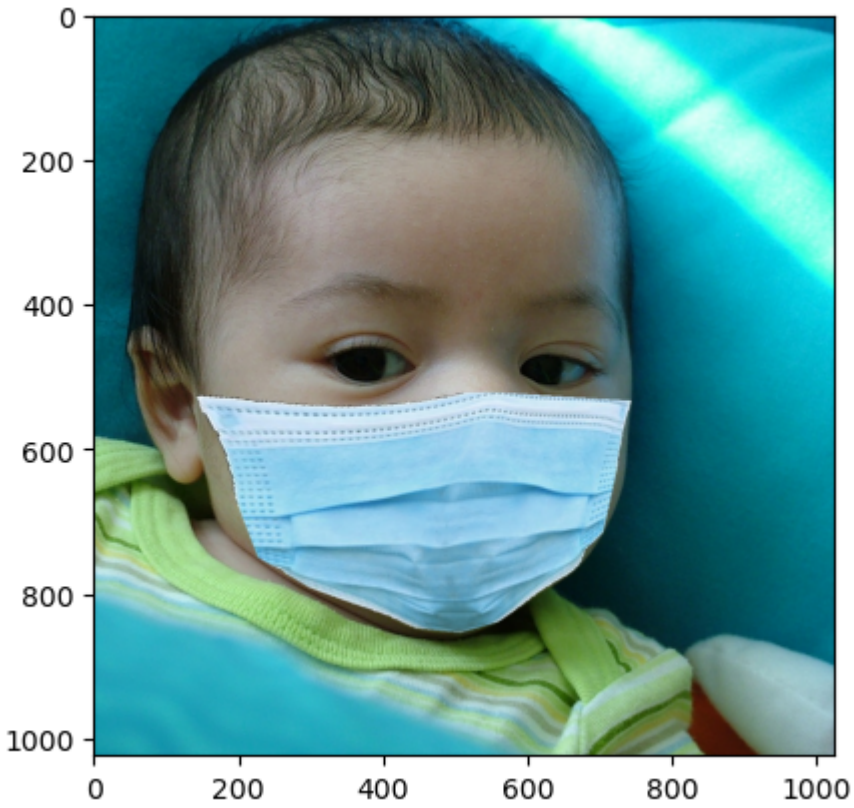
In [3]:
```
plt.imshow(img_array)
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x1f6e23a3460>
```



In [4]:
```
plt.imshow(cv2.cvtColor(img_array, cv2.COLOR_BGR2RGB))
```

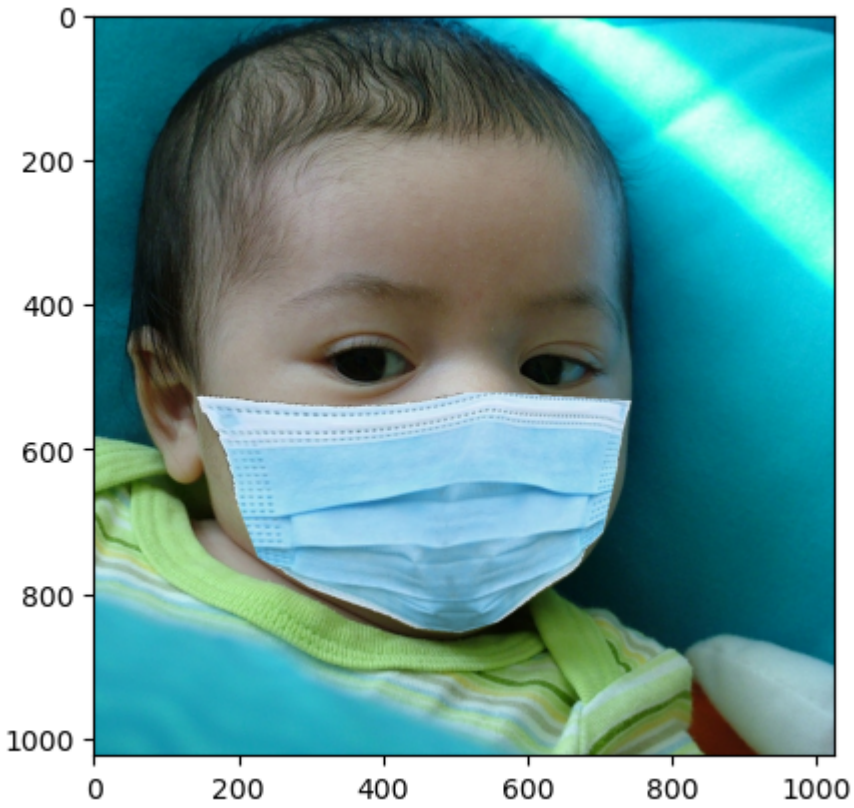
Out[4]:
```
<matplotlib.image.AxesImage at 0x1f6e24ed460>
```



In [5]:
```
img_array.shape
```

Out[5]:
```
(1024, 1024, 3)
```

In [6]:
```
Datadirectory = "Dataset/"#tranning data
Classes = ["With_Mask","Without_Mask"]#list of classes
for category in Classes:
    path = os.path.join(Datadirectory,category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img))
        plt.imshow(cv2.cvtColor(img_array, cv2.COLOR_BGR2RGB))
        plt.show()
        break
    break
```



In [7]:
```
#Imagenet 224*224
img_size = 224

new_array = cv2.resize(img_array, (img_size,img_size))
plt.imshow(cv2.cvtColor(new_array, cv2.COLOR_BGR2RGB))
plt.show()
```

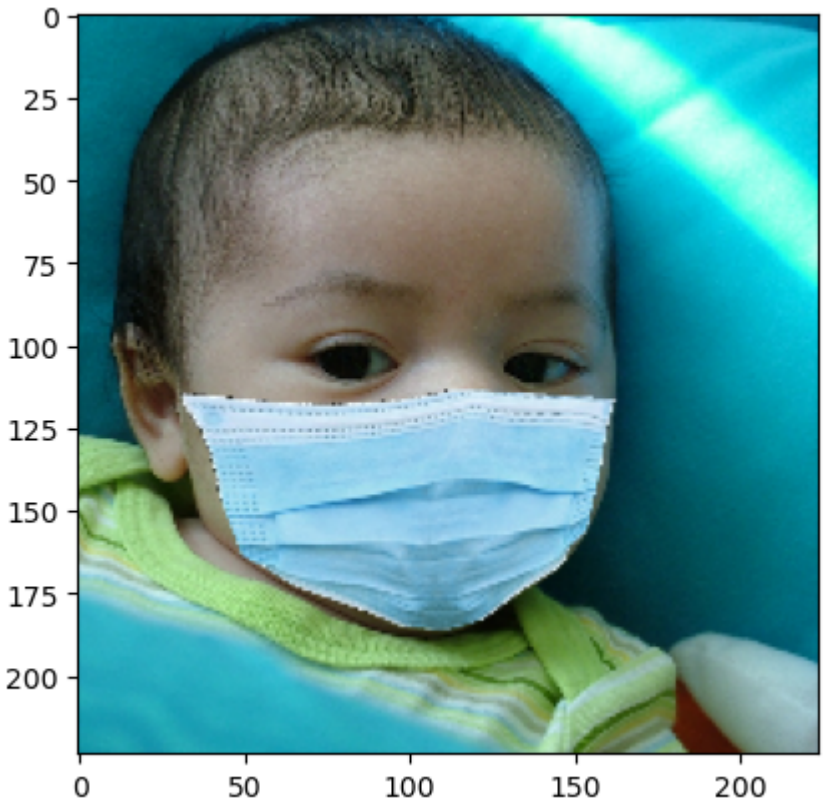


Image reading and converting all to array

In [8]:
```
training_data = []

def create_training_data():
    for category in Classes:
        path = os.path.join(Datadirectory, category)
        class_num = Classes.index(category)
        for img in os.listdir(path):
            try:
                img_array = cv2.imread(os.path.join(path,img))
                new_array = cv2.resize(img_array,(img_size,img_size))
                training_data.append([new_array,class_num])
            except Exception as e:
                pass
```

In [9]:
```
create_training_data()
```

In [10]:
```
print(len(training_data))
```

2878

In [11]:
```
import random

random.shuffle(training_data)
```

In [12]:
```
x =[]
y =[]
```

```
for features, label in training_data:
    x.append(features)
    y.append(label)

x = np.array(x).reshape(-1, img_size, img_size, 3)
```

In [13]:
```
x.shape
```

Out[13]:
```
(2878, 224, 224, 3)
```

In [14]:
```
#normalizing the data
x =x/255.0;
```

In [15]:
```
y[1000]
```

Out[15]:
```
0
```

In [16]:
```
y = np.array(y)
```

In [17]:
```
import pickle

pickle_out = open("x.pickle","wb")
pickle.dump(x, pickle_out)
pickle_out.close()

pickle_out = open("y.pickle","wb")
pickle.dump(y, pickle_out)
pickle_out.close()
```

In [18]:
```
pickle_in = open("x.pickle","rb")
x = pickle.load(pickle_in)

pickle_in = open("y.pickle","rb")
y = pickle.load(pickle_in)
```

Model Tranning

In [19]:
```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
```

In [20]:
```
#pre-trained model from internet with 4.6m parameter
model = tf.keras.applications.mobilenet.MobileNet()
```

In [21]:
```
model.summary()
```

```
Model: "mobilenet_1.00_224"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 input_1 (InputLayer)         [(None, 224, 224, 3)]     0

 conv1 (Conv2D)               (None, 112, 112, 32)      864

 conv1_bn (BatchNormalizatio  (None, 112, 112, 32)     128
 n)

 conv1_relu (ReLU)            (None, 112, 112, 32)      0

 conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)     288

 conv_dw_1_bn (BatchNormaliz  (None, 112, 112, 32)     128
 ation)

 conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0

 conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048

 conv_pw_1_bn (BatchNormaliz  (None, 112, 112, 64)     256
 ation)

 conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0

 conv_pad_2 (ZeroPadding2D)  (None, 113, 113, 64)      0

 conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)       576

 conv_dw_2_bn (BatchNormaliz  (None, 56, 56, 64)       256
 ation)

 conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0

 conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192

 conv_pw_2_bn (BatchNormaliz  (None, 56, 56, 128)      512
 ation)

 conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0

 conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)      1152

 conv_dw_3_bn (BatchNormaliz  (None, 56, 56, 128)      512
 ation)

 conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0

 conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384

 conv_pw_3_bn (BatchNormaliz  (None, 56, 56, 128)      512
 ation)

                 (ReLU)       (None, 56, 56, 128)       0
```

```
conv_pad_4 (ZeroPadding2D)  (None, 57, 57, 128)       0

conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)      1152

conv_dw_4_bn (BatchNormaliz  (None, 28, 28, 128)      512
ation)

conv_dw_4_relu (ReLU)       (None, 28, 28, 128)       0

conv_pw_4 (Conv2D)          (None, 28, 28, 256)       32768

conv_pw_4_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)

conv_pw_4_relu (ReLU)       (None, 28, 28, 256)       0

conv_dw_5 (DepthwiseConv2D)  (None, 28, 28, 256)      2304

conv_dw_5_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)

conv_dw_5_relu (ReLU)       (None, 28, 28, 256)       0

conv_pw_5 (Conv2D)          (None, 28, 28, 256)       65536

conv_pw_5_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)

conv_pw_5_relu (ReLU)       (None, 28, 28, 256)       0

conv_pad_6 (ZeroPadding2D)  (None, 29, 29, 256)       0

conv_dw_6 (DepthwiseConv2D)  (None, 14, 14, 256)      2304

conv_dw_6_bn (BatchNormaliz  (None, 14, 14, 256)      1024
ation)

conv_dw_6_relu (ReLU)       (None, 14, 14, 256)       0

conv_pw_6 (Conv2D)          (None, 14, 14, 512)       131072

conv_pw_6_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_6_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_7 (DepthwiseConv2D)  (None, 14, 14, 512)      4608

conv_dw_7_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_dw_7_relu (ReLU)       (None, 14, 14, 512)       0

conv_pw_7 (Conv2D)          (None, 14, 14, 512)       262144
```

```
conv_pw_7_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_7_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_8 (DepthwiseConv2D)  (None, 14, 14, 512)      4608

conv_dw_8_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_dw_8_relu (ReLU)       (None, 14, 14, 512)       0

conv_pw_8 (Conv2D)          (None, 14, 14, 512)       262144

conv_pw_8_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_8_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_9 (DepthwiseConv2D)  (None, 14, 14, 512)      4608

conv_dw_9_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_dw_9_relu (ReLU)       (None, 14, 14, 512)       0

conv_pw_9 (Conv2D)          (None, 14, 14, 512)       262144

conv_pw_9_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_9_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_10 (DepthwiseConv2D  (None, 14, 14, 512)      4608
)

conv_dw_10_bn (BatchNormali  (None, 14, 14, 512)      2048
zation)

conv_dw_10_relu (ReLU)      (None, 14, 14, 512)       0

conv_pw_10 (Conv2D)         (None, 14, 14, 512)       262144

conv_pw_10_bn (BatchNormali  (None, 14, 14, 512)      2048
zation)

conv_pw_10_relu (ReLU)      (None, 14, 14, 512)       0

conv_dw_11 (DepthwiseConv2D  (None, 14, 14, 512)      4608
)

conv_dw_11_bn (BatchNormali  (None, 14, 14, 512)      2048
zation)

conv_dw_11_relu (ReLU)      (None, 14, 14, 512)       0
```

```
conv_pw_11 (Conv2D)          (None, 14, 14, 512)       262144

conv_pw_11_bn (BatchNormali  (None, 14, 14, 512)       2048
zation)

conv_pw_11_relu (ReLU)       (None, 14, 14, 512)       0

conv_pad_12 (ZeroPadding2D)  (None, 15, 15, 512)       0

conv_dw_12 (DepthwiseConv2D  (None, 7, 7, 512)         4608
)

conv_dw_12_bn (BatchNormali  (None, 7, 7, 512)         2048
zation)

conv_dw_12_relu (ReLU)       (None, 7, 7, 512)         0

conv_pw_12 (Conv2D)          (None, 7, 7, 1024)        524288

conv_pw_12_bn (BatchNormali  (None, 7, 7, 1024)        4096
zation)

conv_pw_12_relu (ReLU)       (None, 7, 7, 1024)        0

conv_dw_13 (DepthwiseConv2D  (None, 7, 7, 1024)        9216
)

conv_dw_13_bn (BatchNormali  (None, 7, 7, 1024)        4096
zation)

conv_dw_13_relu (ReLU)       (None, 7, 7, 1024)        0

conv_pw_13 (Conv2D)          (None, 7, 7, 1024)        1048576

conv_pw_13_bn (BatchNormali  (None, 7, 7, 1024)        4096
zation)

conv_pw_13_relu (ReLU)       (None, 7, 7, 1024)        0

global_average_pooling2d (G  (None, 1, 1, 1024)        0
lobalAveragePooling2D)

dropout (Dropout)            (None, 1, 1, 1024)        0

conv_preds (Conv2D)          (None, 1, 1, 1000)        1025000

reshape_2 (Reshape)          (None, 1000)              0

predictions (Activation)     (None, 1000)              0

=================================================================
Total params: 4,253,864
Trainable params: 4,231,976
Non-trainable params: 21,888
_________________________________________________________________
```

Transfer Learning

In [22]:
```
base_input = model.layers[0].input
```

In [23]:
```
base_output = model.layers[-4].output
```

In [24]:
```
Flat_layer = layers.Flatten()(base_output)
final_output = layers.Dense(1)(Flat_layer)
final_output = layers.Activation('sigmoid')(final_output)
```

In [25]:
```
new_model = keras.Model(inputs =base_input ,outputs = final_output)
```

In [26]:
```
new_model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 conv1 (Conv2D)              (None, 112, 112, 32)      864

 conv1_bn (BatchNormalizatio  (None, 112, 112, 32)     128
 n)

 conv1_relu (ReLU)           (None, 112, 112, 32)      0

 conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)     288

 conv_dw_1_bn (BatchNormaliz  (None, 112, 112, 32)     128
 ation)

 conv_dw_1_relu (ReLU)       (None, 112, 112, 32)      0

 conv_pw_1 (Conv2D)          (None, 112, 112, 64)      2048

 conv_pw_1_bn (BatchNormaliz  (None, 112, 112, 64)     256
 ation)

 conv_pw_1_relu (ReLU)       (None, 112, 112, 64)      0

 conv_pad_2 (ZeroPadding2D)  (None, 113, 113, 64)      0

 conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)       576

 conv_dw_2_bn (BatchNormaliz  (None, 56, 56, 64)       256
 ation)

 conv_dw_2_relu (ReLU)       (None, 56, 56, 64)        0

 conv_pw_2 (Conv2D)          (None, 56, 56, 128)       8192

 conv_pw_2_bn (BatchNormaliz  (None, 56, 56, 128)      512
 ation)

 conv_pw_2_relu (ReLU)       (None, 56, 56, 128)       0

 conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)      1152

 conv_dw_3_bn (BatchNormaliz  (None, 56, 56, 128)      512
 ation)

 conv_dw_3_relu (ReLU)       (None, 56, 56, 128)       0

 conv_pw_3 (Conv2D)          (None, 56, 56, 128)       16384

 conv_pw_3_bn (BatchNormaliz  (None, 56, 56, 128)      512
 ation)

                             (None, 56, 56, 128)       0
```

```
conv_pad_4 (ZeroPadding2D)  (None, 57, 57, 128)       0

conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)      1152

conv_dw_4_bn (BatchNormaliz  (None, 28, 28, 128)      512
ation)

conv_dw_4_relu (ReLU)       (None, 28, 28, 128)       0

conv_pw_4 (Conv2D)          (None, 28, 28, 256)       32768

conv_pw_4_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)

conv_pw_4_relu (ReLU)       (None, 28, 28, 256)       0

conv_dw_5 (DepthwiseConv2D)  (None, 28, 28, 256)      2304

conv_dw_5_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)

conv_dw_5_relu (ReLU)       (None, 28, 28, 256)       0

conv_pw_5 (Conv2D)          (None, 28, 28, 256)       65536

conv_pw_5_bn (BatchNormaliz  (None, 28, 28, 256)      1024
ation)

conv_pw_5_relu (ReLU)       (None, 28, 28, 256)       0

conv_pad_6 (ZeroPadding2D)  (None, 29, 29, 256)       0

conv_dw_6 (DepthwiseConv2D)  (None, 14, 14, 256)      2304

conv_dw_6_bn (BatchNormaliz  (None, 14, 14, 256)      1024
ation)

conv_dw_6_relu (ReLU)       (None, 14, 14, 256)       0

conv_pw_6 (Conv2D)          (None, 14, 14, 512)       131072

conv_pw_6_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_6_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_7 (DepthwiseConv2D)  (None, 14, 14, 512)      4608

conv_dw_7_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_dw_7_relu (ReLU)       (None, 14, 14, 512)       0

conv_pw_7 (Conv2D)          (None, 14, 14, 512)       262144
```

```
conv_pw_7_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_7_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_8 (DepthwiseConv2D)  (None, 14, 14, 512)      4608

conv_dw_8_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_dw_8_relu (ReLU)       (None, 14, 14, 512)       0

conv_pw_8 (Conv2D)          (None, 14, 14, 512)       262144

conv_pw_8_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_8_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_9 (DepthwiseConv2D)  (None, 14, 14, 512)      4608

conv_dw_9_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_dw_9_relu (ReLU)       (None, 14, 14, 512)       0

conv_pw_9 (Conv2D)          (None, 14, 14, 512)       262144

conv_pw_9_bn (BatchNormaliz  (None, 14, 14, 512)      2048
ation)

conv_pw_9_relu (ReLU)       (None, 14, 14, 512)       0

conv_dw_10 (DepthwiseConv2D  (None, 14, 14, 512)      4608
)

conv_dw_10_bn (BatchNormali  (None, 14, 14, 512)      2048
zation)

conv_dw_10_relu (ReLU)      (None, 14, 14, 512)       0

conv_pw_10 (Conv2D)         (None, 14, 14, 512)       262144

conv_pw_10_bn (BatchNormali  (None, 14, 14, 512)      2048
zation)

conv_pw_10_relu (ReLU)      (None, 14, 14, 512)       0

conv_dw_11 (DepthwiseConv2D  (None, 14, 14, 512)      4608
)

conv_dw_11_bn (BatchNormali  (None, 14, 14, 512)      2048
zation)

conv_dw_11_relu (ReLU)      (None, 14, 14, 512)       0
```

```
conv_pw_11 (Conv2D)          (None, 14, 14, 512)       262144

conv_pw_11_bn (BatchNormali  (None, 14, 14, 512)       2048
zation)

conv_pw_11_relu (ReLU)       (None, 14, 14, 512)       0

conv_pad_12 (ZeroPadding2D)  (None, 15, 15, 512)       0

conv_dw_12 (DepthwiseConv2D  (None, 7, 7, 512)         4608
)

conv_dw_12_bn (BatchNormali  (None, 7, 7, 512)         2048
zation)

conv_dw_12_relu (ReLU)       (None, 7, 7, 512)         0

conv_pw_12 (Conv2D)          (None, 7, 7, 1024)        524288

conv_pw_12_bn (BatchNormali  (None, 7, 7, 1024)        4096
zation)

conv_pw_12_relu (ReLU)       (None, 7, 7, 1024)        0

conv_dw_13 (DepthwiseConv2D  (None, 7, 7, 1024)        9216
)

conv_dw_13_bn (BatchNormali  (None, 7, 7, 1024)        4096
zation)

conv_dw_13_relu (ReLU)       (None, 7, 7, 1024)        0

conv_pw_13 (Conv2D)          (None, 7, 7, 1024)        1048576

conv_pw_13_bn (BatchNormali  (None, 7, 7, 1024)        4096
zation)

conv_pw_13_relu (ReLU)       (None, 7, 7, 1024)        0

global_average_pooling2d (G  (None, 1, 1, 1024)        0
lobalAveragePooling2D)

dropout (Dropout)            (None, 1, 1, 1024)        0

flatten (Flatten)            (None, 1024)              0

dense (Dense)                (None, 1)                 1025

activation (Activation)      (None, 1)                 0

=================================================================
Total params: 3,229,889
Trainable params: 3,208,001
Non-trainable params: 21,888
_________________________________________________________________
```

Setting for classification

In [27]:
```
new_model.compile(loss="binary_crossentropy",optimizer = "adam",metrics=["ac
```

In [28]:
```
new_model.fit(x,y, epochs= 2,validation_split = 0.1)
```

```
Epoch 1/2
81/81 [==============================] - 144s 2s/step - loss: 0.0337 - accura
cy: 0.9915 - val_loss: 0.0014 - val_accuracy: 1.0000
Epoch 2/2
81/81 [==============================] - 127s 2s/step - loss: 0.0110 - accura
cy: 0.9973 - val_loss: 0.0031 - val_accuracy: 0.9965
```

Out[28]:
```
<keras.callbacks.History at 0x1f6e3baf880>
```

In [29]:
```
new_model.save('my_model3.h5')
```

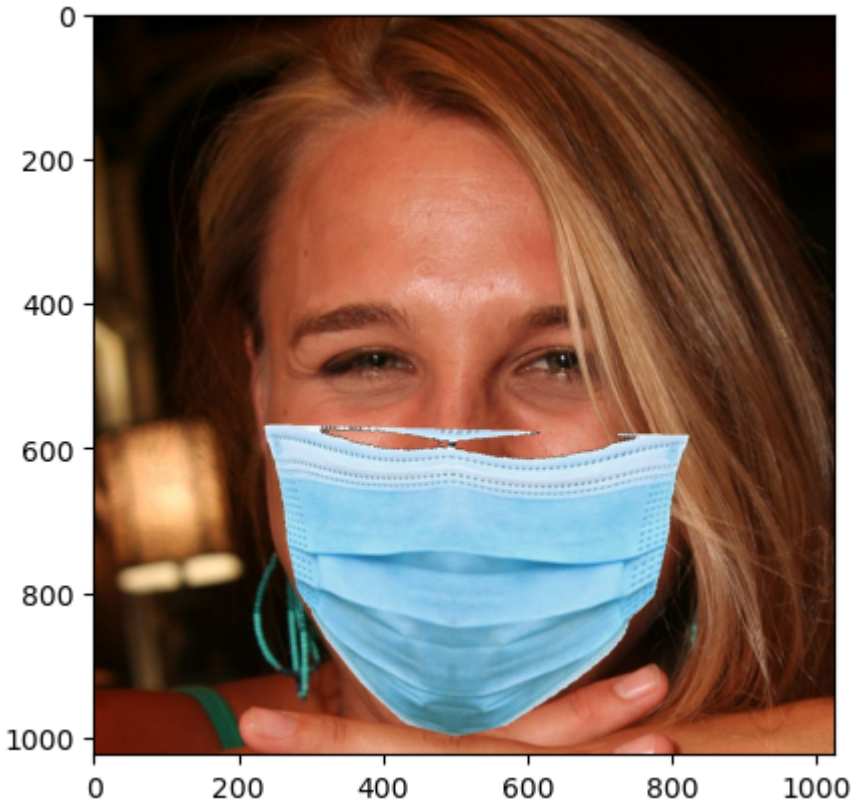
In [30]:
```
new_model = tf.keras.models.load_model('my_model.h5')
```

Checking for known prediction

In [31]:
```
frame = cv2.imread('00001_Mask.jpg')
```

In [32]:
```
plt.imshow(cv2.cvtColor(frame ,cv2.COLOR_BGR2RGB))
```

Out[32]:
```
<matplotlib.image.AxesImage at 0x1f6e8e6daf0>
```



In [33]:
```
final_image = cv2.resize(frame ,(224,224))
final_image = np.expand_dims(final_image,axis=0)
final_image = final_image/255.0
```

In [34]:
```
predictions = new_model.predict(final_image)
```

```
1/1 [==============================] - 1s 547ms/step
```

In [35]:
```
predictions
#positive value means without a mask and negative measns with mask
```

Out[35]:
```
array([[4.300461e-28]], dtype=float32)
```

Checking for an unknown image

In [36]:
```
frame = cv2.imread('istockphoto-497866588-612x612.jpg')
```

In [37]:
```
plt.imshow(cv2.cvtColor(frame ,cv2.COLOR_BGR2RGB))
```

Out[37]:
```
<matplotlib.image.AxesImage at 0x1f7011a1c70>
```

In [38]:
```
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frc
```

In [39]:
```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

In [40]:
```
faces = faceCascade.detectMultiScale(gray,1.1,4)
for x,y,w,h in faces :
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
```

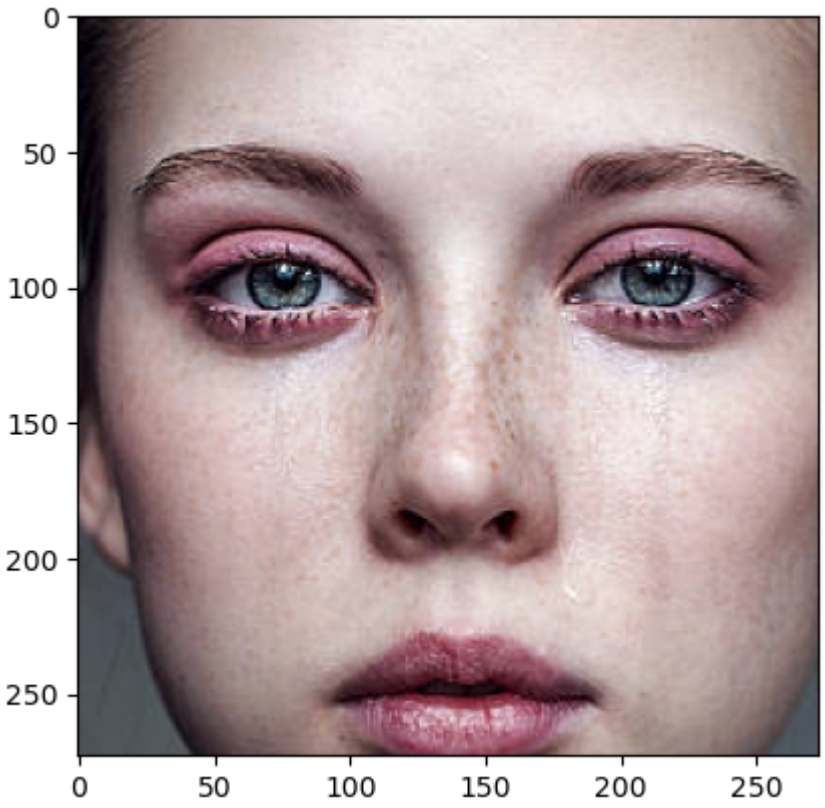
```
    cv2.rectangle(frame, (x, y), (x+w, y+h), (255,0,0), 2)
    facess =faceCascade.detectMultiScale(roi_gray)
    if len(facess) == 0:
        print("Face not detected")
    else:
        for (ex,ey,ew,eh) in facess:
            face_roi = roi_color[ey: ey+eh, ex:ex+ew]
```

In [41]:
```
plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
```

Out[41]: <matplotlib.image.AxesImage at 0x1f702327d30>

In [42]:
```
plt.imshow(cv2.cvtColor(face_roi, cv2.COLOR_BGR2RGB))
```

Out[42]: <matplotlib.image.AxesImage at 0x1f70bc55c70>



In [43]:
```
final_image = cv2.resize(face_roi ,(224,224))
final_image = np.expand_dims(final_image,axis=0)
final_image = final_image/255.0
```

In [44]:
```
predictions=new_model.predict(final_image)
```

```
1/1 [==============================] - 0s 26ms/step
```

In [45]:
```
predictions
```

Out[45]:
```
array([[0.00194874]], dtype=float32)
```

Real Time

In [52]:
```
import cv2

path = "haarcascade_frontalface_default.xml"
font_scale = 1.5
font = cv2.FONT_HERSHEY_PLAIN

rectangle_bgr = (255, 255, 255)
img = np.zeros((500, 500))
text = "Some text in a box!"
(text_width, text_height) = cv2.getTextSize(text, font, fontScale=font_scale
text_offset_x = 10
text_offset_y = img.shape[0] - 25

((text_offset_x, text_offset_y), (text_offset_x + text_width +
```

```
cv2.rectangle(img, box_coords[0], box_coords[1], rectangle_bgr, cv2.FILLED)
cv2.putText(img, text, (text_offset_x,text_offset_y), font, fontScale=font_s

cap = cv2.VideoCapture(1)
if not cap.isOpened():
    cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cann't open webcam")

while True:
    ret,frame = cap.read()

    faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(gray, 1.1,4)
    for x,y,w,h in faces :
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255,0,0), 2)
        facess =faceCascade.detectMultiScale(roi_gray)
        if len(facess) == 0:
            print("Face not detected")
        else:
            for (ex,ey,ew,eh) in facess:
                face_roi = roi_color[ey: ey+eh, ex:ex+ew]

    final_image = cv2.resize(face_roi, (244,244))
    final_image = np.expand_dims(final_image,axis = 0)
    final_image = final_image/255.0
    font = cv2.FONT_HERSHEY_SIMPLEX
    predictions = new_model.predict(final_image)

    font_scale = 1.5
    font = cv2.FONT_HERSHEY_PLAIN

    if(predictions < 0.1):
        status = "No Musk"

        x1,y1,w1,h1 = 0,0,175,75
        cv2.rectangle(frame, (x1, x1), (x1 + w1, y1 + h1), (0,0,0), -1)
        cv2.putText(frame, status, (x1 + int(w1/10),y1 +int(h1/2)), cv2.FONT
        cv2.putText(frame,status,(100, 150),font, 3,(0, 0, 255),2,cv2.LINE_4
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))
    else:
        status = "Face Mask"

        x1,y1,w1,h1 = 0,0,175,75
        cv2.rectangle(frame, (x1, x1), (x1 + w1, y1 + h1), (0,0,0), -1)
        cv2.putText(frame, status, (x1 + int(w1/10),y1 +int(h1/2)), cv2.FONT
        cv2.putText(frame,status,(100, 150),font, 3,(0, 255, 0),2,cv2.LINE_4
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0))


    cv2.imshow('Face Mask Detection ',frame)
```

```
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 35ms/step
Face not detected
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 31ms/step
```

```
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 34ms/step
Face not detected
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 35ms/step
Face not detected
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 35ms/step
Face not detected
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 32ms/step
Face not detected
1/1 [==============================] - 0s 34ms/step
Face not detected
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
Face not detected
1/1 [==============================] - 0s 31ms/step
Face not detected
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 36ms/step
```

```
Face not detected
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 38ms/step
Face not detected
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
Face not detected
1/1 [==============================] - 0s 31ms/step
Face not detected
1/1 [==============================] - 0s 41ms/step
Face not detected
1/1 [==============================] - 0s 36ms/step
Face not detected
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
Face not detected
1/1 [==============================] - 0s 31ms/step
Face not detected
1/1 [==============================] - 0s 33ms/step
Face not detected
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 34ms/step
Face not detected
1/1 [==============================] - 0s 35ms/step
```

In [ ]:

In [ ]: