ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

——————— * ———————

PHÁT TRIỂN ỨNG DỤNG INTERNET OF THINGS THỰC HÀNH
(CO3038)

LAB 02 TÍCH HỢP MÔ HÌNH HỌC MÁY VÀO HỆ THỐNG IOT VÀ HIỆN THỰC ĐỌC DỮ LIỆU TỪ PHẦN CỨNG

BỘ MÔN: KỸ THUẬT MÁY TÍNH
GVHD: VŨ TRỌNG THIÊN
NHÓM LỚP: L01
—o0o—
SVTH: MAI THỊNH PHÁT (1914590)

TP. HỒ CHÍ MINH, 2/2024

Mục lục

Danh sách hình vẽ

Chương 1

Tổng quan

1.1 Giới thiệu về Teachable Machine

Teachable Machine là một công cụ trực tuyến do Google phát triển, cho phép người dùng dễ dàng huấn luyện các mô hình máy học. Được thiết kế để làm cho máy học trở nên dễ tiếp cận hơn đối với một đối tượng người dùng rộng lớn hơn bằng cách cung cấp một giao diện thân thiện với người dùng.

Với Teachable Machine, người dùng có thể huấn luyện các mô hình cho việc phân loại hình ảnh, phân loại âm thanh và nhận diện tư thế. Quy trình thường bao gồm ba bước chính:

1. Thu thập Dữ liệu: Người dùng cung cấp ví dụ cho các lớp khác nhau bằng cách chụp ảnh trực tiếp hoặc upload hình ảnh có sẵn lên hệ thống.

2. Huấn luyện Mô hình: Teachable Machine sử dụng một mô hình đã được huấn luyện trước làm điểm khởi đầu và tinh chỉnh nó dựa trên các ví dụ được cung cấp. Quá trình này được thực hiện trên thiết bị của người dùng, giúp tăng tốc quá trình và bảo vệ quyền riêng tư.

3. Kiểm thử và Sử dụng Mô hình: Sau khi huấn luyện, người dùng có thể kiểm thử mô hình trong giao diện Teachable Machine. Mô hình đã được huấn luyện có thể được xuất và tích hợp vào các ứng dụng, trang web hoặc dự án khác.

Công cụ này mang tính giáo dục và là một sự giới thiệu về các khái niệm máy học cho người mới học. Nó giúp giảm bớt sự phức tạp trong quá trình huấn luyện mô hình và thể hiện ứng dụng tiềm năng của máy học một cách thực tế. Tuy nhiên, để thực hiện các dự án máy học phức tạp và tùy chỉnh hơn, các nhà phát triển thường chuyển sang các framework và công cụ máy học cao cấp hơn.

1.2 Cấu trúc bài báo cáo

Kết của của bài thực hành này sẽ được trình bày trong báo cáo như sau:

- **Chương 1 Tổng quan**: Giới thiệu về Teachable Machine
- **Chương 2 Huấn luyện mô hình**: Trình bày cách huấn luyện và tích hợp mô hình vào ứng dụng IOT.
- **Chương 3 Phân tích và thiết kế hệ thống:** Tìm hiểu các hệ thống liên quan, rút ra ưu nhược của nó , đồng thời trình bày về các phân tích và thiết kế hệ thống đã sử dụng trong đề tài này.

- **Chương 4 Bài toán và giải pháp:** Trình bày các vấn đề , bài toán được đặt ra đối với đề tài và cách giải quyết chúng.
- **Chương 5 Hiện thực hệ thống:** Trình bày sơ lược kết quả của quá trình hiện thực hệ thống.
- **Chương 6 Kiểm thử hệ thống:** Trình bày về phương pháp kiểm thử đã áp dụng và kết quả thu được.
- **Chương 7 Tổng kết:** Tóm tắt các kết quả đạt được, hạn chế còn tồn đọng và hướng phát triển hệ thống trong tương lai.

Chương 2

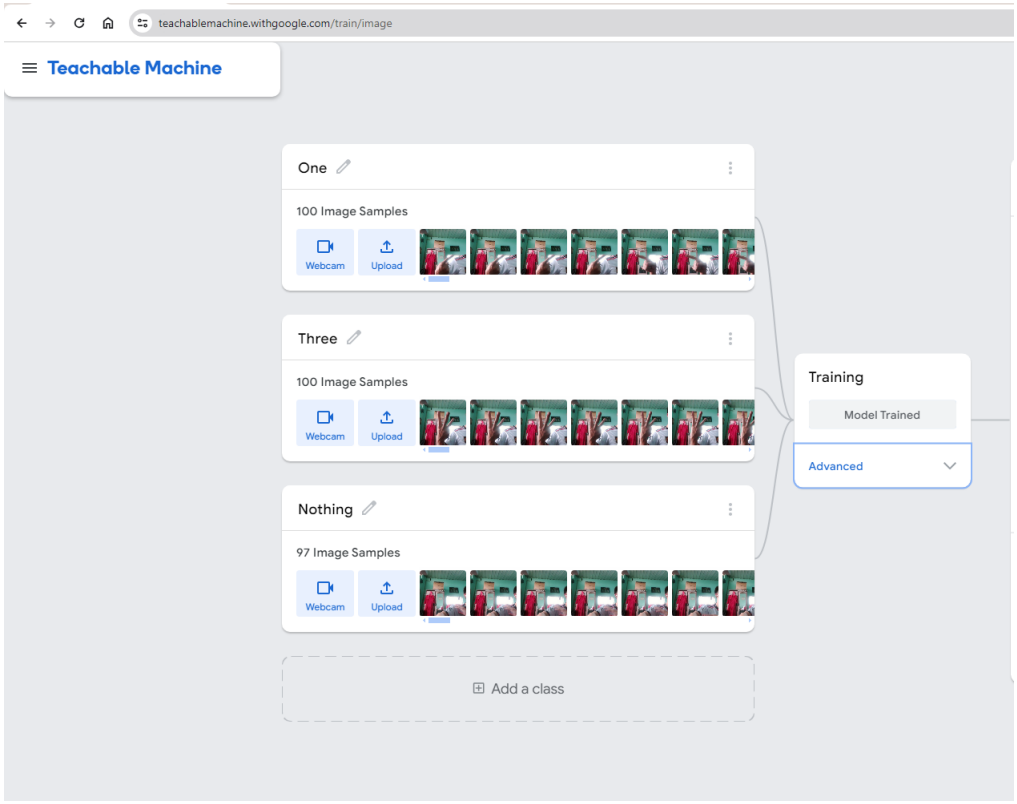
Huấn luyện mô hình

Trong phần này em sẽ trình bày quá trình huấn luyện mô hình trí tuệ nhân tạo được sử dụng trong bài thực hành này. Cụ thể là mô hình sẽ được huấn luyện để có thể nhận biết con số muốn biểu thị dựa vào ký hiệu ngón tay. Trong phần này em sẽ đi qua ba bước chính chính như sau:

1. **Chuẩn bị dữ liệu**: Dữ liệu sẽ được chụp trực tiếp từ camera cho các lớp tương ứng.
2. **Huấn luyện**: Lựa chọn các siêu tham số phù hợp cho mô hình và tiến hành huấn luyện.
3. **Kiểm tra lại mô hình**: Sử dụng webcam và đánh giá mô hình trực tiếp trên ứng dụng Teachable Machine.

2.1 Chuẩn bị dữ liệu

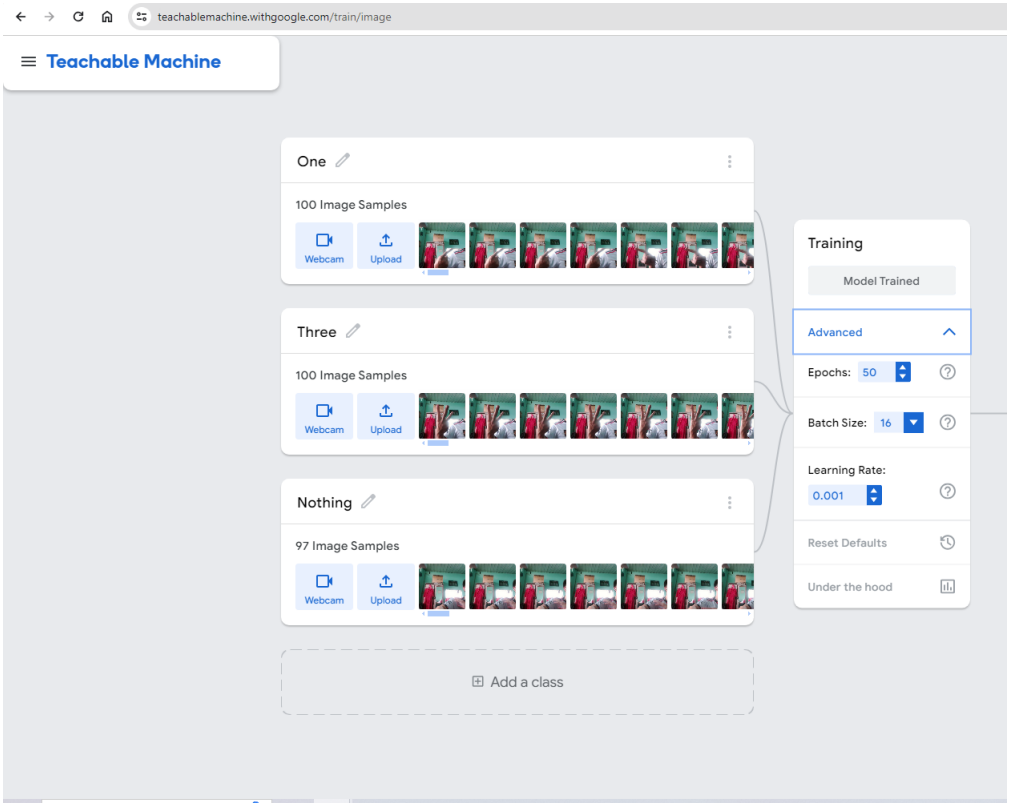
Trong mô hình được xây dựng sẽ phân loại hai lớp đó là lớp **One** (biểu thị cho dơ một ngón tay), lớp **Three** (biểu thị cho dơ ba ngón tay) và lớp Nothing (biểu thị cho không dơ ngón nay nào). Để nạp dữ liệu chúng ta có thể sử dụng camera kết nối với máy tính và chụp ảnh trực tiếp trên **Teachable Machine** hoặc là tải ảnh có sẵn tương ứng lên các lớp



Hình 2.1: Chuẩn bị dữ liệu

2.2 Điều chỉnh siêu tham số và tiến hành huấn luyện mô hình

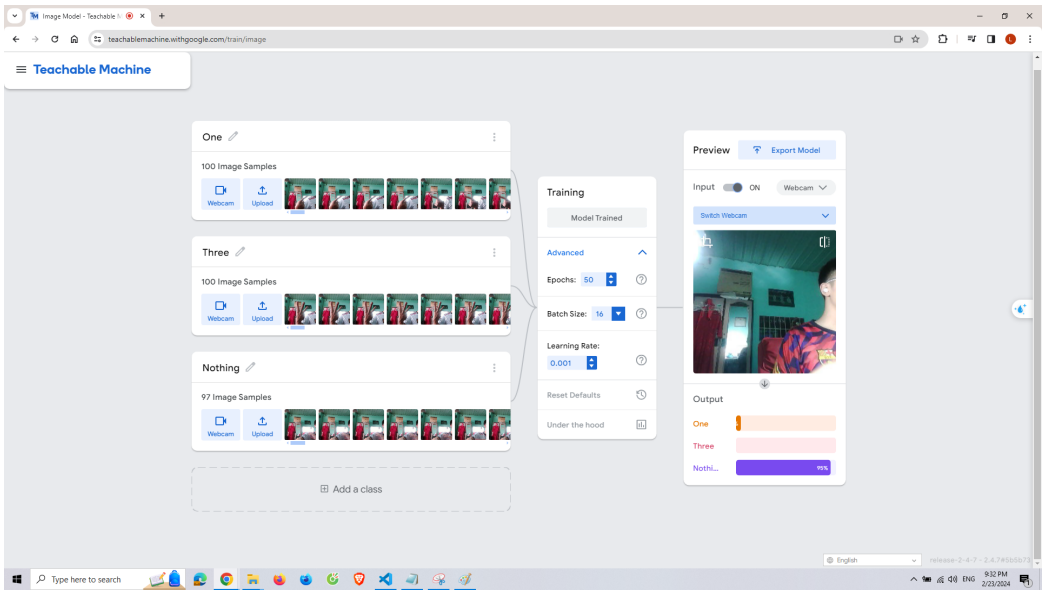
Sau khi đã chuẩn bị xong dữ liệu chúng ta sẽ điều chỉnh các siêu tham số cho mô hình. Sau đó ta bấn chọn **train model** để tiến hành huấn luyện mô hình.



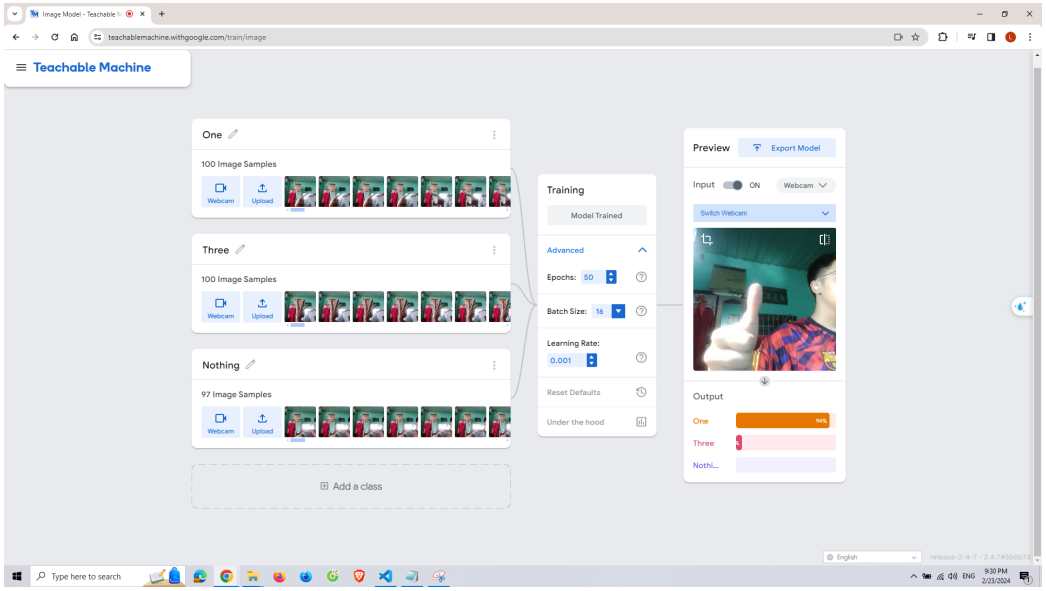
Hình 2.2: Điều chỉnh siêu tham số và tiến hành huấn luyện mô hình

2.3 Tiến hành đánh giá kết quả mô hình

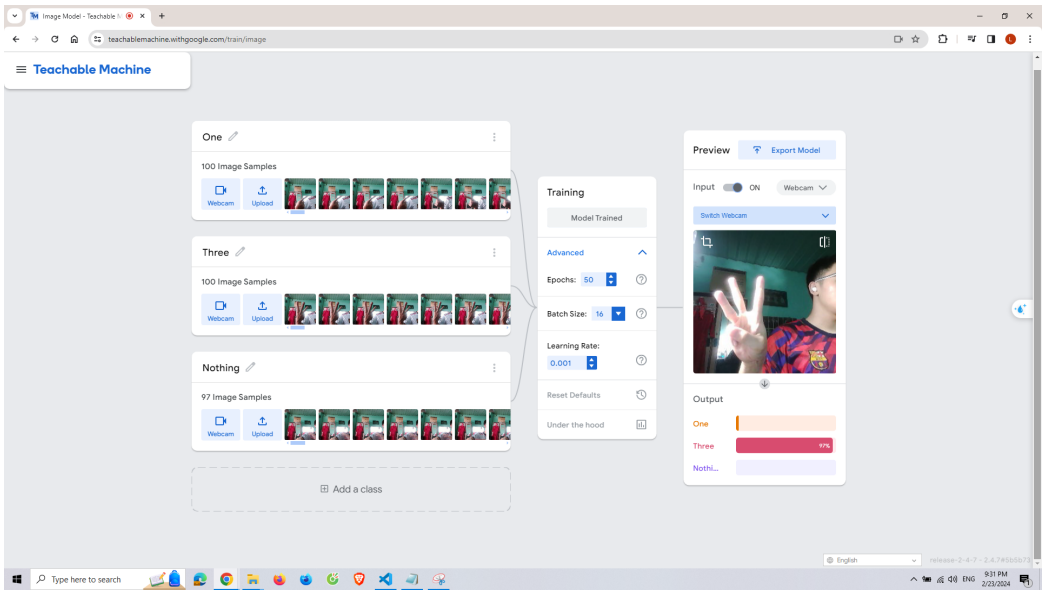
Sau khi đã huấn luyện thành công mô hình chúng ta sẽ đánh giá trực tiếp mô hình trên teachable.



Hình 2.3: khi không dơ ngón tay nào



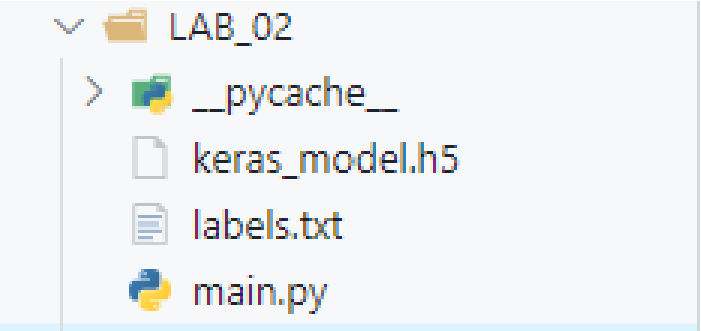
Hình 2.4: khi dơ 1 ngón tay nào



Hình 2.5: khi dơ 3 ngón tay nào

2.4 Tích hợp mô hình đã huấn luyện ở trên vào chương trình

Sau khi đã huấn luyện và kiểm thử mô hình, chúng ta sẽ tích hợp mô hình vào chương trình.



Hình 2.6: Tải mô hình AI về máy

đoạn code trong file *simple_ai.py* sau đây sẽ mô tả các bước tích hợp các hàm sửa lý về AI:

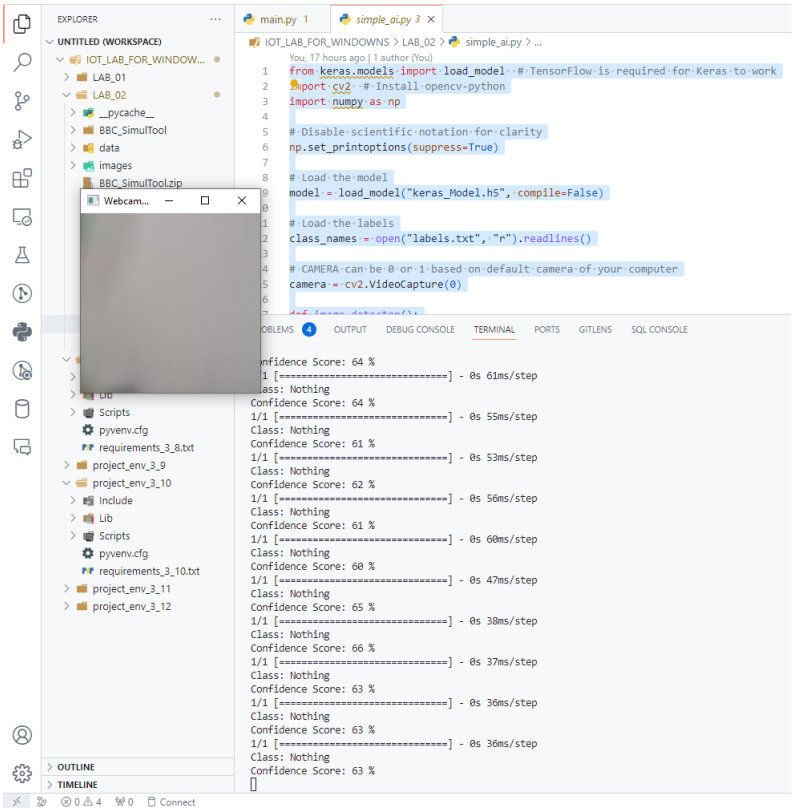
```
from keras.models import load_model
import cv2
import numpy as np

np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# Open camera
camera = cv2.VideoCapture(0)

def image_detector():
    # Grab the webcamera's image.
    ret, image = camera.read()

    # Resize the raw image into (224-height,224-width) pixels
    image = cv2.resize(image, (224, 224), interpolation=cv2.
    INTER_AREA)

    # Show the image in a window
    cv2.imshow("Webcam Image", image)

    # Make the image a numpy array and reshape it to the models
    input shape.
    image = np.asarray(image, dtype=np.float32).reshape(1, 224,
    224, 3)

    # Normalize the image array
    image = (image / 127.5) - 1

    # Predicts the model
    prediction = model.predict(image)
    index = np.argmax(prediction)
    class_name = class_names[index]
```

```
36      confidence_score = prediction[0][index]
37
38      # Print prediction and confidence score
39      print("Class:", class_name[2:], end="")
40      print("Confidence Score:", str(np.round(confidence_score *
    100))[:-2], "%")
41
42      # # Listen to the keyboard for presses.
43      # keyboard_input = cv2.waitKey(1)
44
45      # # 27 is the ASCII for the esc key on your keyboard.
46      # if keyboard_input == 27:
47      #     break
48      return class_name
49
50
51
```

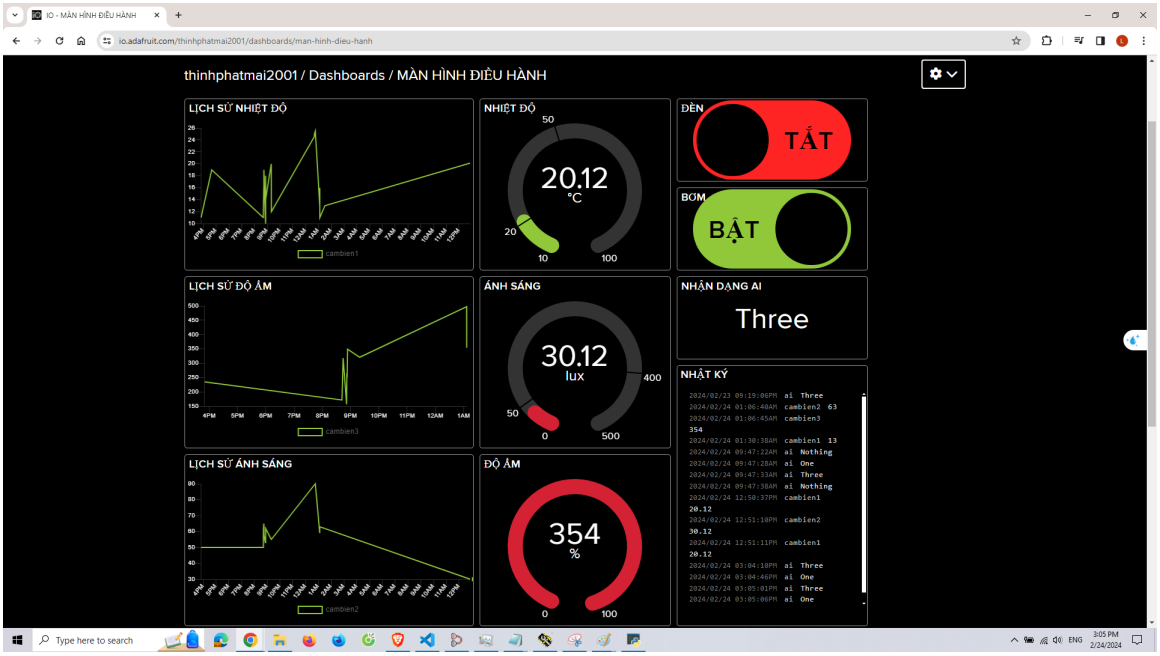
Listing 2.1: tích hợp AI vào ứng dụng

Ở đoạn mã trên ta sẽ viết hàm **image_detector** để nhận hình ảnh từ camera và sử dụng hình ảnh đó đưa ra kết quả từ mô hình học máy. Từ dòng 17 đến 30 là quá trình xử lý ảnh đầu vào. Dòng 33 là dự báo của mô hình, dòng 34 ta sẽ lấy ra chỉ số của nhãn mà mô hình cho là có độ tin cậy cao nhất, dòng 35 xuất ra tên lớp mà mô hình tin cậy nhất. Cuối cùng ta sẽ trả về tên lớp đó.

Dưới đây là hình ảnh của quá trình chạy dưới bằng máy tính cá nhân và gửi kết quả lên Adafruit-io



Hình 2.7: kết quả chạy thủ công dưới máy tính



Hình 2.8: kết quả được gửi lên adafruit-io

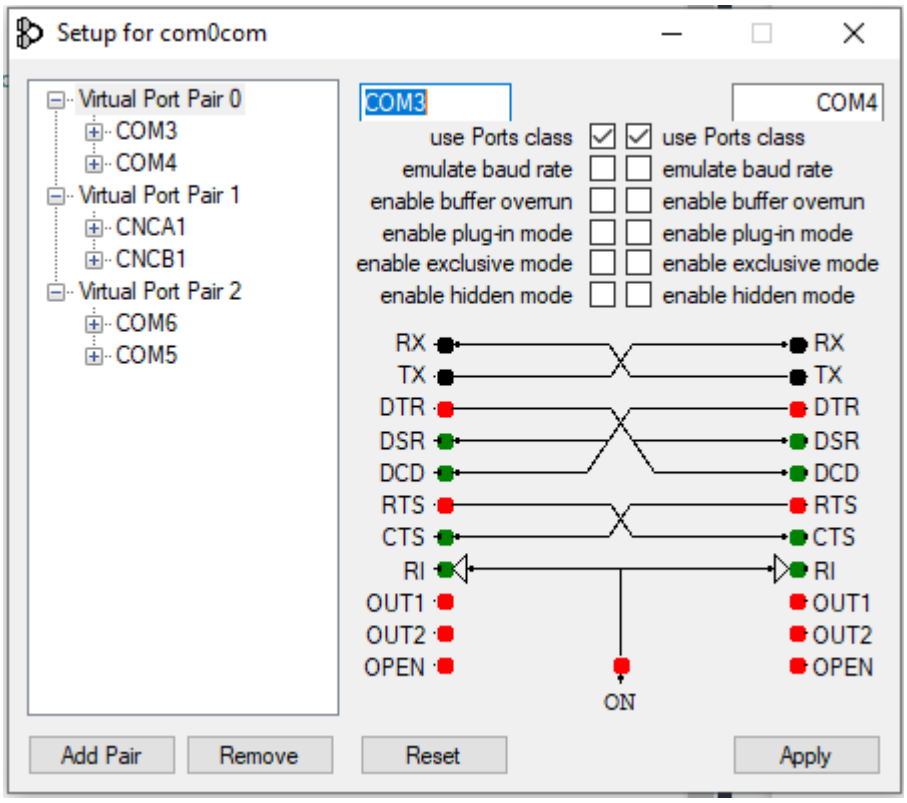
Chương 3

Kết nối UART

Trong phần này em sẽ trình bày các bước tạo kết nối UART giữa thiết bị phần cứng với chương trình python trong máy tính. Để giả lập thiết bị phần cứng và kết nối giữa phần cứng với python ta sẽ sử dụng các phần mền ảo hóa như **com0com** và **Hercules**

3.1 khởi tạo kết nối

Chương trình com0com đã khởi tạo các cổng com ảo cho phép kết nối giữa chương trình python với thiết bị ảo hóa phần cứng. Trong phần này em sẽ sử dụng cổng **Com3** cho chương trình python và **Com4** chương trình giả lập phần cứng.



Hình 3.1: các cổng com ảo

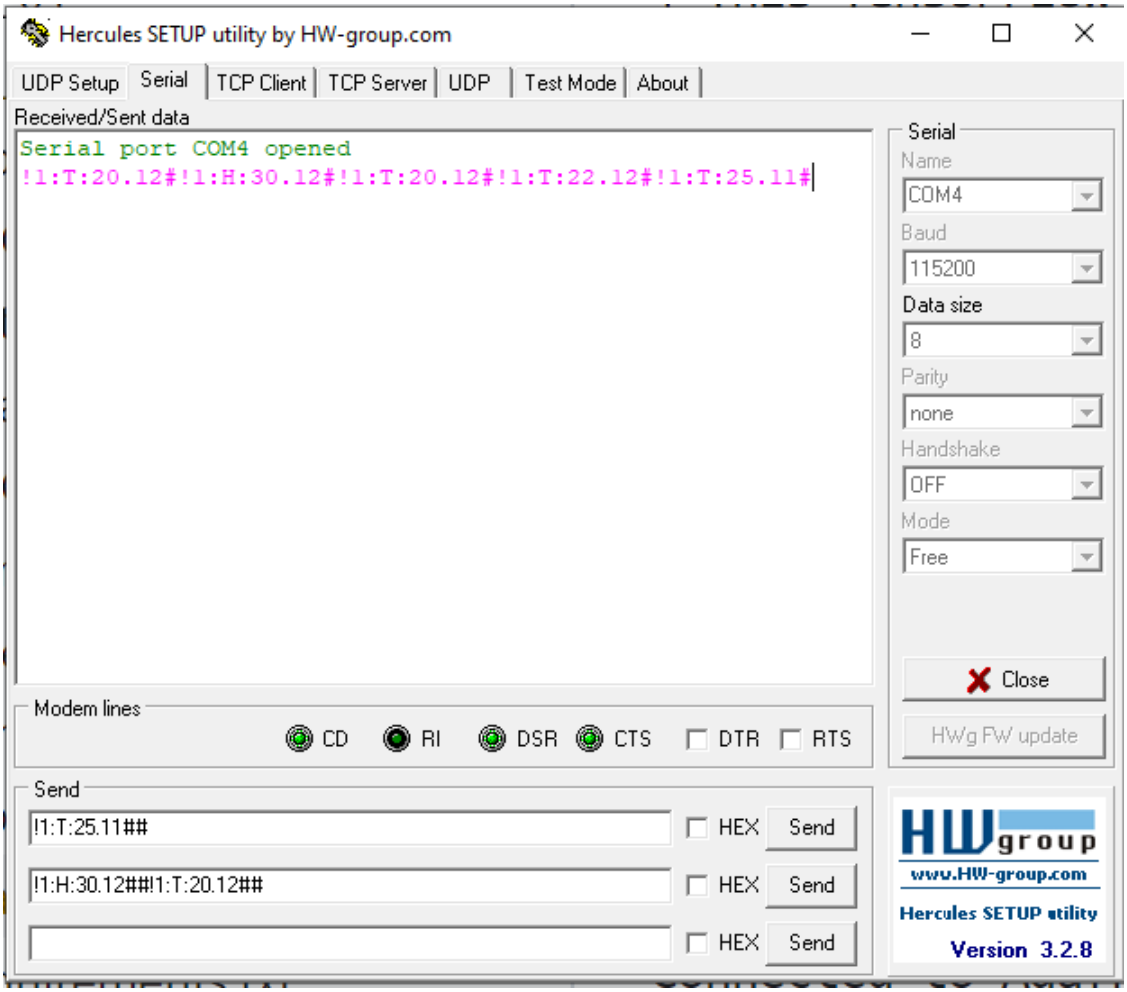
```python
import serial.tools.list_ports

def getPort():
    ports = serial.tools.list_ports.comports()
    N = len(ports)
    commPort = "None"
    for i in range(0, N):
        port = ports[i]
        strPort = str(port)
        if "USB Serial Device" in strPort:
            splitPort = strPort.split(" ")
            commPort = (splitPort[0])
    # return commPort
    return "COM3"

if getPort()!="None":
    ser = serial.Serial( port=getPort(), baudrate=115200)
    print(ser)

def processData(client, data):
    data = data.replace("!", "")
    data = data.replace("#", "")
    splitData = data.split(":")
    print(splitData)
    if splitData[1] == "T":
        client.publish("cambien1", splitData[2])
    elif splitData[1] == "H":
        client.publish("cambien2", splitData[2])

mess = ""
def readSerial(client):
    bytesToRead = ser.inWaiting()
    if (bytesToRead > 0):
        global mess
        mess = mess + ser.read(bytesToRead).decode("UTF-8")
        while ("#" in mess) and ("!" in mess):
            start = mess.find("!")
            end = mess.find("#")
            processData(client, mess[start:end + 1])
            if (end == len(mess)):
                mess = ""
            else:
                mess = mess[end+1:]


```

Listing 3.1: tích hợp AI vào ứng dụng

Trong chương trình trên do trước đó ta đã mặc định gián cổng COMP3 cho chương trình python thế nên giá trị trả về của hàm **getPort** là chuỗi "COMP3". Cậu lệnh ở dòng 16 và 17 mô tả quá trình kiểm tra và mở cổng kết nối giữ python với cổng com. Hàm **processData** sẽ là hàm có nhiệm vụ phân tách dữ liệu đã được xử lý và gửi giữ liệu này đến server của Adafruit bằng

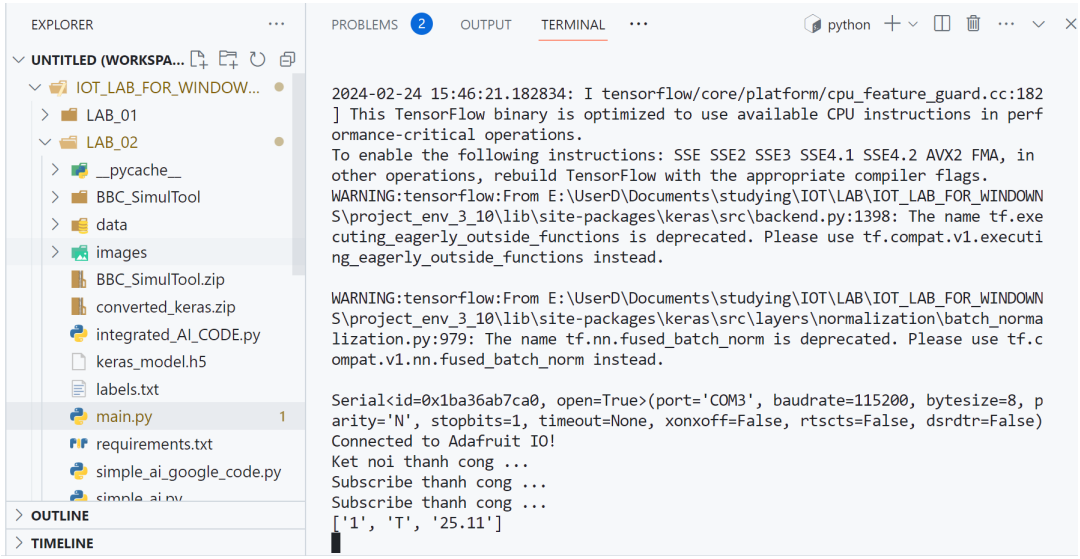
phương thức **publish** của đối tượng client. Hàm **readSerial** sẽ được gọi trong **main** với vai trò là lắng nghe dữ liệu gửi lên từ phần cứng, tiền sử lý và chuyển tiếp đến hàm **processData**.

DƯới đây là hình ảnh của quá trình gửi dữ liệu từ phần mền giả lập phần cứng đến chương trình python và kết quả xuất ra từ chương trình python và adafruit-IO.

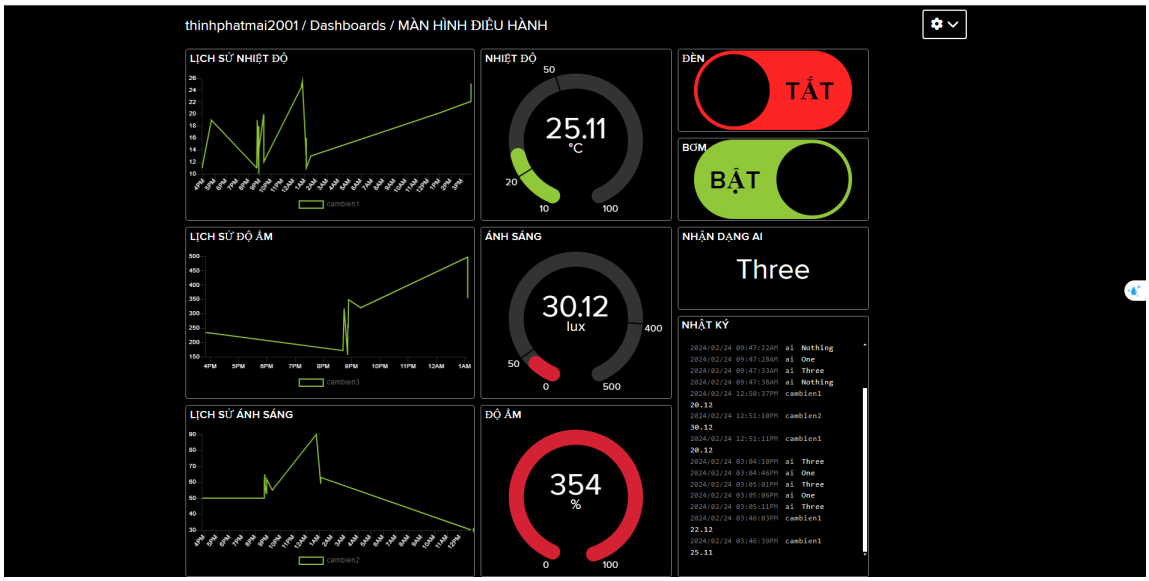


Hình 3.2: gửi giá trị từ Hecules

giả trị được gửi đi tù hecules là **"!1:T:25.11##"** và hình dưới đây là kết quả đọc được từ chương trình python và adafruit-io.



Hình 3.3: kết quả đọc dữ liệu từ chương trình Python



Hình 3.4: màn hình Dashboard đã nhận được dữ liệu chính xác

3.2 Repo mã nguồn

Mã nguồn và báo cáo đều được lưu trữ tại: `https://github.com/LeoPkm2-1/Lab_IOT_-CO3038-/tree/master/IOT_LAB_FOR_WINDOWNS/LAB_02`

Tài liệu tham khảo

[1] The Python Tutorial
https://docs.python.org/3/tutorial/index.html [Truy cập: 20-02-2024]

[2] Python Tutorial
https://www.w3schools.com/python/ [Truy cập: 20-02-2024]

[3] adafruit-io python
https://pypi.org/project/adafruit-io/ [Truy cập: 20-02-2024]

[4] Teachable-Machine
https://teachablemachine.withgoogle.com/ [Truy cập: 20-02-2024]