Project Development Phase

Sprint -2

Date	13 November 2022
Team ID	PNT2022TMID34928
Project Name	IOT BASED SMART CROP PROTECTION SYSTEM

Python Code:

```
 import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import cv2
import numpy as np
#Provide your IBM Watson Device Credentials
organization = "2ldaf5"
deviceType1 = "Sensor"
deviceId1 = "DHT"
authMethod = "token"
authToken1 = "NeVIAy2K16H)d9sXvz"

deviceType2 = "Sensor1"
deviceId2 = "Soil_moisture"
authToken2= "zwr247qk1Xca0w?QEs"

deviceType3 = "Actuator"
deviceId3 = "Water_pump"
authToken3= "Pze?D!@FjZeAtfMB4q"
```

```
deviceType4= "Sensor2"

deviceId4= "PIR"

authToken4= "i-yXXf?FnB011nEycG"


deviceType5= "Sensor3"

deviceId5="Ultrasonic"

authToken5="e&QzDxiHpQ4GaRyPGJ"


deviceType6="Detector"

deviceId6="Camera"

authToken6="f7LMx6-a(uhdnDcKa-"


deviceType7="Output"

deviceId7="LED"

authToken7="qJIBVJHP9@Ihl8@CK3"

# Initialize GPIO

def myCommandCallback(cmd):

   print("Command received: %s \n" % cmd.data['command'])

   status=cmd.data['command']

   if status=="Waterpump_on":

      print ("Water Pump is Turned ON \n")

   else :

      print ("Water Pump is Turned OFF \n")

def myCommandCallback1(cmd):

   print("Command received: %s \n" % cmd.data['command1'])

   status=cmd.data['command1']

   if status=="LEDlight_on":

      print ("LED is Turned ON \n")

   else :

      print ("LED is Turned OFF \n")

def cam():
   net=cv2.dnn.readNet('C:/Users/hp/OneDrive/Desktop/opencvtrial/yolov3.weights','C:/Users/hp
/OneDrive/Desktop/opencvtrial/yolov3.cfg.txt')
 classes=[]
 with open('C:/Users/hp/OneDrive/Desktop/opencvtrial/coco.names','r') as f:
    classes=f.read().splitlines()
 cap=cv2.VideoCapture('blackbear.mp4')
 for i in range(100):
      _,img=cap.read()
      height,width,_=img.shape
      blob=cv2.dnn.blobFromImage(img,1/255,(416,416),(0,0,0),swapRB=True,crop=False)
#(img,reduction the pixels size,size of the image,rgb colour)
      net.setInput(blob)
      output_layers_names=net.getUnconnectedOutLayersNames()
      layeroutput=net.forward(output_layers_names)
      boxes=[]
      confidences=[]
      class_ids=[]
      for output in layeroutput:
          for detection in output:
              scores=detection[5:]
              class_id=np.argmax(scores)
              confidence=scores[class_id]
              if confidence > 0.5:
                  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)))
```

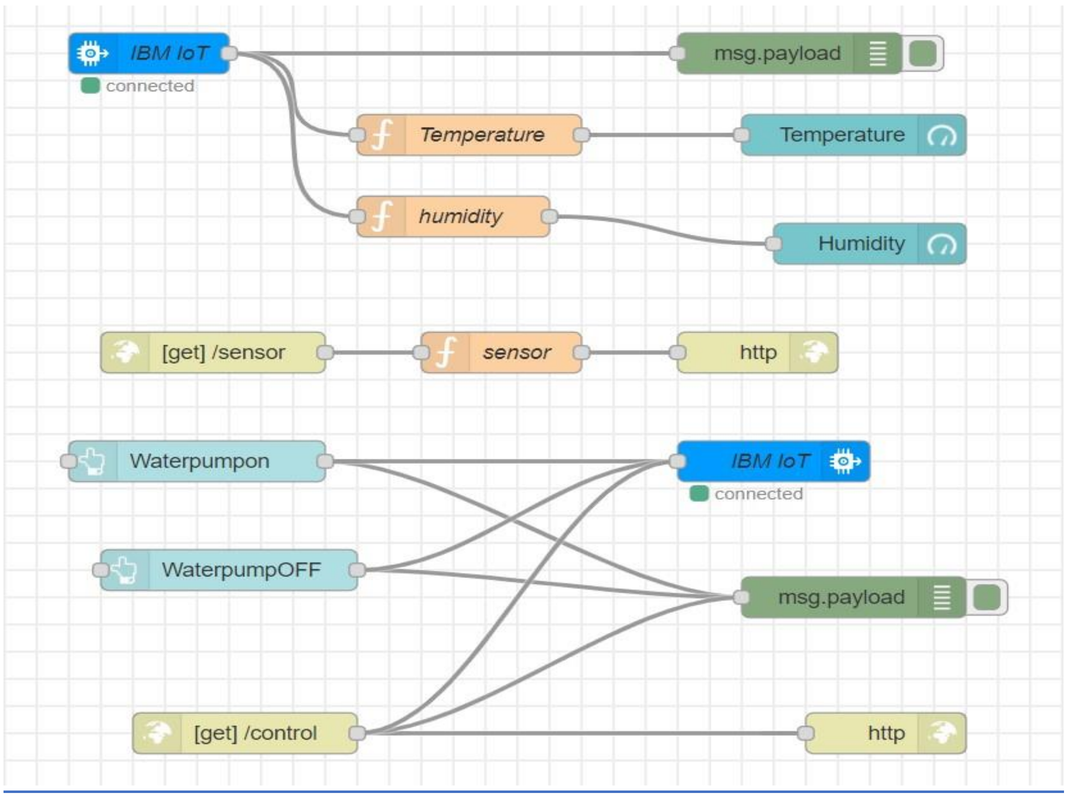
```
                class_ids.append(class_id)

                animal=classes[class_id]

        indexes=cv2.dnn.NMSBoxes(boxes,confidences,0.5,0.4)

        font=cv2.FONT_HERSHEY_COMPLEX

        colors=np.random.uniform(0,255,size=(len(boxes),3))

        for i in indexes.flatten():

          x,y,w,h=boxes[i]

          label=str(classes[class_ids[i]])

          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+20),font,2,(255,255,0),2)

        cv2.imshow('Target Image',img)

        key=cv2.waitKey(1)

        if key ==ord('q'):

          break

  print("Alert! detected animal is "+str(animal))

  print("LED turned ON")

  cap.release()

  cv2.destroyAllWindows()

  data6 = { 'Intruded_Animal' :  str(animal) }

  def myOnPublishCallback6():

    print ("Published Intruded Animal is "+str(animal) ,"to IBM Watson")

  success6 = deviceCli6.publishEvent("Allert", "json", data6, qos=0,
on_publish=myOnPublishCallback6)

  if not success6:

    print("Not connected to IoTF")

  time.sleep(1)

try:

   deviceOptions1 = {"org": organization, "type": deviceType1, "id": deviceId1, "auth-method":
authMethod, "auth-token": authToken1}

   deviceCli1 = ibmiotf.device.Client(deviceOptions1)
```

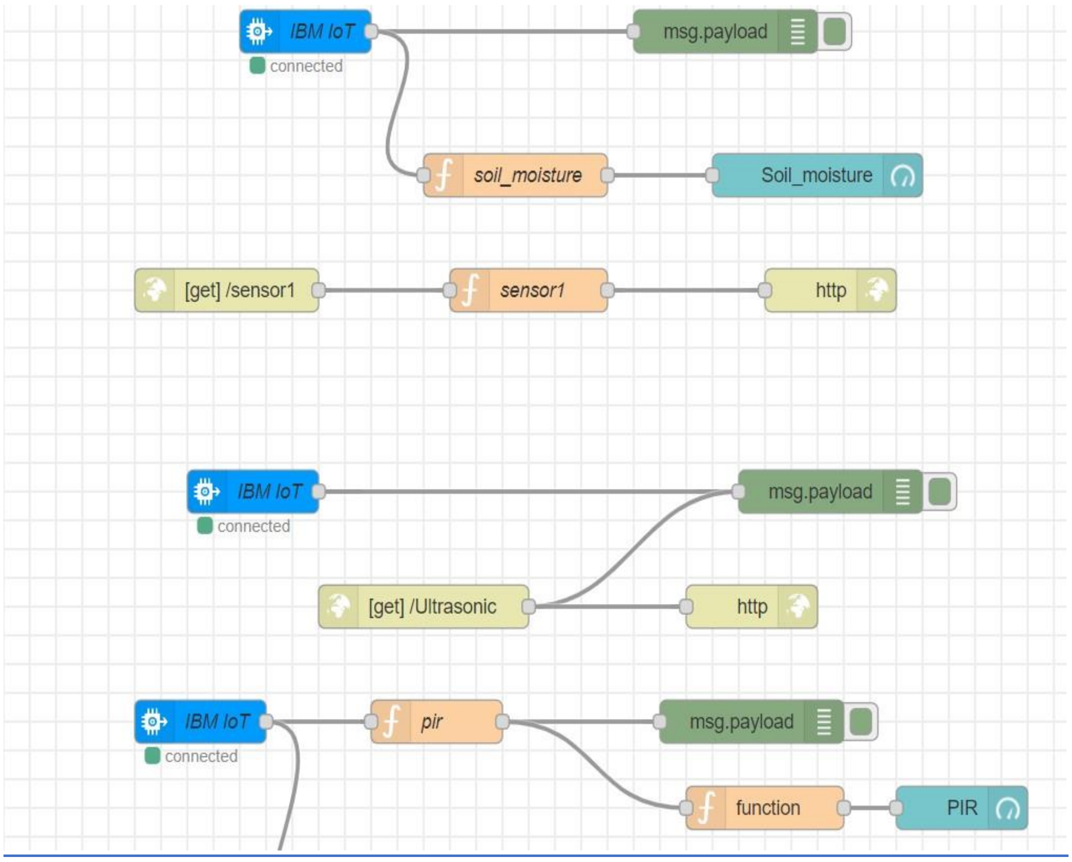
```
    #...........................................

    deviceOptions2 = {"org": organization, "type": deviceType2, "id": deviceId2, "auth-method":
authMethod, "auth-token": authToken2}

    deviceCli2 = ibmiotf.device.Client(deviceOptions2)

    #...........................................

    deviceOptions3 = {"org": organization, "type": deviceType3, "id": deviceId3, "auth-method":
authMethod, "auth-token": authToken3}

    deviceCli3 = ibmiotf.device.Client(deviceOptions3)

    #...........................................

    deviceOptions4 = {"org": organization, "type": deviceType4, "id": deviceId4, "auth-method":
authMethod, "auth-token": authToken4}

    deviceCli4 = ibmiotf.device.Client(deviceOptions4)

    #...........................................

    deviceOptions5 = {"org": organization, "type": deviceType5, "id": deviceId5, "auth-method":
authMethod, "auth-token": authToken5}

    deviceCli5 = ibmiotf.device.Client(deviceOptions5)

    #...........................................

    deviceOptions6 = {"org": organization, "type": deviceType6, "id": deviceId6, "auth-method":
authMethod, "auth-token": authToken6}

    deviceCli6 = ibmiotf.device.Client(deviceOptions6)

    #...........................................

    deviceOptions7 = {"org": organization, "type": deviceType7, "id": deviceId7, "auth-method":
authMethod, "auth-token": authToken7}

    deviceCli7 = ibmiotf.device.Client(deviceOptions7)

    #...........................................

except Exception as e:

            print("Caught exception connecting device: %s" % str(e))

            sys.exit()

deviceCli1.connect()

deviceCli2.connect()

deviceCli3.connect()

deviceCli4.connect()

deviceCli5.connect()
```

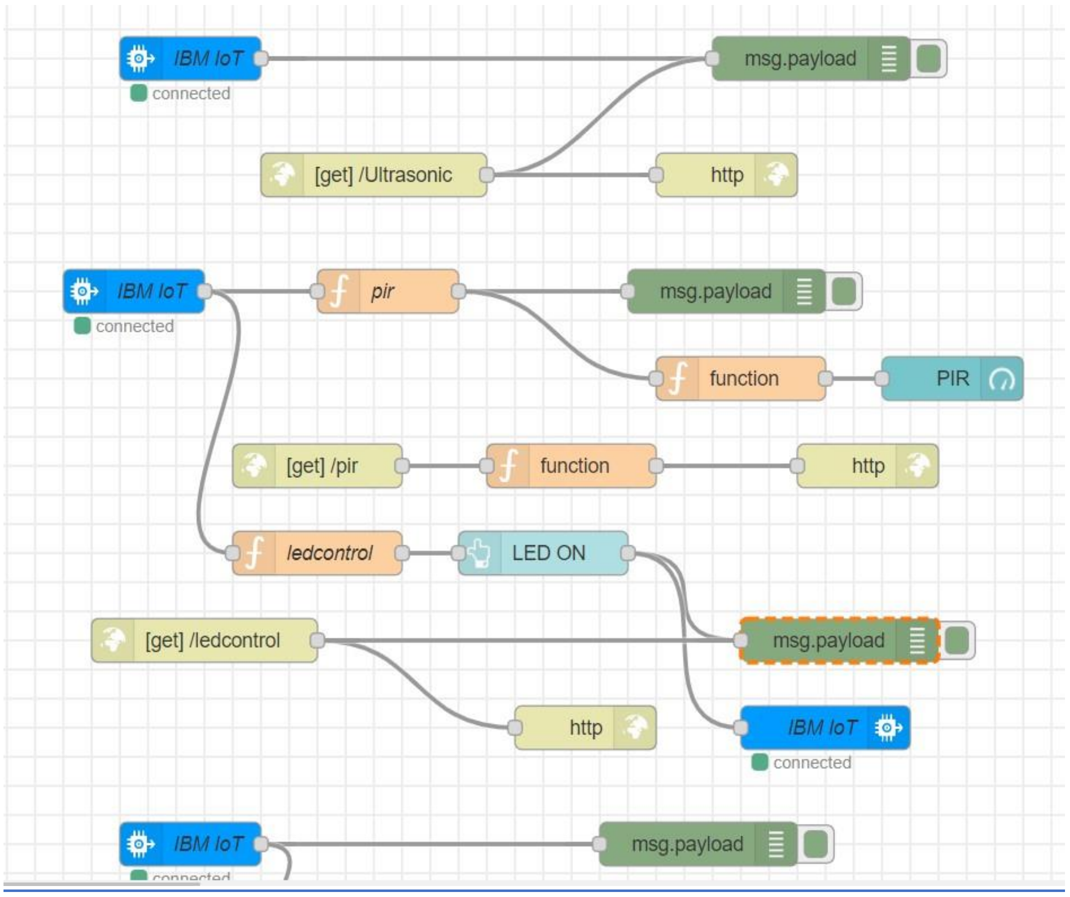
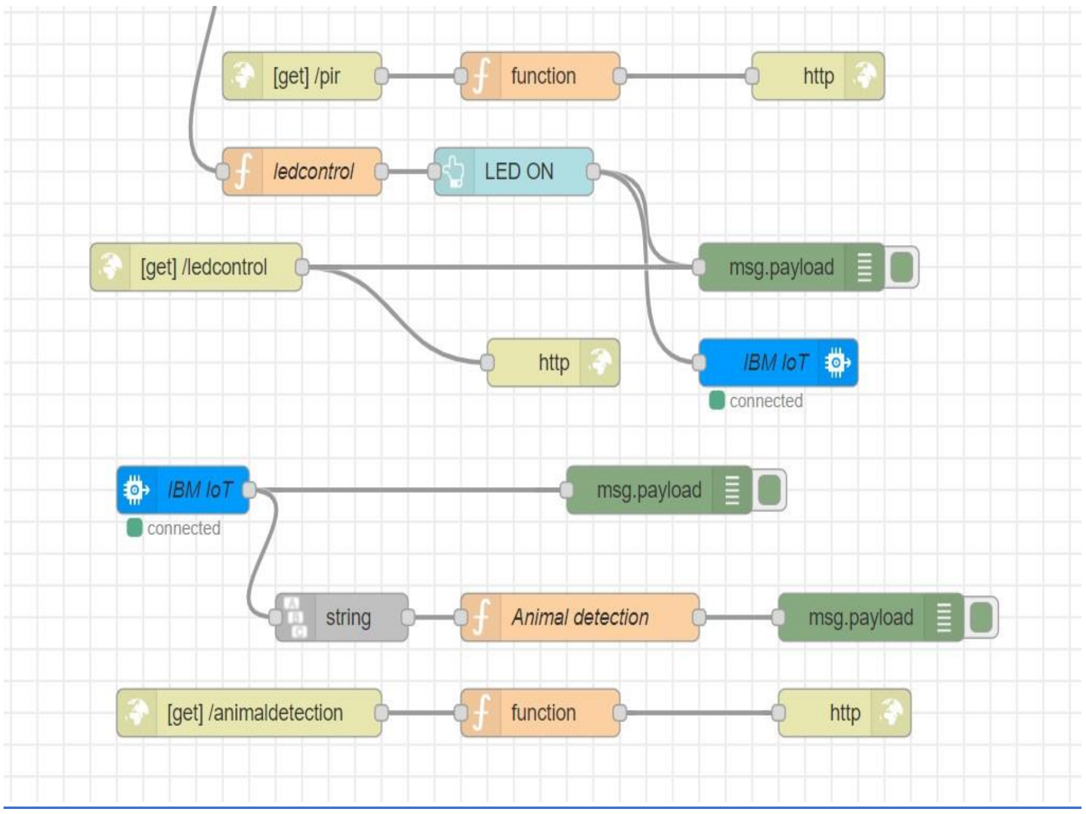
```
deviceCli6.connect()

deviceCli7.connect()

while (True):

       #Get Sensor Data from DHT11

        temp=random.randint(0,45)

       Humid=random.randint(0,100)

       data1 = { 'Temperature' : temp , 'Humidity': Humid}

       def myOnPublishCallback1():

          print ("Published Temperature   =  %s C" % temp, "Humidity   =  %s %%" % Humid, "to IBM
Watson \n")

       success1 = deviceCli1.publishEvent("DHT Sensor", "json", data1, qos=0,
on_publish=myOnPublishCallback1)

       if not success1:

          print("Not connected to IoTF\n")

       time.sleep(1)


       #Get Sensor Data from SOIL Moisture

       Soil_moisture=random.randint(0,100)

       data2 = { 'Soil_moisture' : Soil_moisture}

       def myOnPublishCallback2():

          print ("Published Soil_moisture =  %s %%" % Soil_moisture, "to IBM Watson")

       success2 = deviceCli2.publishEvent("Soil Moisture Sensor", "json", data2, qos=0,
on_publish=myOnPublishCallback2)

       if not success2:

          print("Not connected to IoTF")

       time.sleep(1)

       #Automatically turning on/off water pump

       if Soil_moisture <= 20:

         print("Water pump is turned on")

      deviceCli3.commandCallback = myCommandCallback

      #Get Sensor Data from PIR

       pir=random.randint(0,1)
```

```
        data4 = { 'PIR_output' : pir}
        #Movement detection by PIR
        if(pir==1):
          print("movement is detected\nultrasonic sensor turned ON\n")
          def myOnPublishCallback4():
            print ("Published PIR output    =  %s " % pir, "to IBM Watson")
          success4 = deviceCli4.publishEvent("PIR Sensor", "json", data4, qos=0,
on_publish=myOnPublishCallback4)
          if not success4:
            print("Not connected to IoTF")
          time.sleep(1)
          #Get Sensor dat from ultrasonic sensor
          distance=random.randint(0,500)
          data5 = { 'Distance:' : distance}
          def myOnPublishCallback5():
            print ("Published distance  =  %s m" % distance, "to IBM Watson")
          success5 = deviceCli5.publishEvent("Ultrasonic Sensor", "json", data5, qos=0,
on_publish=myOnPublishCallback5)
          if not success5:
            print("Not connected to IoTF")
          time.sleep(1)
          #turning on camera
          if(distance<=200):
            print("camera turned on\n")
            cam()
        deviceCli7.commandCallback = myCommandCallback1
 # cv2.waitKey(0)
deviceCli1.disconnect()
deviceCli2.disconnect()
deviceCli4.disconnect()
deviceCli5.disconnect()
deviceCli6.disconnect()
```

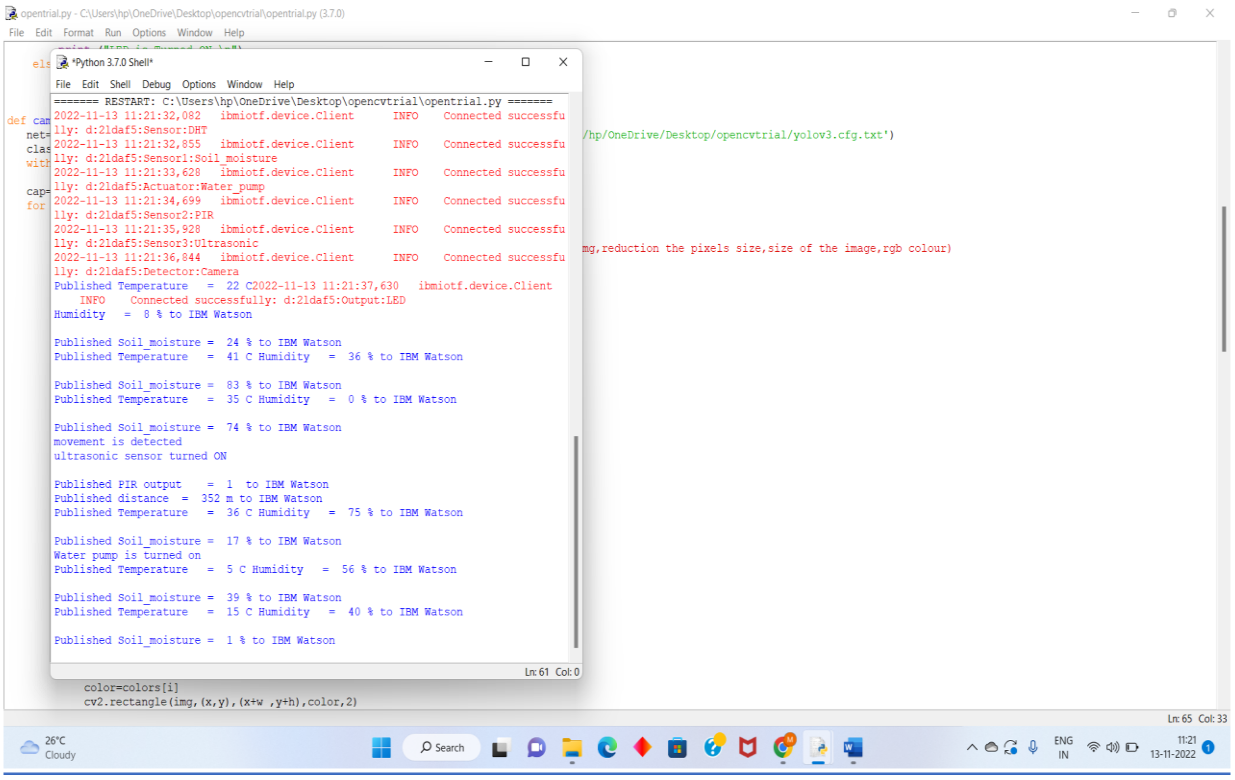
Node Red Flow:



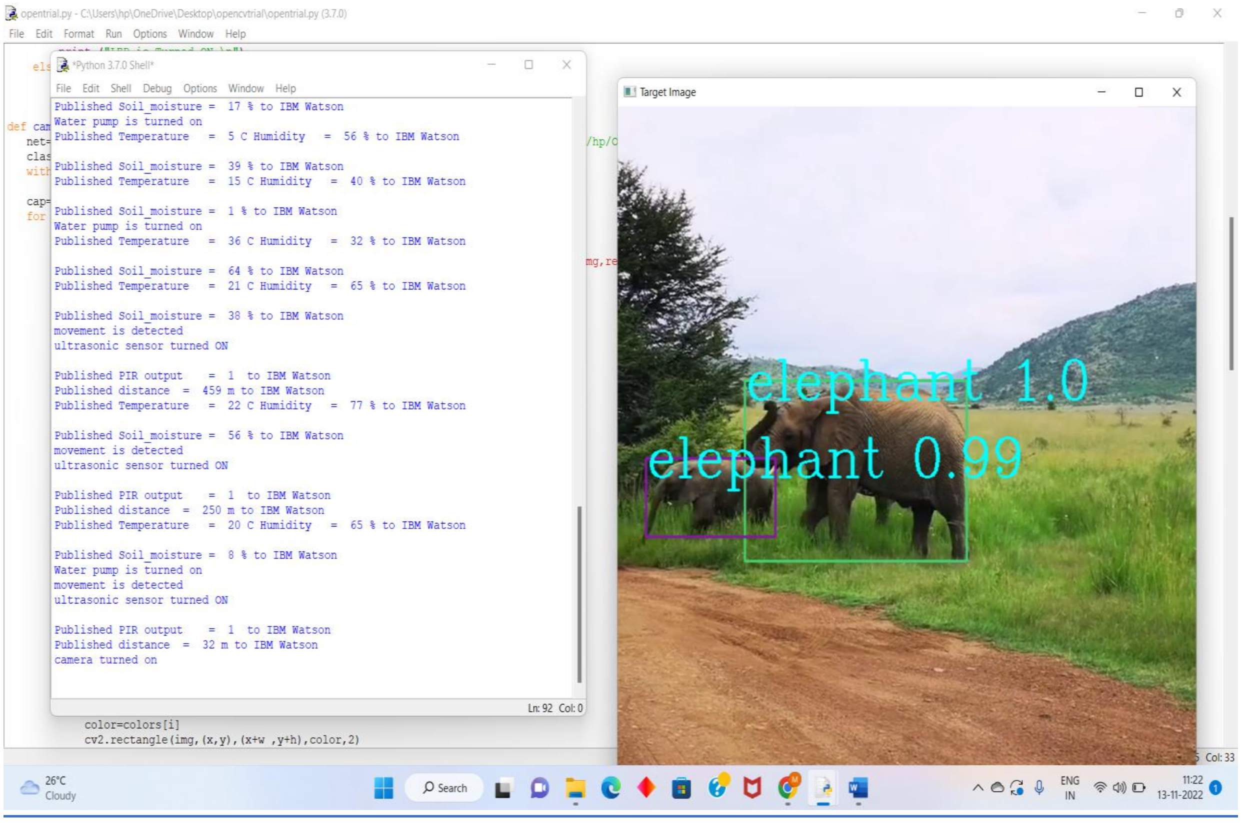



IBM IoT
connected
msg.payload
[get] /Ultrasonic
http
IBM IoT
connected
pir
msg.payload
function
PIR
[get] /pir
function
http
ledcontrol
LED ON
[get] /ledcontrol
msg.payload
http
IBM IoT
connected
IBM IoT
msg.payload

[get] /pir
function
http
ledcontrol
LED ON
[get] /ledcontrol
msg.payload
http
IBM IoT
connected
IBM IoT
connected
msg.payload
string
Animal detection
msg.payload
[get] /animaldetection
function
http

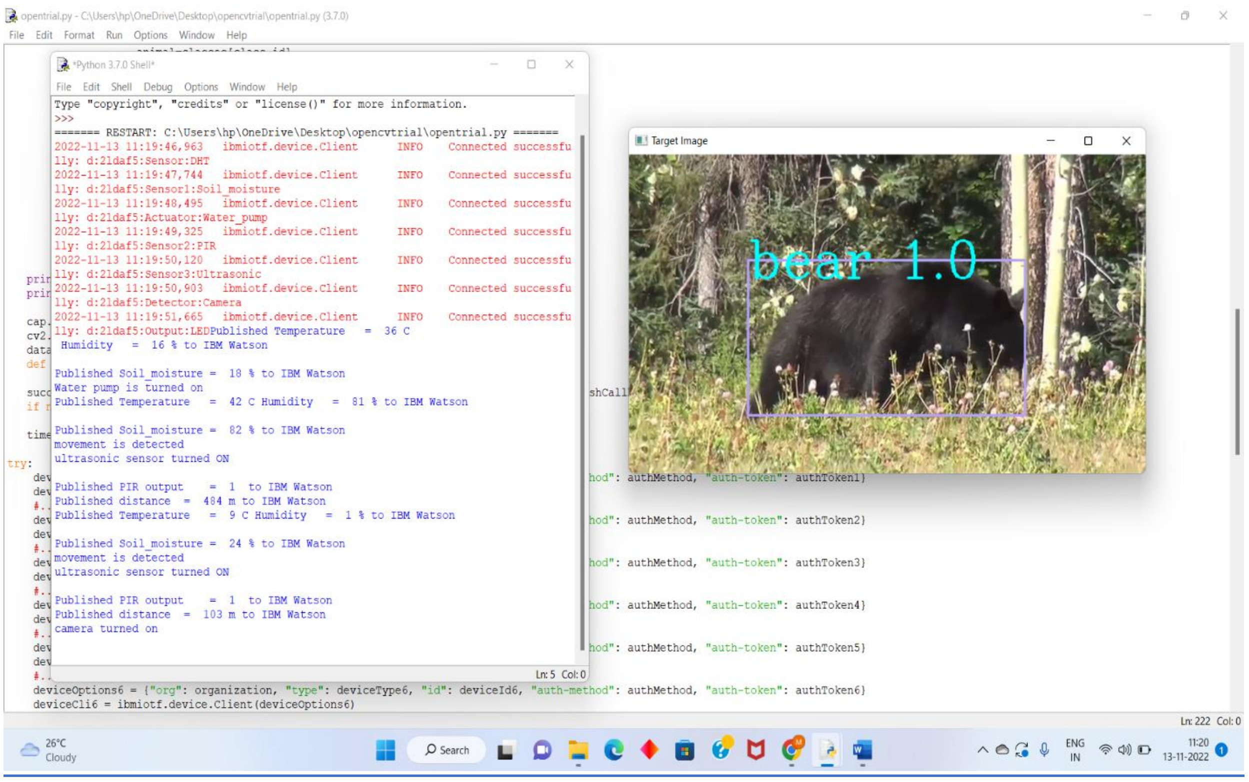
Python Output:



```
opentrial.py - C:\Users\hp\OneDrive\Desktop\opencvtrial\opentrial.py (3.7.0)

*Python 3.7.0 Shell*

======= RESTART: C:\Users\hp\OneDrive\Desktop\opencvtrial\opentrial.py =======
2022-11-13 11:21:32,082   ibmiotf.device.Client      INFO    Connected successfu
lly: d:21daf5:Sensor:DHT
2022-11-13 11:21:32,855   ibmiotf.device.Client      INFO    Connected successfu
lly: d:21daf5:Sensor1:Soil_moisture
2022-11-13 11:21:33,628   ibmiotf.device.Client      INFO    Connected successfu
lly: d:21daf5:Actuator:Water_pump
2022-11-13 11:21:34,699   ibmiotf.device.Client      INFO    Connected successfu
lly: d:21daf5:Sensor2:PIR
2022-11-13 11:21:35,928   ibmiotf.device.Client      INFO    Connected successfu
lly: d:21daf5:Sensor3:Ultrasonic
2022-11-13 11:21:36,844   ibmiotf.device.Client      INFO    Connected successfu
lly: d:21daf5:Detector:Camera
Published Temperature   =  22 C2022-11-13 11:21:37,630   ibmiotf.device.Client
    INFO    Connected successfully: d:21daf5:Output:LED
Humidity   =  8 % to IBM Watson

Published Soil_moisture =  24 % to IBM Watson
Published Temperature   =  41 C Humidity   =  36 % to IBM Watson

Published Soil_moisture =  83 % to IBM Watson
Published Temperature   =  35 C Humidity   =  0 % to IBM Watson

Published Soil_moisture =  74 % to IBM Watson
movement is detected
ultrasonic sensor turned ON

Published PIR output    =  1  to IBM Watson
Published distance  =  352 m to IBM Watson
Published Temperature   =  36 C Humidity   =  75 % to IBM Watson

Published Soil_moisture =  17 % to IBM Watson
Water pump is turned on
Published Temperature   =  5 C Humidity   =  56 % to IBM Watson

Published Soil_moisture =  39 % to IBM Watson
Published Temperature   =  15 C Humidity   =  40 % to IBM Watson

Published Soil_moisture =  1 % to IBM Watson

Ln: 61  Col: 0

/hp/OneDrive/Desktop/opencvtrial/yolov3.cfg.txt')
mg,reduction the pixels size,size of the image,rgb colour)

        color=colors[i]
        cv2.rectangle(img,(x,y),(x+w ,y+h),color,2)

Ln: 65  Col: 33
26°C
Cloudy
11:21
13-11-2022
```



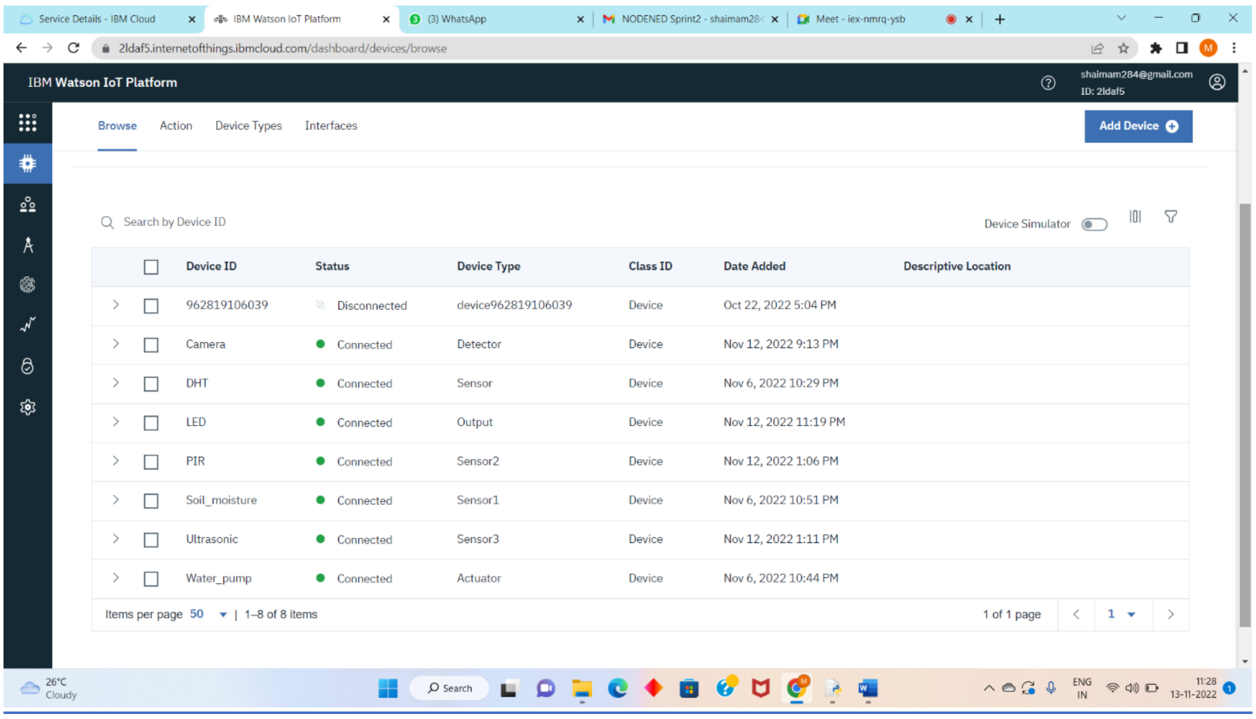
```
opentrial.py - C:\Users\hp\OneDrive\Desktop\opencvtrial\opentrial.py (3.7.0)

*Python 3.7.0 Shell*

Published Soil_moisture =  17 % to IBM Watson
Water pump is turned on
Published Temperature   =  5 C Humidity   =  56 % to IBM Watson

Published Soil_moisture =  39 % to IBM Watson
Published Temperature   =  15 C Humidity   =  40 % to IBM Watson

Published Soil_moisture =  1 % to IBM Watson
Water pump is turned on
Published Temperature   =  36 C Humidity   =  32 % to IBM Watson

Published Soil_moisture =  64 % to IBM Watson
Published Temperature   =  21 C Humidity   =  65 % to IBM Watson

Published Soil_moisture =  38 % to IBM Watson
movement is detected
ultrasonic sensor turned ON

Published PIR output    =  1  to IBM Watson
Published distance  =  459 m to IBM Watson
Published Temperature   =  22 C Humidity   =  77 % to IBM Watson

Published Soil_moisture =  56 % to IBM Watson
movement is detected
ultrasonic sensor turned ON

Published PIR output    =  1  to IBM Watson
Published distance  =  250 m to IBM Watson
Published Temperature   =  20 C Humidity   =  65 % to IBM Watson

Published Soil_moisture =  8 % to IBM Watson
Water pump is turned on
movement is detected
ultrasonic sensor turned ON

Published PIR output    =  1  to IBM Watson
Published distance  =  32 m to IBM Watson
camera turned on

Ln: 92  Col: 0

        color=colors[i]
        cv2.rectangle(img,(x,y),(x+w ,y+h),color,2)

Col: 33
26°C
Cloudy
11:22
13-11-2022
```


opentrial.py - C:\Users\hp\OneDrive\Desktop\opencvtrial\opentrial.py (3.7.0)
File Edit Format Run Options Window Help
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Target Image
bear 1.0
Ln: 5 Col: 0
Ln: 222 Col: 0
26°C
Cloudy
Search
ENG
IN
11:20
13-11-2022

```
Type "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:\Users\hp\OneDrive\Desktop\opencvtrial\opentrial.py =======
2022-11-13 11:19:46,963   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Sensor:DHT
2022-11-13 11:19:47,744   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Sensor1:Soil_moisture
2022-11-13 11:19:48,495   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Actuator:Water_pump
2022-11-13 11:19:49,325   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Sensor2:PIR
2022-11-13 11:19:50,120   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Sensor3:Ultrasonic
2022-11-13 11:19:50,903   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Detector:Camera
2022-11-13 11:19:51,665   ibmiotf.device.Client      INFO    Connected successfu
lly: d:2ldaf5:Output:LEDPublished Temperature   =  36 C
 Humidity   =  16 % to IBM Watson

Published Soil_moisture =  18 % to IBM Watson
Water pump is turned on
Published Temperature   =  42 C Humidity   =  81 % to IBM Watson

Published Soil_moisture =  82 % to IBM Watson
movement is detected
ultrasonic sensor turned ON

Published PIR output    =  1  to IBM Watson
Published distance  =  484 m to IBM Watson
Published Temperature   =  9 C Humidity   =  1 % to IBM Watson

Published Soil_moisture =  24 % to IBM Watson
movement is detected
ultrasonic sensor turned ON

Published PIR output    =  1  to IBM Watson
Published distance  =  103 m to IBM Watson
camera turned on
```

```
hod": authMethod, "auth-token": authToken1}
hod": authMethod, "auth-token": authToken2}
hod": authMethod, "auth-token": authToken3}
hod": authMethod, "auth-token": authToken4}
hod": authMethod, "auth-token": authToken5}
    deviceOptions6 = {"org": organization, "type": deviceType6, "id": deviceId6, "auth-method": authMethod, "auth-token": authToken6}
    deviceCli6 = ibmiotf.device.Client(deviceOptions6)
```

IBM Watson Screen Shot :



Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
962819106039	Disconnected	device962819106039	Device	Oct 22, 2022 5:04 PM	
Camera	● Connected	Detector	Device	Nov 12, 2022 9:13 PM	
DHT	● Connected	Sensor	Device	Nov 6, 2022 10:29 PM	
LED	● Connected	Output	Device	Nov 12, 2022 11:19 PM	
PIR	● Connected	Sensor2	Device	Nov 12, 2022 1:06 PM	
Soil_moisture	● Connected	Sensor1	Device	Nov 6, 2022 10:51 PM	
Ultrasonic	● Connected	Sensor3	Device	Nov 12, 2022 1:11 PM	
Water_pump	● Connected	Actuator	Device	Nov 6, 2022 10:44 PM	

Items per page 50 | 1–8 of 8 items 1 of 1 page 1