```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator


train_datagen = ImageDataGenerator(
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)
train_set = train_datagen.flow_from_directory(
        'dataset/training_set',
        target_size=(64, 64),
        batch_size=32,
        class_mode='binary')


test_datagen = ImageDataGenerator(rescale=1./255)
test_set = test_datagen.flow_from_directory(
        'dataset/test_set',
        target_size=(64, 64),
        batch_size=32,
        class_mode='binary')


cnn = tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_sha
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
cnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
cnn.fit(x = train_set, validation_data = test_set, epochs= 10)


import numpy as np
from keras.preprocessing import image
test_image = image.load_img(
    'dataset/single_prediction/Sach.jpg',
    target_size=(64,64))
test_image=image.img_to_array(test_image)
test_image = np.expand_dims(test_image,axis=0)
result = cnn.predict(test_image)
train_set.class_indices
if result[0][0]== 1:
  prediction = 'With Mask'
else:
  prediction = 'Without Mask'

print(prediction)
```

```
cnn.save('facemask_detect.h5')


import numpy as np
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from keras.models import load_model
import cv2


cnn_model = load_model('facemask_detect.h5')
camera = cv2.VideoCapture(0)
face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalfa


rectcolor_dict={'With Mask':(0,255,0),'Without Mask':(0,0,255)}


while True:
    ret,img = camera.read()
    faces = face_classifier.detectMultiScale(img,scaleFactor=1.3,minNeighbors=5)

    for x,y,w,h in faces:
        face_img = img[y:y+w,x:x+w]
        face_img = cv2.resize(face_img,(64,64))
        face_img=image.img_to_array(face_img)
        face_img = np.expand_dims(face_img,axis=0)
        result = cnn_model.predict(face_img)

        if result[0][0]== 1:
            prediction = 'With Mask'
        else:
            prediction = 'Without Mask'
        cv2.rectangle(img,(x,y),(x+w,y+h),rectcolor_dict[prediction],2)
        cv2.rectangle(img,(x,y-40),(x+w,y),rectcolor_dict[prediction],-1)
        cv2.putText(img,prediction,(x,y-10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255)

    cv2.imshow('LIVE',img)
    key = cv2.waitKey(1)

    if (key==27):
        break

cv2.destroyAllWindows
camera.release()
```