import cv2

import numpy as np

import datetime

import subprocess

from pydub import AudioSegment

import time

import sys

sys.path.append('/home/pi/Templates')

import mask\_pred

net = cv2.dnn.readNet('training\_final.weights', 'testing.cfg')

net.setPreferableBackend(cv2.dnn.DNN\_BACKEND\_OPENCV)

net.setPreferableTarget(cv2.dnn.DNN\_TARGET\_CPU)

classes = []

with open("classes.txt", "r") as f:

classes = f.read().splitlines()

while True:

g = datetime.datetime.now()

subprocess.Popen("sudo fswebcam image.jpg",shell=True).communicate()

img = cv2.imread('/home/pi/Mask\_Detection/image.jpg')

height, width, \_ = img.shape

font = cv2.FONT\_HERSHEY\_PLAIN

blob = cv2.dnn.blobFromImage(img, 1/255, (416, 416), (0,0,0), swapRB=True, crop=False)

net.setInput(blob)

output\_layers\_names = net.getUnconnectedOutLayersNames()

layerOutputs = net.forward(output\_layers\_names)

boxes = []

confidences = []

class\_ids = []

for output in layerOutputs:

for detection in output:

scores = detection[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.2:

center\_x = int(detection[0]\*width)

center\_y = int(detection[1]\*height)

w = int(detection[2]\*width)

h = int(detection[3]\*height)

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, 0.4, 0.4)

img, label = mask\_pred.mask(indexes,x, y, w, h,boxes,classes,confidences,class\_ids,img,g,font)

cv2.imshow('Image', img)

key = cv2.waitKey(1000)

time.sleep(3)

if(label=="MASK\_NOT\_FOUND"):

audio = AudioSegment.from\_mp3("/home/pi/Mask\_Detection/Telugu.mp3")

subprocess.call(["ffplay", "-nodisp", "-autoexit", '/home/pi/Mask\_Detection/Telugu.mp3'])

audio = AudioSegment.from\_mp3("/home/pi/Mask\_Detection/English.mp3")

subprocess.call(["ffplay", "-nodisp", "-autoexit", '/home/pi/Mask\_Detection/English.mp3'])

key = cv2.waitKey(1000)

cv2.destroyAllWindows()