## source code:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import serial
from mail import report_send_mail
import time
from mail import*
from pygame import mixer



net = cv2.dnn.readNetFromDarknet("yolov3.cfg","yolov3.weights")




classes = ['normal','Fire','normal']



def classifer(label):
    print(label)
    index = int(classes.index(label))
    mixer.init()
    mixer.music.load("sound.mp3")
    mixer.music.set_volume(0.9)
    mixer.music.play()



cap = cv2.VideoCapture(0)
```

```
while 1:
    _, img = cap.read()
    img = cv2.resize(img,(1280,720))
    hight,width,_ = img.shape
    blob = cv2.dnn.blobFromImage(img, 1/255,(416,416),(0,0,0),swapRB = True,crop= False)


    net.setInput(blob)


    output_layers_name = net.getUnconnectedOutLayersNames()


    layerOutputs = net.forward(output_layers_name)


    boxes =[]
    confidences = []
    class_ids = []


    for output in layerOutputs:
        for detection in output:
            score = detection[5:]
            class_id = np.argmax(score)
            confidence = score[class_id]
            if confidence > 0.7:
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * hight)
                w = int(detection[2] * width)
                h = int(detection[3]* hight)
                x = int(center_x - w/2)
                y = int(center_y - h/2)
                boxes.append([x,y,w,h])
                confidences.append((float(confidence)))
```

```
            class_ids.append(class_id)



indexes = cv2.dnn.NMSBoxes(boxes,confidences,.5,.4)

boxes =[]
confidences = []
class_ids = []

for output in layerOutputs:
    for detection in output:
        score = detection[5:]
        class_id = np.argmax(score)
        confidence = score[class_id]
        if confidence > 0.5:
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * hight)
            w = int(detection[2] * width)
            h = int(detection[3]* hight)
            x = int(center_x - w/2)
            y = int(center_y - h/2)
            boxes.append([x,y,w,h])
            confidences.append((float(confidence)))
            class_ids.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes,confidences,.8,.4)
font = cv2.FONT_HERSHEY_PLAIN
colors = np.random.uniform(0,255,size =(len(boxes),3))
if  len(indexes)>0:
    for i in indexes.flatten():
        x,y,w,h = boxes[i]
```

```
            label = str(classes[class_ids[i]])
            cv2.imwrite('image.jpg', img)
            classifer(label)
            confidence = str(round(confidences[i],2))
            color = colors[i]
            cv2.rectangle(img,(x,y),(x+w,y+h),color,2)
            cv2.putText(img,label + " " + confidence, (x,y+400),font,2,color,2)

    cv2.imshow('img',img)
    if cv2.waitKey(1) == ord('q'):
        break

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