Robocup@home Education online challenge 2021

The high school affiliated to Xi'an Jiaotong University,China
Mentor:Zesong Liao
Team members:Yuhan wang,Zhaoyu Cheng,JiaChen Hu
Team: Anti epidemic vanguard

Report for the Intelligent Anti-epidemic Service Robot

1.Designing background

The situation of the COVID-19 is still grim even to this day. With the resumption of work and production, the flow of people is increasing, and people are facing complex process, such as measuring body temperature, checking health codes and masks, registering information on outsiders and so on, to ensure their health when they want to enter somewhere.Nowadays, most of these are checked by man, which undoubtedly brought huge work and pressure to the relevant staff. What's worse, there may also be staff neglecting their duties and failing to check carefully as these tasks were hard and boring. That could bring hidden danger to the public.

Due to the impact of the epidemic, robots has become increasingly necessary.Many technology companies have begun to design anti-epidemic robots that can achieve the functions of getting body temperature, mask recognition, and historical data analysis. However, according to our team's survey, most of these robots have scattered and single functions, therefore they couldn't be used in different environments and do comprehensive work. So we designed the Intelligent Anti-epidemic Service Robot to integrate the functions and added new idea so that it can bring all-round assistance.

2.Realized functions

①Verify the body temperature:
If someone's temperature is higher than normal value(37.3℃), the alarm will ask him to find the staff.
②Identity authentication:
If your face is in system, the robot will recognize you.
③Health code checking:
The robot can verify green, yellow or red health code.If someone has a yellow or red one, the alarm will ask him to find the staff.
④Mask checking:
If you stand in front of the robot without a mask, the alarm will sound.
⑤Mask selling:
If you don't have a mask, you can scan the QR code to buy one.
⑥Voice interaction:

Tell people the latest data of the COVID-19.You can also ask the robot for more information, such as where is the high risk area and what can we do to protect us from the epidemic.

⑦Access control:

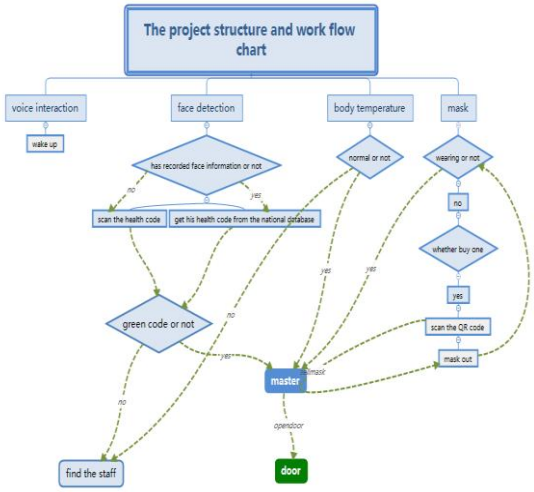
If you want to enter the door, you should have a normal body temperature, a green health code (or a face that robot knows), and wear a mask.Then the robot will open the door for you through Bluetooth.

⑧People following:

If the robot sees you, it will turn the wheels, walk around, turn left and right, head up and down to follow your face.

3.Flow chart

The project structure is as follows:



4.Hardware specification and function introduction

Our robot mainly uses Jupiter robot, on this basis, we add the following hardware

①The open source control board from Labplus which uses the ESP32 chip:

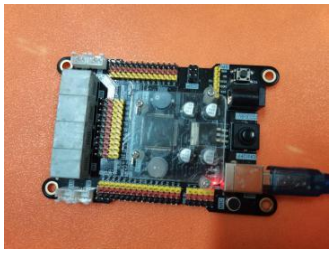
One goes with the robot as a master while the other goes with the door as a slave. The master sends the "opendoor" signal to the slave through Bluetooth.The master is also used to drive the motor to sell the mask.

掌控板V2.0　　拓展板

②A mask-selling box with a motor and a wheel to rotate out the mask.

③Arduino Mega 2560:

Get the value from the temperature sensor through serial port, and send the command to the control board to open the door



5.Technical principles(software)

①Identity authentication and mask checking:

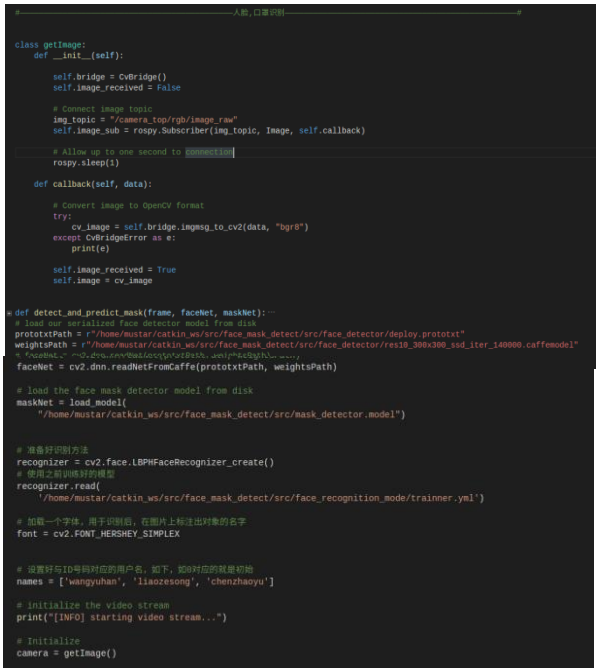
We downloaded face-mask-detection model based on deep learning framework models TensorFlow(to get the face) and Caffe(to detect the mask). We developed these codes and integrated them into the robot.

(from https://www.github.com/balajisrinivas/Face-Mask-Detection)

and face-recognize model based on opencv.

(from https://www.cnblogs.com/dengfaheng/p/10959134.html)

And debugged them

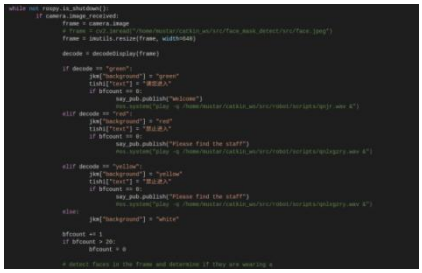


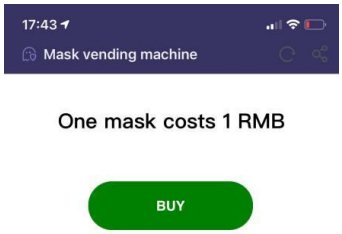
```
class getImage:
    def __init__(self):

        self.bridge = CvBridge()
        self.image_received = False

        # Connect image topic
        img_topic = "/camera_top/rgb/image_raw"
        self.image_sub = rospy.Subscriber(img_topic, Image, self.callback)

        # Allow up to one second to connection
        rospy.sleep(1)

    def callback(self, data):

        # Convert image to OpenCV format
        try:
            cv_image = self.bridge.imgmsg_to_cv2(data, "bgr8")
        except CvBridgeError as e:
            print(e)

        self.image_received = True
        self.image = cv_image

def detect_and_predict_mask(frame, faceNet, maskNet):
  # load our serialized face detector model from disk
  prototxtPath = r"/home/mustar/catkin_ws/src/face_mask_detect/src/face_detector/deploy.prototxt"
  weightsPath = r"/home/mustar/catkin_ws/src/face_mask_detect/src/face_detector/res10_300x300_ssd_iter_140000.caffemodel"
  faceNet = cv2.dnn.readNetFromCaffe(prototxtPath, weightsPath)

  # load the face mask detector model from disk
  maskNet = load_model(
      "/home/mustar/catkin_ws/src/face_mask_detect/src/mask_detector.model")


  # 准备好识别方法
  recognizer = cv2.face.LBPHFaceRecognizer_create()
  # 使用之前训练好的模型
  recognizer.read(
      '/home/mustar/catkin_ws/src/face_mask_detect/src/face_recognition_mode/trainner.yml')

  # 加载一个字体，用于识别后，在图片上标注出对象的名字
  font = cv2.FONT_HERSHEY_SIMPLEX


  # 设置好与ID号码对应的用户名，如下，如0对应的就是初始
  names = ['wangyuhan', 'liaozesong', 'chenzhaoyu']

  # initialize the video stream
  print("[INFO] starting video stream...")

  # Initialize
  camera = getImage()
```

```
        for (box, pred) in zip(locs, preds):
            # unpack the bounding box and predictions
            (startX, startY, endX, endY) = box
            (mask, withoutMask) = pred
            confidence = 30

            gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
            idnum, confidence = recognizer.predict(gray[startY:endY,startX:endX])
            # 计算出一个检验结果
            if confidence < 90:
                name = names[idnum]
                confidence = "{0}%".format(round(100-confidence))
            else:
                name = "unknown"
                confidence = "{0}%".format(round(100-confidence))

            name_str1.set(name)
            #tem_str1.set(tem_str)

            # 输出检验结果以及用户名
            cv2.putText(frame,str(name),(startX,endY+20),font,1,(255,255,255),1)
            cv2.putText(frame,str(confidence),(startX,endY-5),font,0.6,(255,255,255),1)


            # determine the class label and color we'll use to draw
            # the bounding box and text

            if mask > withoutMask:
                label = "Mask"
            else:
                label = "No Mask"
                if nomask_count == 0:
                        say_pub.publish("Please wear a mask!")
                        #os.system("rosrun sound_play say.py 'Please wear a mask!'")
            else:
                label = "No Mask"
                if nomask_count == 0:
                        say_pub.publish("Please wear a mask!")
                        #os.system("rosrun sound_play say.py 'Please wear a mask!'")
                nomask_count += 1
                if nomask_count > 30:
                        nomask_count = 0

            #label = "Mask" if mask > withoutMask else "No Mask"
            color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

            # include the probability in the label
            label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

            # display the label and bounding box rectangle on the output
            # frame
            cv2.putText(frame, label, (startX, startY - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
            cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

        # 转换格式
        cvimage = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
        pilImage = PIL.Image.fromarray(cvimage)  # array:数组
        pilImage = pilImage.resize((870, 620), PIL.Image.ANTIALIAS)  # antialias:最质量
        tkImage = ImageTk.PhotoImage(image=pilImage)

        canvas.create_image(0, 0, anchor='nw', image=tkImage)
    window.update()

window.mainloop()
rospy.spin()
```

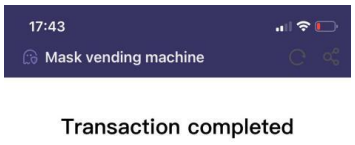
②Health code checking:

Based on pyzbar, the QR code recognition library.





③Mask selling:



Transaction completed

④Voice interaction:

We use sound_play_node from the system itself and the voice package from Iflytek to make the robot answer simple questions.And the open source process snowboy downloaded from github as a waking-up.

```
def xfspeech_callback(data):
    '''main Callback Function'''
    rospy.loginfo(rospy.get_caller_id() + "I heard %s", data.data)
    speech = data.data
    if "How are you" in speech:
        soundhandle.say("Fine,thanks!")
    elif "your name" in speech:
        soundhandle.say("Hi,I'm snowboy")
    elif "can you do" in speech:
        soundhandle.say("I am a robot to serve you.I can help you")
    elif "old" in speech:
        soundhandle.say("I heard you ask about my age. I am five years old.")
    elif "Follow me" in speech:
        playrobotsound()
        soundhandle.say("OK. I will start follow you.")
        control_follow(1)
    elif "Stop follow" in speech:
        playrobotsound()
        soundhandle.say("OK. I will stop follow you.")
        control_follow(0)
    elif "dance" in speech:
        say_pub.publish("You want me to sing and dance? sure. let me show you")
        #soundhandle.playWave("/home/mustar/catkin_ws/src/rc-home-edu-learn-ros/rc
        turn_cmd = Twist()
```

```
def wakeup_callback(data):
    rospy.loginfo(rospy.get_caller_id() + "wakeup_callback: I heard %s", data.data)
    replystr=["hello","yes?","yes,I am here","yes,can i help you"]
    reply=replystr[random.randint(0,len(replystr)-1)]
    say_pub.publish(reply)


if __name__ == '__main__':

    rospy.init_node('robotmain', anonymous=True)
    say_pub = rospy.Publisher('/kws_data', String, queue_size=10)
    rospy.Subscriber("xfspeech", String, xfspeech_callback)
    rospy.Subscriber("xfwakeup", String, wakeup_callback)
    cmd_vel = rospy.Publisher('cmd_vel_mux/input/navi', Twist, queue_size=10)


    print("run snowboy wakeup")
    os.system("/home/mustar/catkin_ws/src/snowboy/examples/C++/wakeup.sh &")
    time.sleep(2)
```

⑤People following:

We calculate out the center point of one’s face in the picture as well as the face area and then turn the wheel according to a function until the point is close to the center point of the screen and the face size is suitable.

```
control_turn=0
if face_x_y_space.x<310:
    x_error=310-face_x_y_space.x
    control_turn=x_error/100.0
    if control_turn  < -0.5:
        control_turn = -0.5
elif face_x_y_space.x>330:
    x_error=330-face_x_y_space.x
    control_turn=x_error/100.0
    if control_turn  > 0.5:
        control_turn =0.5
else:
    control_turn=0


pitch=2.6
if face_x_y_space.y<220:
    y_error=220-face_x_y_space.y
    pitch=round(2.6-y_error*0.0033,2)

elif face_x_y_space.y>250:
    y_error=face_x_y_space.y-250
    pitch=round(2.6+y_error*0.0033,2)
```

```
control_speed = 0.0
if face_x_y_space.space < 15000:
    space_error = 15000 - face_x_y_space.space
    control_speed = space_error*0.00005
    if control_speed  > 0.3:
        control_speed =0.3
elif face_x_y_space.space > 20000:
    space_error = 20000 - face_x_y_space.space
    control_speed = space_error*0.00005
    if control_speed  < -0.3:
        control_speed = -0.3
```

⑥@ Laser mapping and Navigation