OUR Journal: ODU Undergraduate Research Journal

Volume 10 Article 7

2023

COVID-19 Crowd Detection

Mustafa Ibrahim
Old Dominion University

Aly M. Zeineldin
Old Dominion University

Yameen Khan

Ayman Elmesalami
Old Dominion University

Soad Ibrahim
Old Dominion University

COVID-19 CROWD DETECTION

By Mustafa Ibrahim, Aly M. Zeineldin, & Yameen Khan

Advisors: Ayman El Mesalami & Soad Ibrahim

TABLE OF CONTENTS

ABSTRACT— Object detection was introduced by researchers for face detection. Researchers explain how the detected face is divided into minor frames to be recognized by the algorithm. Due to COVID-19 and government regulations, many people face problems going to shopping centers and shop safely. It has been very hard for both the government and the people to manage social distancing. In our study, we developed a system using Raspberry Pi-4 that will detect the distance between people along with counting the number of distance and mask violations. An error message will appear on the screen in red, showing the total number of distance and mask violations, which could later be used by the customer as statistical evidence for better safety precautions.

I. INTRODUCTION

Individuals' health safety is in danger in various areas like shopping centers, the railroad system, and lanes. This pandemic circumstance, where individuals assemble in crowds, made it necessary to start managing social distance. Managing a crowd of varying densities involves detection of the individual humans in the crowd. Other tracking algorithms have faced various issues in crowd detection due to the presence of other objects in the environment.

Social distancing means staying a certain distance away from people in public places. Due to COVID-19, social distancing became crucial. The goal is to decrease the transmission rates of COVID-19 by reducing contact between infected and healthy individuals.

COVID-19 has proven to be one of the most dangerous viruses to ever exist. In comparison to other diseases like H1N1, swine flu, and Ebola COVID-19, swine flu is the most contagious. COVID-19's airborne nature makes its transmission faster as no physical contact is required. COVID-19 death numbers are higher than Ebola's. According to the New York Times, COVID-19 has claimed the lives of 1.02 million people in the United States alone. The United States Centers for Disease Control and Prevention (CDC) came up with a critical guideline to help contain such a pandemic. A mask should be required in all indoor spaces for everyone ages 2 and above. A mask should successfully cover the nose and mouth of an individual. Several types of masks are available, with the most common one being the N95 respirator. Other types are also available (N99, N100, P95, P99, P100, R95, R99, and R100), which offer the same or better protection as the N95 respirator.

Figure 1 Mask Types

During the summer of 2022, computer science undergraduates at Old Dominion University undertook a deeper investigation to help with COVID-19 safety measures. This study provides a more efficient tracking method that focuses on capturing a video that is later divided into frames and then counts the number of people in each frame. If they are closer to the assigned distance, then it will automatically display an error message in red with the number of distance violations. Moreover, the algorithm is designed to recognize the number of face masks worn by individuals and keep track of the number of face mask violations.

II. METHODOLOGY

Equipment

1. The Raspberry Pi V4 Model B (4GB)

The Raspberry Pi 4 is the fourth and latest model of the Raspberry Pi computers. It has a Broadcom BCM2711B0 quad-core ARM processor along with a 4k-capable Broadcom Video Core VI video processor, a USB 3.0 port, and USB Type-C for power.

FIGURE 2 RASPBERRY PI 4

2. Raspberry Pi Camera Module v2

The v2 camera has a Sony IMX219-megapixel sensor. The camera could be used for both photos and videos. It supports 1080p30, 720p60, and VGA90 video modes. It has a 15cm cable that connects to the CSI port on the Raspberry Pi. It is compatible with all Raspberry Pi models: 1, 2, 3, and 4. It can be operated through the MMAL and V4L libraries, along with various third-party libraries such as the Picamera Python library.

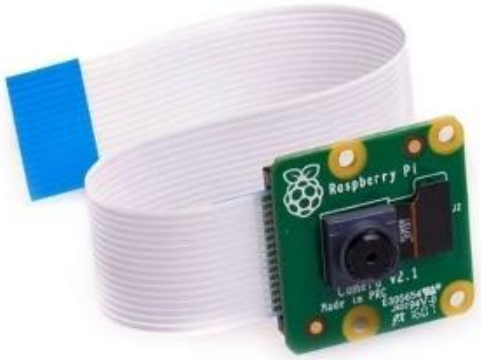

FIGURE 3 RASPBERRY PI CAMERA MODULE V2

3. SanDisk Micro SD card 32GB

A Secure Digital card is a memory card used to store data on various technological devices like cameras and mobile phones. Any SD card would work for our purposes so long as it is a Micro SD card and fits in the Raspberry Pi. The capacity would vary depending on the user's intended use and data size.

FIGURE 4 SANDISK MICRO SD CARD

4. The Raspberry Pi 4 Power Supply

USB-C power supply with 5.1V/3.0A DC output.

FIGURE 5 RASPBERRY PI 4 POWER SUPPLY

5. CanaKit Premium Raspberry Pi 4 Micro HDMI Cable – 6 Feet

Micro HDMI to HDMI cable to connect the Raspberry Pi 4 to the TV screen for visual display. A 6 ft cable was used, producing up to 60 frames per second and 4k resolution. CanaKit has been previously tested on the Raspberry Pi 4, but other brands are also available.

FIGURE 6 HDMI CABLE

III. IMPLEMENTATION

Figure 7 is a picture captured from a video in a public environment. The blue frames indicate that the distance maintained between individuals is valid. Meanwhile, the red frames indicate that there is a distance violation.

Figure 7 Crowd detection in frames

First, the Raspberry Pi camera captures the video of a public setting. Then the video is split into minor frames where the frames are divided and processed frame by frame. Next, with the use of TensorFlow, only people are detected. When the process is complete, the algorithm measures the distance between each frame using pixels. If it is less than the required distance, an error message will appear showing the distance violation number. The same process is repeated to accurately recognize and identify face masks.

IV. LIBRARIES

1. OpenCV

- OpenCV is a machine learning library that focuses on video capturing and image processing. It supports multiple programming languages including C++, Java, and Python. For the purposes of this study, OpenCV was used for object detection to identify people and masks.

- Link: https://pypi.org/project/opencv-python/

2. NumPy

- The Numerical Python library (NumPy) is a library designed to work with multidimensional arrays to perform complex mathematical operations such as trigonometric, statistical, and algebraic expressions.

- Link: https://pypi.org/project/numpy/

3. Multiprocessing

- This library allows the system to run more than one operation at the same time. In the case of this study, more than one processor is operating at the same moment as configurations, spreadsheets, and data recording are all running simultaneously.

- Link: https://pypi.org/project/multiprocess/

4. Selenium

- Selenium is a web-testing library framework.

- Link: https://pypi.org/project/selenium/

V. FUNCTIONS

1. Check

- This function is responsible for finding the distance between individuals where the violation distance is set to 2 ft and could always be changeable.

2. Setup

- This function is responsible for using the Yolo library for weights and configuration. As part of Yolo, coco.names was used as an image dataset to identify objects. Using deep neural networks, both people and masks were successfully trained.

3. Setup 2

- This function is responsible for defining all the variables needed (threshold, weights, size, confidence) of a new model. These variables are used to train the data being identified in the setup function.

4. Inference from file

- Responsible for reading the image file and returning it into the inferred image.

5. Inference

- This function is responsible for breaking the video into frames, detecting people and masks, and drawing the frame around them. First the image is read, resized, and then added to the list.

6. Process

- This function is required to keep a record of mask and distance violations and the number of faces in the program to be printed out on the screen later in the program.

7. ThingSpeak

- This function uses the data recorded in the process function and uploads it on the ThingSpeak website to receive graphical data.

8. Alert

- This function opens WhatsApp and sends the client a safety measurement violation if a majority (> 50%) is violating COVID-19 guidelines.

VI. OUTPUT

The input of the program can be either a video or observance from the live camera. As shown in Figure 4, a video is uploaded to the Raspberry Pi. First, frames are drawn on each person. A green color shows whether there is no distance violation. Otherwise, a red color is displayed along with a yellow line between individuals to represent the distance violation. Another frame is drawn around their masks. Green represents that the mask is worn correctly, while red means there is a mask violation, and yellow indicates that the mask is worn inappropriately.

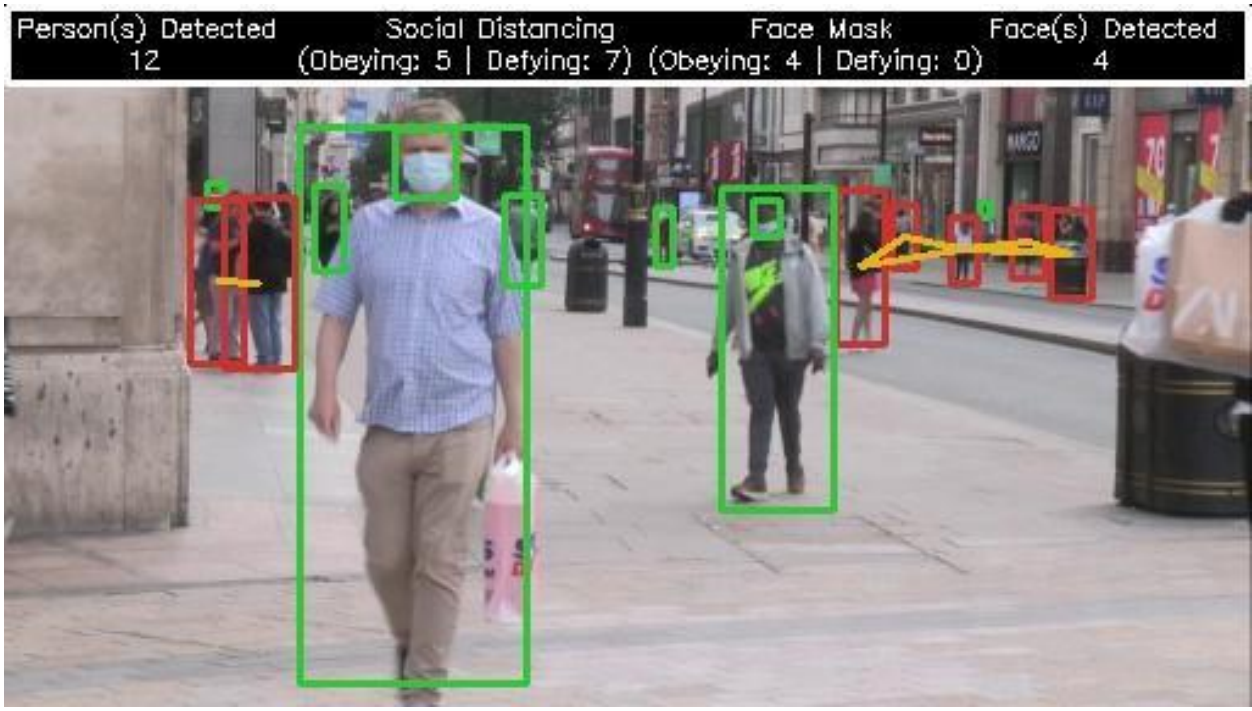

Figure 8 Output

A message appears at the top of the screen showing all the data calculated in the program: total number of people, social distance violations and compliances, face mask violations and compliances, and total number of face masks detected.

VII. EXTRA FEATURES

The data is then uploaded to ThinkerSpeak to show a visual representation of it using graphs and charts, making it easier for further analysis. Also, ThinkerSpeak data is sent to the owner of the program via WhatsApp.

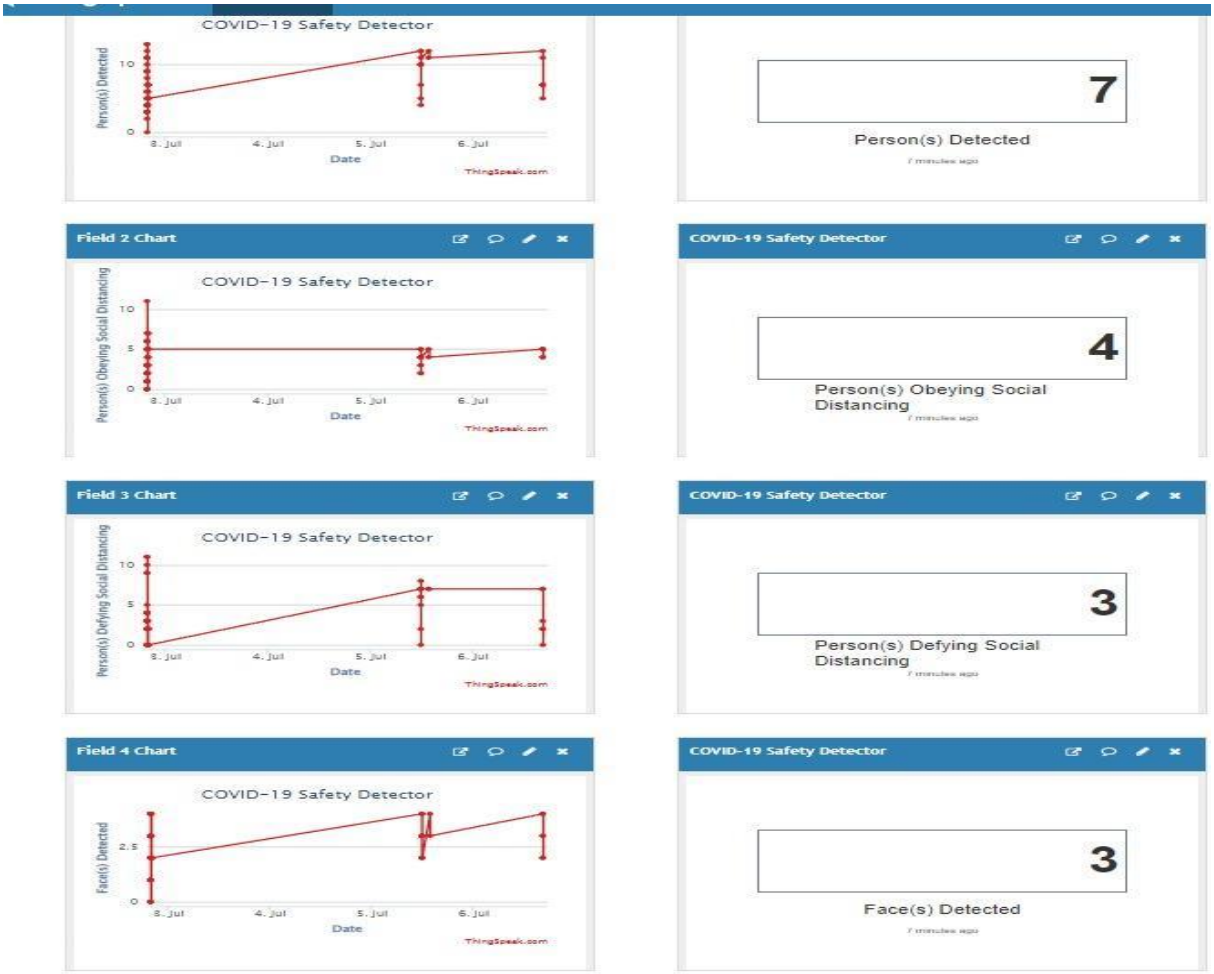

Figure 9 ThinkerSpeak results

Output.mp4

FIGURE 7 OUTPUT LIVE DEMONSTRATION

VIII. Future Work

Further research could be conducted to make this study more applicable in real life. It can be used to trigger alarms if many distance violations exist. It can be used to evaluate people's responses to government regulations in different cities and states. This study could be implemented in different environments like classrooms, public transportation, and any public setting. It can be used as statistical evidence for COVID-19 related reports. It could also be used to track immigrants' responses to COVID-19 guidelines through sending out mobile alerts in cases of mask or distance violations.

IX. Conclusion

Overall, social distancing is crucial, especially in times of a pandemic. The idea of object detection could effectively be used as a data collection tool along with an alert device to ensure the safety of the community. If social distancing guidelines could be safely followed, COVID-19 cases could dramatically decrease. Face masks are another important aspect in containing COVID-19 cases. Research proved the effectiveness of COVID-19 surgical masks as villages that enforced mask guidelines were less likely to suffer high numbers of COVID-19 cases. The system is used to monitor people's responses to COVID-19 guidelines. It shows statistical evidence through Thinker Swim in the form of graphs to represent the effectiveness of social distancing and mask guidelines of a certain environment.

CODE

```python
from dataclasses import field
from urllib.request import urlopen
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.by import By
from multiprocessing import Queue
from keyboard import press
import numpy as np
import multiprocessing
import time
import cv2
import pyautogui
import argparse
import imutils
import os
def check(a, b):
    dist = ((a[0] - b[0]) ** 2 + 550 / ((a[1] + b[1]) / 2) * (a[1] - b[1]) ** 2) ** 0.5
    calibration = (a[1] + b[1]) / 2
    if 0 < dist < 0.5 * calibration:
        return True
    else:
        return False
def setup(yolo):
    global net, ln, LABELS
    weights = os.path.sep.join([yolo, "yolov3.weights"])
    config = os.path.sep.join([yolo, "yolov3.cfg"])
    labels_path = os.path.sep.join([yolo, "coco.names"])
    LABELS = open(labels_path).read().strip().split("\n")
    net = cv2.dnn.readNetFromDarknet(config, weights)
    ln = net.getLayerNames()
    ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]
def setup2(config, model, labels, size=416, confidence=0.5, threshold=0.3):
    global confidence1, threshold1, size1, labels1, net1
    confidence1 = confidence
    threshold1 = threshold
    size1 = size
    labels1 = labels
    net1 = cv2.dnn.readNetFromDarknet(config, model)
def inference_from_file(file):
    mat = cv2.imread(file)
    return inference(mat)
def inference(image):
    ih, iw = image.shape[:2]
    ln = net1.getLayerNames()
    ln = [ln[i[0] - 1] for i in net1.getUnconnectedOutLayers()]
    blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (size1, size1), swapRB=True, crop=False)
    net1.setInput(blob)
    start = time.time()
    layer_outputs = net1.forward(ln)
    end = time.time()
    inference_time = end - start
    boxes = []
    confidences = []
    class_ids = []
    for output in layer_outputs:
        for detection in output:
            scores = detection[5:]
```

```
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > confidence1:
                box = detection[0:4] * np.array([iw, ih, iw, ih])
                (center_x, center_y, width, height) = box.astype("int")
                x = int(center_x - (width / 2))
                y = int(center_y - (height / 2))
                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                class_ids.append(class_id)
    idxs = cv2.dnn.NMSBoxes(boxes, confidences, confidence1, threshold1)
    results = []
    if len(idxs) > 0:
        for i in idxs.flatten():
            x, y = (boxes[i][0], boxes[i][1])
            w, h = (boxes[i][2], boxes[i][3])
            key = class_ids[i]
            confidence = confidences[i]
            results.append((key, labels1[key], confidence, x, y, w, h))
    return iw, ih, inference_time, results
def process(image):
    global processedImg, mask_violation, no_mask_violation, distance_violation, no_distance_violation, people, faces, closer
    mask_violation = 0
    no_mask_violation = 0
    distance_violation = 0
    no_distance_violation = 0
    people = 0
    faces = 0
    closer = 0
    (H, W) = (None, None)
    frame = image.copy()
    width, height, inference_time, results = inference(frame)
    for detection in results:
        key, name, confidence, x, y, w, h = detection
        if(key > 0):
            mask_violation += 1
        else:
            no_mask_violation +=1
        color = colors[key]
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
    if W is None or H is None:
        (H, W) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)
    layer_outputs = net.forward(ln)
    confidences = []
    outline = []
    for output in layer_outputs:
        for detection in output:
            scores = detection[5:]
            maxi_class = np.argmax(scores)
            confidence = scores[maxi_class]
            if LABELS[maxi_class] == "person" and confidence > 0.5:
                box = detection[0:4] * np.array([W, H, W, H])
                (center_x, center_y, width, height) = box.astype("int")
                x = int(center_x - (width / 2))
                y = int(center_y - (height / 2))
```

```
                outline.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
    box_line = cv2.dnn.NMSBoxes(outline, confidences, 0.5, 0.3)
    if len(box_line) > 0:
        flat_box = box_line.flatten()
        pairs = []
        center = []
        status = []
        for i in flat_box:
            (x, y) = (outline[i][0], outline[i][1])
            (w, h) = (outline[i][2], outline[i][3])
            center.append([int(x + w / 2), int(y + h / 2)])
            status.append(False)
        for i in range(len(center)):
            for j in range(len(center)):
                close = check(center[i], center[j])
                if close:
                    pairs.append([center[i], center[j]])
                    status[i] = True
                    status[j] = True
        people = len(box_line)
        index = 0
        for i in flat_box:
            (x, y) = (outline[i][0], outline[i][1])
            (w, h) = (outline[i][2], outline[i][3])
            if status[index] == True:
                cv2.rectangle(frame, (x, y), (x + w, y + h), (50, 50, 204), 2)
                closer += 1
            elif status[index] == False:
                cv2.rectangle(frame, (x, y), (x + w, y + h), (55, 201, 45), 2)
            index += 1
        for h in pairs:
            cv2.line(frame, tuple(h[0]), tuple(h[1]), (22, 180, 231), 2)
    distance_violation = closer
    no_distance_violation = people - closer
    faces = mask_violation + no_mask_violation
    cv2.rectangle(frame,(0,0),(512,32),(0,0,0),-1)
    cv2.rectangle(frame,(1,1),(510,32),(255,255,255),2)
    combined_text1 = "Person(s) Detected        Social Distancing          Face Mask       Face(s) Detected"
    combined_text2 = "        {}          (Obeying: {} | Defying: {}) (Obeying: {} | Defying: {})        {}".format(people, no_distance_violation, distance_violation, no_mask_
    cv2.putText(frame, combined_text1, (5, 13), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (255, 255, 255), 0)
    cv2.putText(frame, combined_text2, (5, 26), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (255, 255, 255), 0)
    field1 = people
    field2 = no_distance_violation
    field3 = distance_violation
    field4 = faces
    field5 = no_mask_violation
    field6 = mask_violation
    thingspeak = base_url + "&field1={}&field2={}&field3={}&field4={}&field5={}&field6={}".format(field1, field2, field3, field4, field5, field6)
    urlopen(thingspeak)
    processedImg = frame.copy()
def alert(queue):
    contact_name = "'Mustafa Atif Ibrahim'"
    search_for = "Mustafa Atif Ibrahim"
    message = queue.get()
```

```
    whatsapp_url = r'https://web.whatsapp.com/'
    opt = Options()
    opt.add_argument(r'user-data-dir=D:\Desktop Backup\COVID-19 Safety Detector\whatsapp-profile')
    driver = webdriver.Chrome(r'D:\Desktop Backup\COVID-19 Safety Detector\chromedriver\chromedriver.exe', options=opt)
    driver.get(whatsapp_url )
    time.sleep(30)
    search_box = driver.find_element(By.XPATH, "//div[@title='Search input textbox']")
    ActionChains(driver).move_to_element(search_box).click(search_box).perform()
    time.sleep(2)
    search_box.send_keys(search_for)
    time.sleep(2)
    chat_box = driver.find_element(By.XPATH, "//span[@title="+contact_name+"]")
    ActionChains(driver).move_to_element(chat_box).click(chat_box).perform()
    time.sleep(2)
    clip_button = driver.find_element(By.XPATH, "//span[@data-testid='clip']")
    ActionChains(driver).move_to_element(clip_button).click(clip_button).perform()
    time.sleep(2)
    image_button = driver.find_element(By.XPATH, "//span[@data-testid='attach-image']")
    ActionChains(driver).move_to_element(image_button).click(image_button).perform()
    time.sleep(2)
    pyautogui.typewrite(r"D:\Desktop Backup\COVID-19 Safety Detector\alert.jpg",0.1)
    time.sleep(2)
    press('enter')
    time.sleep(2)
    message_box = driver.find_element(By.XPATH, "//div[@data-testid='drawer-middle']//span//div//span//div//div//div//div//div//div//div//div//div//div//div//div//div[@role='textbox']")
    message_box.send_keys(message)
    time.sleep(2)
    sending = driver.find_element(By.XPATH, "//span[@data-testid='send']")
    ActionChains(driver).move_to_element(sending).click(sending).perform()
    time.sleep(10)
    driver.close()
    driver.quit()
def main():
    ap = argparse.ArgumentParser()
    ap.add_argument('-n', '--network', default="normal", help='Network Type: normal / tiny / prn')
    ap.add_argument('-d', '--device', default=0, help='Device to use')
    ap.add_argument('-s', '--size', default=416, help='Size for yolo')
    ap.add_argument('-c', '--confidence', default=0.5, help='Confidence for yolo')
    args = ap.parse_args()
    classes = ["Wearing Mask Properly", "Not Wearing Mask Properly", "Not Wearing Mask At All"]
    if args.network == "normal":
        setup2("models/mask-yolov4.cfg", "models/mask-yolov4.weights", classes)
    elif args.network == "prn":
        setup2("models/mask-yolov3-tiny-prn.cfg", "models/mask-yolov3-tiny-prn.weights", classes)
    else:
        setup2("models/mask-yolov4-tiny.cfg", "models/mask-yolov4-tiny.weights", classes)
    global colors
    colors = [(55, 201, 45), (22, 180, 231), (50, 50, 204)]
    write_key = "MI46NVROLNG2XVXX"
    global base_url
    base_url = "https://api.thingspeak.com/update?api_key={}".format(write_key)
    create = None
    frameno = 0
    process_time = 0
    alert_image = "alert.jpg"
    input_filename = "input1.mp4"
    yolo = "yolo-coco/"
```

```
    yolo = "yolo-coco/"
    output_filename = "output.avi"
    if(input_filename == ""):
        cap = cv2.VideoCapture(0)
    else:
        cap = cv2.VideoCapture(input_filename)
    while(True):
        ret, frame = cap.read()
        if not ret:
            break
        current_img = frame.copy()
        current_img = imutils.resize(current_img, width=512, height=512)
        frameno += 1
        if(frameno%2 == 0 or frameno == 1):
            setup(yolo)
            process(current_img)
            processed_frame = processedImg
            cv2.imshow("COVID-19 Safety Violations Detector", processed_frame)
            if(time.time() - process_time >= 150 and process_time > 0):
                process1.kill()
            if(people > 0 and faces > 0 and time.time() - process_time >= 300):
                if(float(distance_violation) / float(people) > 0.5 and float(mask_violation) / float(faces) > 0.5):
                    cv2.imwrite(alert_image, processed_frame)
                    queue1 = Queue()
                    process1 = multiprocessing.Process(target= alert, args=(queue1,))
                    process_time = time.time()
                    process1.start()
                    queue1.put('COVID-19 Safety Alert! Majority are defying face mask and social distancing rules. Please spread the COVID-19 safety rules of obeying face mask
                else:
                    if(float(mask_violation) / float(faces) > 0.5):
                        cv2.imwrite(alert_image, processed_frame)
                        queue1 = Queue()
                        process1 = multiprocessing.Process(target= alert, args=(queue1,))
                        process_time = time.time()
                        process1.start()
                        queue1.put('COVID-19 Safety Alert! Majority are defying face mask rule. Please spread the COVID-19 safety rule of obeying face mask.')
                    elif(float(distance_violation) / float(people) > 0.5):
                        cv2.imwrite(alert_image, processed_frame)
                        queue1 = Queue()
                        process1 = multiprocessing.Process(target= alert, args=(queue1,))
                        process_time = time.time()
                        process1.start()
                        queue1.put('COVID-19 Safety Alert! Majority are defying social distancing rule. Please spread the COVID-19 safety rule of obeying social distancing.')
            if create is None:
                fourcc = cv2.VideoWriter_fourcc(*'XVID')
                create = cv2.VideoWriter(output_filename, fourcc, 30, (processed_frame.shape[1], processed_frame.shape[0]), True)
        create.write(processed_frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
if __name__ == '__main__':
    main()
```

REFERENCES

"CanaKit Premium Raspberry Pi 4 Micro HDMI Cable - 6 Feet." *CanaKit*, https://www.canakit.com/raspberry-4-mico-hdmi-cable.html.

"CanaKit Premium Raspberry Pi 4 Micro HDMI Cable - 6 Feet." *CanaKit*, https://www.canakit.com/raspberry-4-mico-hdmi-cable.html.

"CDC Recommendation for Masks and Travel." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 3 May 2022, https://www.cdc.gov/media/releases/2022/s0503-covid-19-travel.html.

"How to Protect Yourself & Others." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html.

Johnson, Dave. "What Is an SD Card? Here's What You Need to Know about the Small Memory Cards for Electronic Devices." *Business Insider*, Business Insider, 27 Jan. 2021, https://www.businessinsider.com/what-is-an-sd-card.

News Center. "Surgical Masks Reduce COVID-19 Spread, Large-Scale Study Shows." *News Center*, 1 Sept. 2021, https://med.stanford.edu/news/all-news/2021/09/surgical-masks-covid-19.html.

"Official Raspberry Pi 4 Power Supply (USB-C) - White." *CanaKit*, https://www.canakit.com/official-raspberry-pi-4-power-supply.html#:~:text=Due%20to%20the%20higher%20power,B%2B%2C%20A%2B%20or%20Pi%20Zero.

Piltch, Avram. "Best Microsd Cards for Raspberry Pi 2022." *Tom's Hardware*, Tom's Hardware, 2 Aug. 2022, https://www.tomshardware.com/best-picks/raspberry-pi-microsd-cards.

Raspberry Pi. “Buy A Raspberry Pi 15W USB-C Power Supply.” *Raspberry Pi*, https://www.raspberrypi.com/products/type-c-power-supply/.

Raspberry Pi. “Buy A Raspberry Pi Camera Module 2.” *Raspberry Pi*, https://www.raspberrypi.com/products/camera-module-v2/.

Westover, Brian. “Raspberry Pi 4 Model B Review.” *Tom's Guide*, 27 May 2021, https://www.tomsguide.com/reviews/raspberry-pi-4-model-b.