



THE DARIU FOUNDATION

Trí Tuệ Nhân Tạo & Khoa Học Dữ Liệu



Lê Trọng Nhân - Lê Phương Nam
Nguyễn Văn Hạnh - Băng Ngọc Bảo Tâm

Mục lục

I	Video AI với Teachable Machine	7
Chương 1.	Học máy với Google - Teachable Machine	9
1	Giới thiệu	10
2	Bắt đầu với Học Máy Google	10
3	Huấn luyện dữ liệu sai	13
4	Xuất kết quả huấn luyện	14
5	Lưu dự án vào Google drive	16
6	Mở dự án từ Google drive	16
7	Câu hỏi ôn tập	17
Chương 2.	Học máy Google trên Python	19
1	Giới thiệu	20
2	Tích hợp học máy Google vào PyCharm	20
3	Câu hỏi ôn tập	24
Chương 3.	Tích hợp Webcam để nhận diện hình ảnh trên Python	25
1	Giới thiệu	26
2	Tích hợp Webcam	26
3	Tích hợp trợ lý ảo	28
4	Câu hỏi ôn tập	31
II	Nhận dạng âm thanh với Teachable Machine	33
Chương 4.	Huấn luyện âm thanh trên Teachable Machine	35
1	Giới thiệu	36
2	Huấn luyện hệ thống âm thanh	36
3	Xuất kết quả huấn luyện	39
4	Lưu dự án vào Google drive	41
5	Mở dự án từ Google drive	42
6	Câu hỏi ôn tập	43
Chương 5.	AI và Robot trên Scratch 3.0	45
1	Giới thiệu	46
2	Scratch 3.0 trực tuyến	46
3	Trí tuệ nhân tạo trên Scratch 3.0	47

3.1	Kích hoạt trí tuệ nhân tạo	47
3.2	Bắt đầu việc nhận dạng	48
4	Kết nối với Robot Microbit	49
Chương 6. Tích hợp Micro để nhận dạng âm thanh trên Python		51
1	Giới thiệu	52
2	Tích hợp Micro	52
3	Câu hỏi ôn tập	64
III Khoa học dữ liệu		65
Chương 7. Khoa học Dữ liệu cùng Python		67
1	Giới thiệu	68
2	Thư viện thường được sử dụng	68
2.1	Numpy	68
2.2	Pandas	68
2.3	Matplotlib	69
2.4	SciPy	70
2.5	Scikit-Learn	71
3	Thu thập và xử lý dữ liệu	72
3.1	Thu thập dữ liệu	72
3.2	Xử lý dữ liệu	72
3.2.1	Thêm dữ liệu vào chương trình	72
3.2.2	Trích xuất, chia nhỏ dữ liệu	73
4	Câu hỏi ôn tập	75
Chương 8. Vẽ đồ thị biểu diễn dữ liệu trực quan		77
1	Giới thiệu	78
2	Vẽ biểu đồ bằng thư viện Matplotlib	78
2.1	Biểu đồ đường	78
2.2	Biểu đồ cột	81
2.3	Biểu đồ tròn	83
3	Biểu diễn kết quả thống kê bằng khuôn hình	85
4	Câu hỏi ôn tập	88
Chương 9. Dự đoán tương lai gần bằng thuật toán Hồi Quy Tuyến Tính		89
1	Giới thiệu	90
2	Xử lý dữ liệu	90
3	Biểu diễn số liệu dưới dạng đồ thị	90
4	Sử dụng thuật toán Hồi Quy Tuyến Tính trong việc dự đoán tương lai gần	92
5	Câu hỏi ôn tập	94
IV Robotics với Python - Ứng dụng Google Home		95
Chương 10. Trợ lý ảo trên Điện thoại		97
1	Giới thiệu	98
2	Cài đặt Giọng nói - Văn bản	98

3	Cài đặt ứng dụng Trợ lý ảo	99
4	Hướng dẫn sử dụng	100
4.1	Đặt tên cho Trợ lý ảo	100
4.2	Cấu hình lệnh cho trợ lý ảo	100
5	Câu hỏi ôn tập	101
Chương 11. Tích hợp Microbit vào Trợ lý ảo		103
1	Giới thiệu	104
2	Cài đặt phần mềm Mu	105
3	Chương trình đầu tiên	106
4	Hiện thực chương trình cho Microbit	107
4.1	Nhận dữ liệu từ điện thoại	107
4.2	Gửi dữ liệu lên điện thoại	108
5	Câu hỏi ôn tập	109
Chương 12. Điều khiển ngoại vi qua giao tiếp không dây		111
1	Giới thiệu	112
2	Nhận dữ liệu không dây	112
3	Đóng/Mở công tắc Relay	113
4	Giám sát cảm biến và gửi cảnh báo	114
5	Câu hỏi ôn tập	116

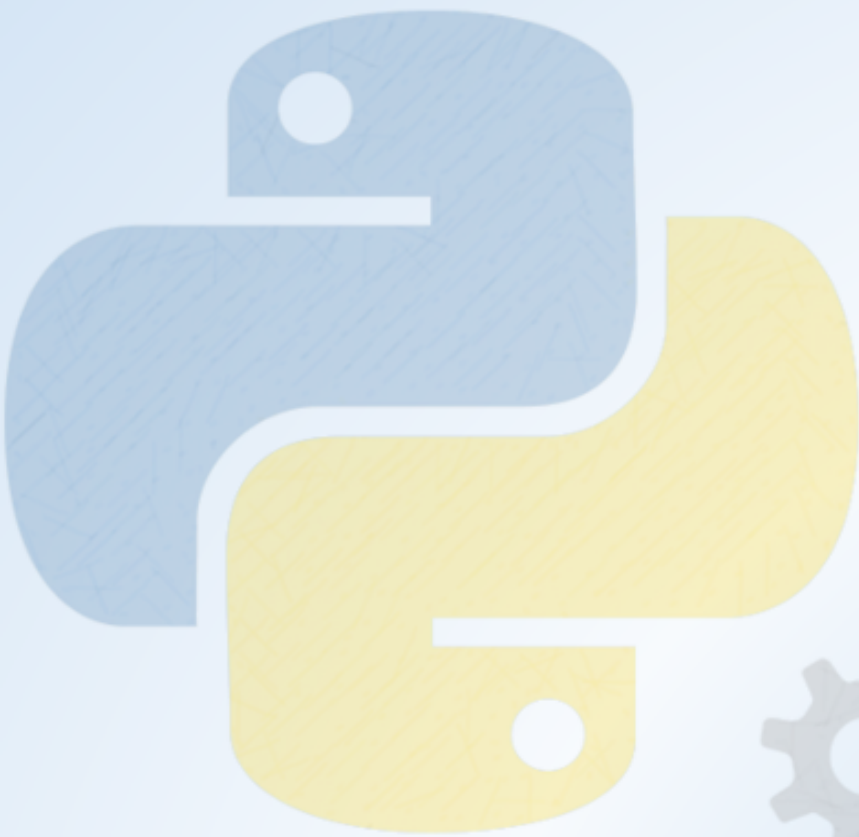
Phần I

Video AI với Teachable Machine



CHƯƠNG 1

Học máy với Google - Teachable Machine



Pandas



1 Giới thiệu

Trong thời đại bùng nổ của cuộc cách mạng công nghệ 4.0, chắc hẳn chúng ta đã nghe nói nhiều về khái niệm Trí tuệ nhân tạo, hay còn gọi là AI (Artificial Intelligence). Tuy nhiên để có thể hiểu nó một cách cơ bản và trực quan không phải là một điều dễ dàng. Vì vậy, từ năm 2017 Google đã xây dựng một công cụ trực tuyến, có tên gọi là Teachable Machine, để mang lại cho mọi người một cái nhìn tổng quan nhất về trí tuệ nhân tạo. Chỉ cần rất ít thao tác, ở mọi lứa tuổi, chúng ta có thể cảm nhận được trí tuệ nhân tạo gồm có những khái niệm cơ bản nào, nó có thể làm được gì.

Teachable Machine ban đầu chỉ là một trang web trực tuyến, với mục đích chính là cho người dùng trải nghiệm với các sản phẩm của trí tuệ nhân tạo, như nhận diện hình ảnh hay nhận diện âm thanh. Tuy nhiên hiện nay, với phiên bản 2.0, Teachable Machine đã trở thành một công cụ trong việc thiết kế một ứng dụng trí tuệ nhân tạo cho chúng ta. Công cụ này đã hỗ trợ nhiều nền tảng lập trình khác nhau, như Scratch (ngôn ngữ cho học sinh) và Python, ngôn ngữ lập trình rất mạnh mẽ cho các ứng dụng thời đại mới. Ngoài ra, rất nhiều các nền tảng AI khác đang dựa trên lõi trung tâm là Google Teachable Machine.

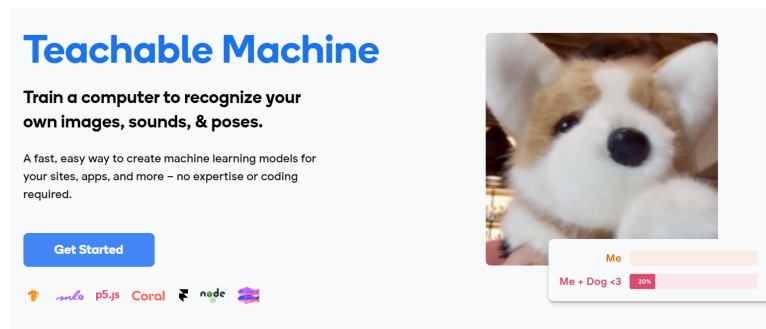
Trong bài hướng dẫn này, chúng tôi sẽ giới thiệu các thao tác cơ bản trên Teachable Machine, trước khi chúng ta bắt đầu việc lập trình kết nối giữa Python và Teachable Machine. Chúng tôi cũng tập trung vào hướng dẫn công cụ đầu tiên, là nhận diện bằng hình ảnh (Image Classifier). Các công cụ khác sẽ được trình bày trong những bài tiếp theo. Các mục tiêu chính trong bài đầu tiên được liệt kê như sau:

- Hiểu được các thành phần cơ bản của AI
- Sử dụng được công cụ Teachable Machine của Google
- Trải nghiệm Học Máy với Google trên trang web
- Lưu dự án AI trên công cụ Google Drive

Trong phần còn lại của hướng dẫn, chúng tôi sẽ xài thuật ngữ việt hóa cho Teachable Machine là "Học Máy Google" cho thuận tiện với bạn đọc.

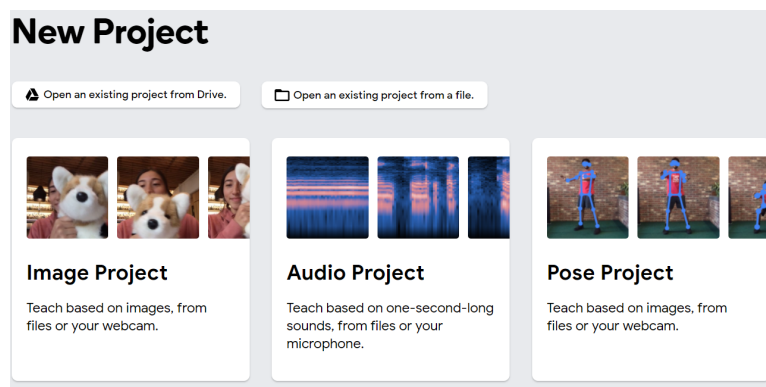
2 Bắt đầu với Học Máy Google

Học Máy Google đã được xây dựng thành một công cụ trực tuyến. Chúng ta bắt đầu bằng việc vào đường dẫn <https://teachablemachine.withgoogle.com/>, giao diện sau đây sẽ hiện ra:



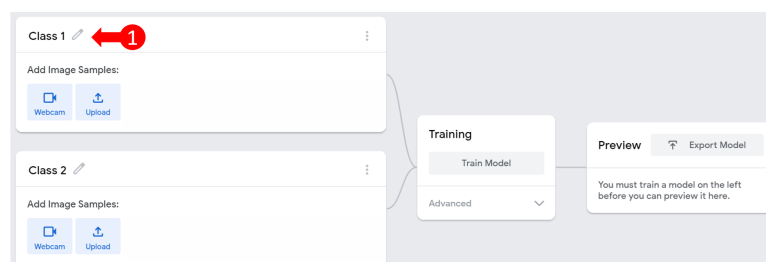
Hình 1.1: Trang web online của Học máy Google

Tại trang web này, chúng ta có thể xem trước một số hình ảnh cũng như video giới thiệu của Google về công nghệ Học máy. Tiếp theo, nhấn vào nút **Get Started** để bắt đầu, giao diện của công cụ học máy với Google được hiện ra như sau:



Hình 1.2: Công cụ Học máy với Google - Image Project

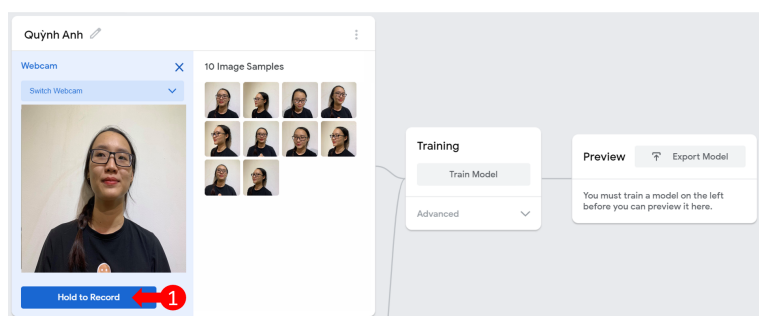
Chúng ta chọn tiếp vào Image Project, công cụ để phân nhận diện hình ảnh. Hai công cụ còn lại (Nhận dạng âm thanh và Nhận dạng dáng người) sẽ được trình bày sau. Giao diện sau đây sẽ được hiện ra:



Hình 1.3: Giao diện để huấn luyện hình ảnh

Để mô hình AI có thể hoạt động, chúng ta phải **huấn luyện** (train) nó trước. Bạn có thể đặt tên cho đối tượng mà bạn muốn nhận diện bằng cách nhấn vào biểu tượng thay đổi bên cạnh tên mặc định Class1 và Class2. Thao tác này còn có một tên gọi khác là gán nhãn cho đối tượng. Bước tiếp theo là cung cấp dữ liệu cho từng đối tượng. Google hỗ trợ 2 phương pháp khác nhau để cung cấp dữ liệu hình ảnh. Chúng ta sẽ bắt đầu trước với việc

sử dụng webcam để cung cấp hình ảnh. Khi nhấn vào biểu tượng webcam, giao diện sau đây sẽ hiện ra:

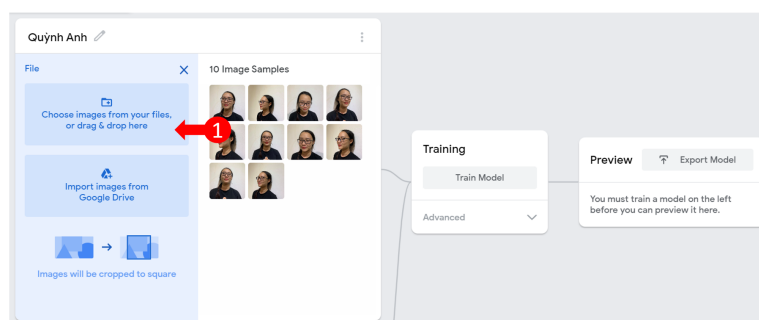


Hình 1.4: Chụp hình ảnh từ webcam

Chúng ta sẽ nhấn nút **Hold To Record** để bắt đầu chụp hình, như minh họa ở Hình 1.4. Càng có nhiều hình cho đối tượng, việc nhận diện sẽ càng chính xác, tuy nhiên thời gian huấn luyện sẽ lâu hơn. Ở đây, chúng tôi chọn mẫu khoảng 10 hình, nhưng bạn hoàn toàn có thể tăng lên 50 hình nếu muốn.

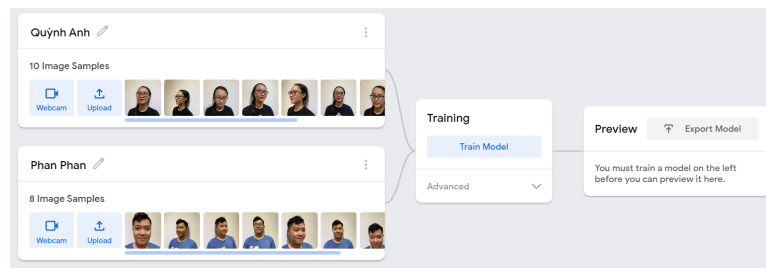
Khi muốn xóa một hình mà bạn không ưng ý, chỉ cần rê chuột vào tấm hình, biểu tượng xóa sẽ xuất hiện để bạn có thể nhấn vào.

Trong một số trường hợp, việc chụp ảnh từ webcam của máy tính không thuận tiện, chúng ta có thể chọn công cụ thứ 2 để cung cấp hình ảnh cho hệ thống trí tuệ nhân tạo. Tắt công cụ webcam đi (nhấn vào dấu X ở góc bên phải khung hình). Khi chọn vào công cụ **Upload**, giao diện sau đây sẽ hiện ra:



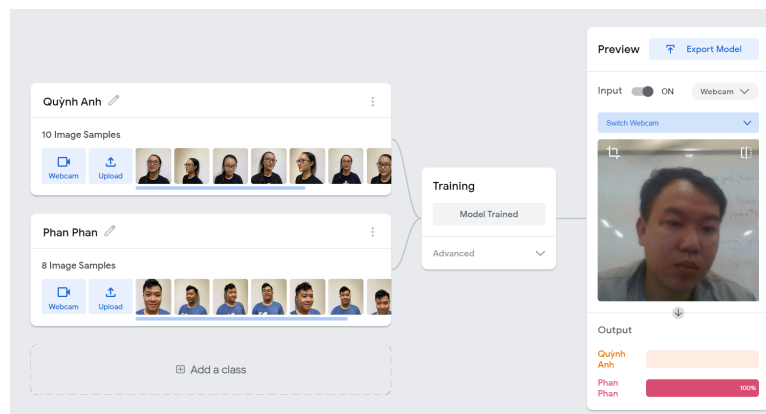
Hình 1.5: Tải thêm hình ảnh từ máy tính

Chọn vào công cụ đầu tiên, bạn đã có thể tải hình ảnh từ máy tính lên hệ thống huấn luyện của Google. Chúng ta sẽ làm tương tự cho nhóm thứ 2, như minh họa ở hình sau đây:



Hình 1.6: Tải dữ liệu cho nhóm thứ 2

Cuối cùng, chúng ta nhấn vào nút **Train Model** để hệ thống bắt đầu quá trình huấn luyện dựa trên tập dữ liệu hình ảnh mà chúng ta cung cấp. Ngay sau khi quá trình huấn luyện kết thúc, hệ thống AI sẽ bắt đầu chạy để nhận dạng. Kết quả của việc nhận dạng là xác suất trùng với các đối tượng đã được huấn luyện, như minh họa ở hình bên dưới:

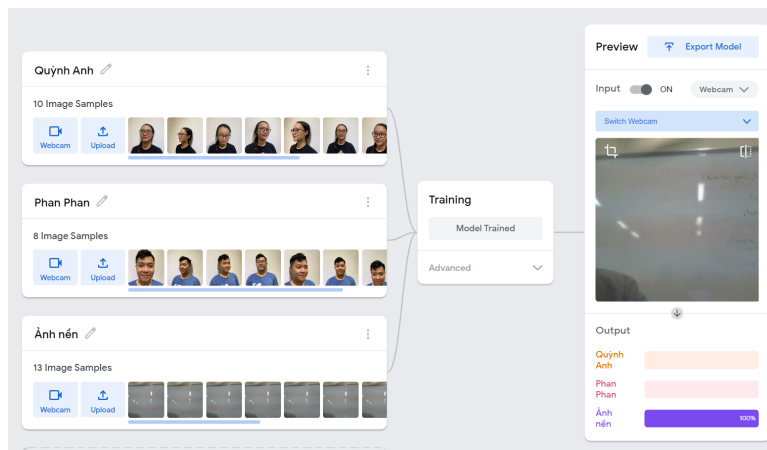


Hình 1.7: Huấn luyện và chạy thử AI

3 Huấn luyện dữ liệu sai

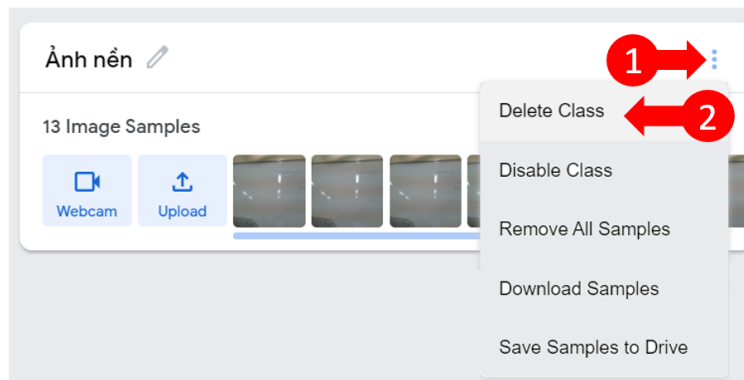
Đây là một khái niệm vô cùng quan trọng trong hệ thống trí tuệ nhân tạo. Bên cạnh việc huấn luyện nhận đúng, bạn cũng phải cần huấn luyện các ngữ cảnh không thuộc các đối tượng nhận dạng mong đợi. Ví dụ như trong hình Hình 1.7, mặc dù hình ảnh thực tế hoàn toàn khác xa với tập dữ liệu, hệ thống AI vẫn đang nhận diện là đối tượng **"Phan Phan"**. Một ví dụ khác cho hiện tượng này là khi chúng ta làm một hệ thống điểm danh tự động. Trong hệ thống đã được huấn luyện trước 10 người thật chẳng hạn. Tuy nhiên trong quá trình vận hành, camera có thể quay vào một khoảng không và hệ thống AI vẫn nhận dạng đây là một ai đó.

Một kĩ thuật đơn giản để khắc phục hiện tượng này là huấn luyện thêm các ngữ cảnh không mong muốn. Từ Hình 1.7, nhấn vào lựa chọn **Add a class** để thêm một đối tượng mới và cũng cung cấp hình ảnh cho đối tượng này như bình thường. Ở đây, chúng tôi thêm dữ liệu gọi là **Ảnh nền**, như minh họa bên dưới:



Hình 1.8: Thêm đối tượng ảnh nền cho hệ thống

Trong trường hợp muốn xóa một đối tượng, chúng ta nhấn vào biểu tượng **option** ở góc bên phải (bước 1), và chọn tiếp **Delete class** (bước 2), như minh họa ở hình bên dưới:

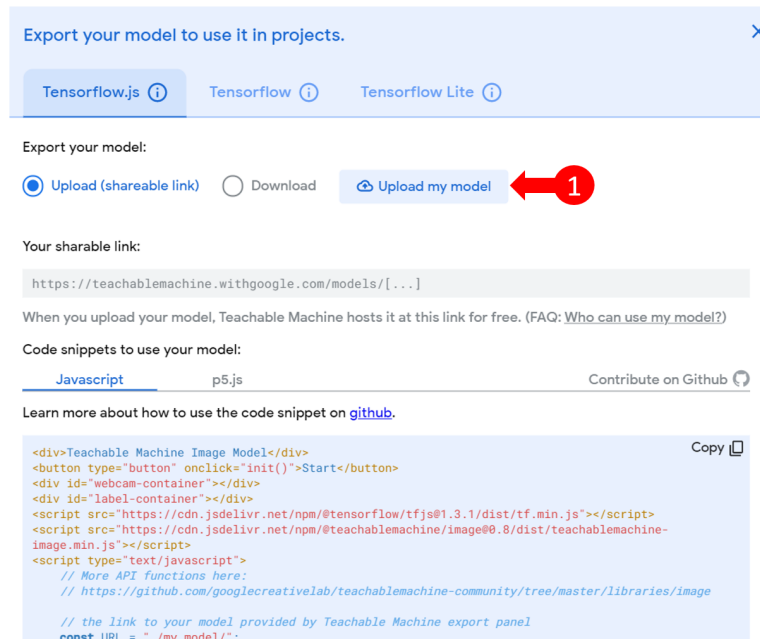


Hình 1.9: Xóa một đối tượng khỏi hệ thống AI

Lưu ý: Mỗi khi thay đổi hình hoặc thêm/bớt đối tượng, phải huấn luyện lại hệ thống.

4 Xuất kết quả huấn luyện

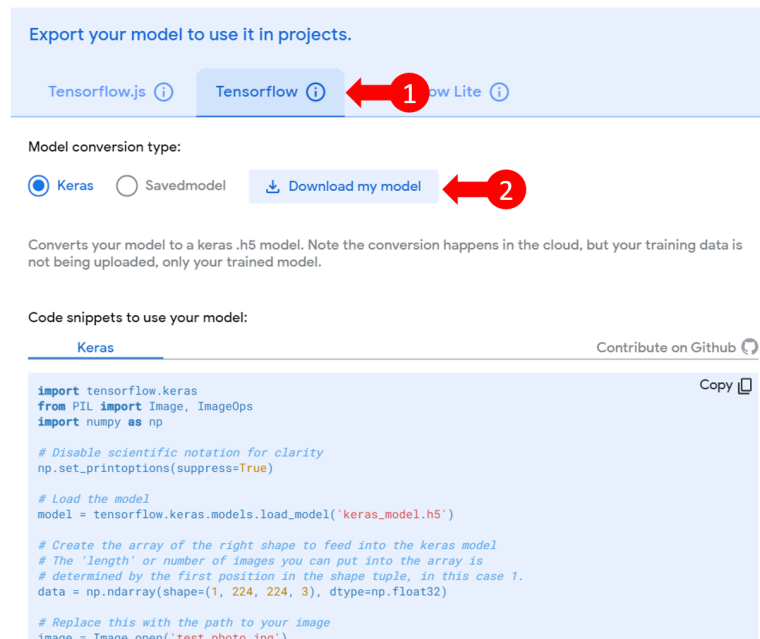
Đây là bước cuối cùng, để chúng ta có thể tích hợp kết quả của việc huấn luyện vào các ngôn ngữ lập trình khác như Scratch hay Python. Bằng cách nhấn vào nút Export Model, giao diện sau đây sẽ hiện ra:



Hình 1.10: Xuất kết quả huấn luyện

Chúng ta sẽ chọn vào mục **Upload my model**, và chờ cho đến khi tác vụ này hoàn tất. Khi việc tải dữ liệu lên server của Google hoàn tất, chúng ta sẽ có một đường dẫn trong mục **Your sharable link**. Bạn hãy sao chép đường dẫn này lại để dùng trong việc lập trình trên Scratch 3.0.

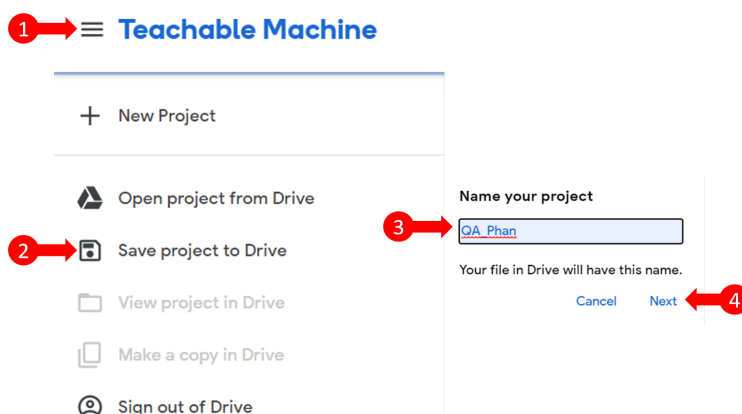
Chuyển sang tab **Tensorflow**, chúng ta tải về một file được mã hóa cho dữ liệu nhận dạng, gọi là file h5, bằng cách nhấn vào nút **Download my model**, như minh họa ở hình bên dưới. Đây sẽ là file cần thiết cho việc lập trình trên Python trong những bài hướng dẫn tiếp theo.



Hình 1.11: Xuất kết quả huấn luyện ra file h5

5 Lưu dự án vào Google drive

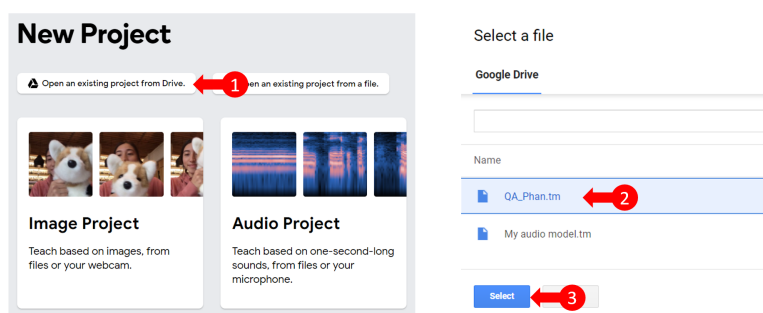
Với những hỗ trợ toàn diện từ phía Google, hệ thống AI của bạn có thể được lưu lại trực tuyến trên Google drive. Bằng các trở lại màn hình chính và làm theo 4 bước hướng dẫn ở hình bên dưới như: Chọn Save project to Drive, Đặt tên cho project và cuối cùng nhấn nút Next.



Hình 1.12: Lưu dự án vào Google drive

6 Mở dự án từ Google drive

Trong trường hợp muốn mở lại dự án cũ để cập nhật hoặc làm tiếp, từ trang chủ của Học Máy Google, chúng ta chọn **Open existing project from Drive** rồi lựa chọn trong danh sách các dự án đã lưu, như minh họa ở hình bên dưới.



Hình 1.13: Mở một dự án từ Google drive

7 Câu hỏi ôn tập

1. Công cụ Học máy với Google có tên là gì?
 - A. Teachable Machine
 - B. Artificial Intelligence
 - C. Deep Learning
 - D. Machine Learning
2. Hiện tại có bao nhiêu công cụ AI đang được hỗ trợ sẵn bởi Google?
 - A. 1
 - B. 2
 - C. 3
 - D. Tất cả đều sai
3. Dữ liệu để huấn luyện cho hệ thống nhận dạng bằng ảnh có thể được lấy từ nguồn nào?
 - A. Trực tiếp từ webcam
 - B. Tải ảnh từ máy tính lên
 - C. Ảnh từ Google Drive của người dùng
 - D. Tất cả các nguồn ảnh trên
4. Sau khi huấn luyện, kết quả thực thi của hệ thống trí tuệ nhân tạo là gì?
 - A. Một hình ảnh dưới dạng xem trực tiếp (live view)
 - B. Xác suất tính theo phần trăm tỉ lệ nhận dạng được
 - C. Một định danh (ID) cho đối tượng nhận dạng
 - D. Tất cả đều đúng
5. Để loại bỏ việc nhận dạng ảnh nền thành một hình nào đó trong danh sách các đối tượng, chúng ta cần:
 - A. Thêm nhiều đối tượng hơn để loại bỏ ảnh nền
 - B. Huấn luyện thêm một đối tượng là ảnh nền
 - C. Huấn luyện lại ảnh cho đối tượng nhiều lần
 - D. Không cần xử lý vì Google đã hỗ trợ
6. Mô hình huấn luyện của Học máy Google hỗ trợ cho những môi trường lập trình nào sau đây?
 - A. Scratch 2.0
 - B. Scratch 3.0
 - C. Python
 - D. Scratch 3.0 và Python
7. Để có thể sử dụng mô hình Học máy Google trên môi trường lập trình trên Python, chúng ta cần:
 - A. Lưu lại đường dẫn trực tuyến của mô hình
 - B. Lưu lại file Tensorflow
 - C. Lưu lại file h5
 - D. Tất cả các bước trên

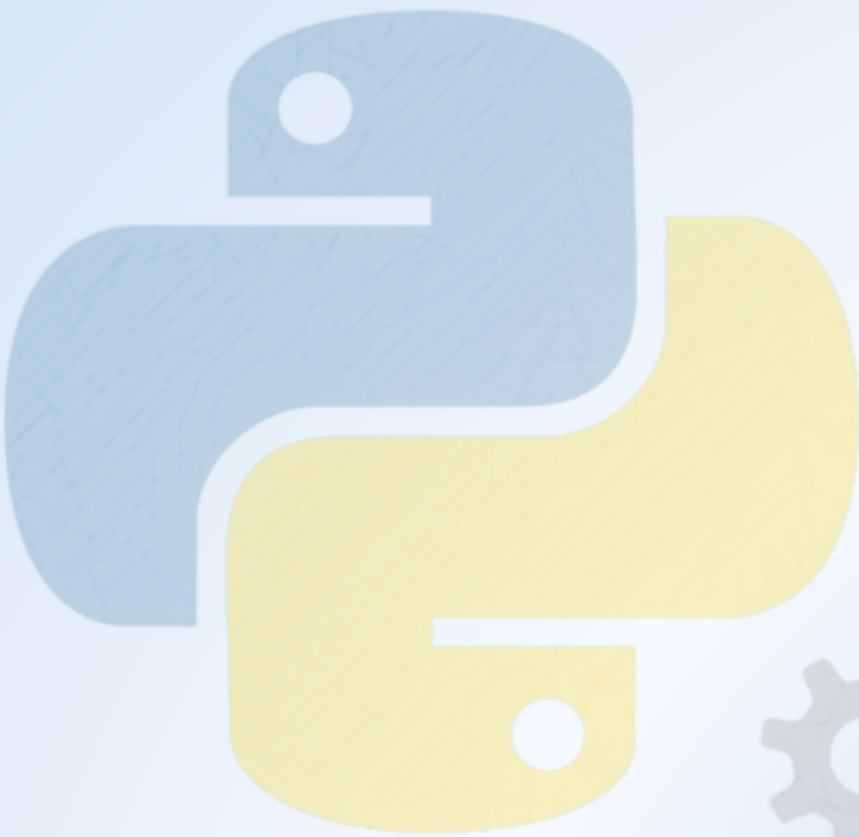
Đáp án

1. A 2. C 3. D 4. B 4. D 5. B 6. D 7. C



CHƯƠNG 2

Học máy Google trên Python



Pandas



1 Giới thiệu

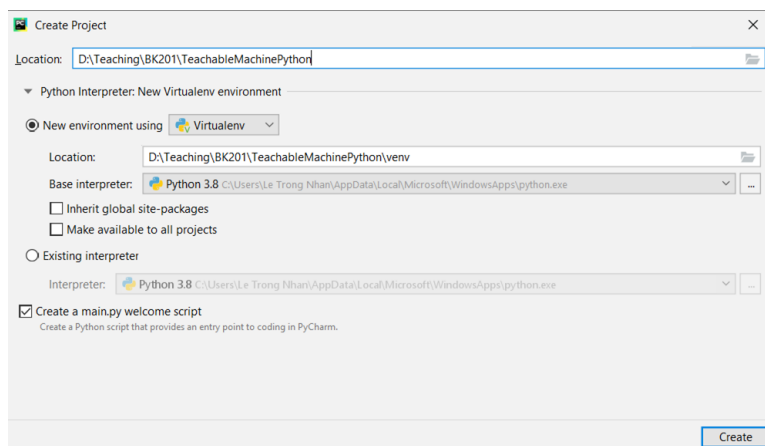
Trong phần hướng dẫn này, chúng ta sẽ sử dụng kết quả huấn luyện ở bài trước và hiện thực một chương trình nhận dạng bằng ngôn ngữ Python. Sở dĩ chúng ta phải làm điều này là vì trang web trực tuyến của Google chỉ là công cụ để kiểm tra hệ thống trí tuệ nhân tạo và chúng ta không thể thêm các chức năng khác vào trang web này.

Trong bài này chúng ta sẽ hiện thực một chương trình bằng ngôn ngữ Python trên PyCharm để làm chủ công nghệ Học máy của Google, trước khi chúng ta tích hợp thêm các chức năng mới, như trợ lý ảo chẳng hạn. Các mục tiêu chính của bài hướng dẫn này như sau:

- Tích hợp kết quả huấn luyện của Google vào PyCharm
- Sử dụng được mã nguồn mở cung cấp bởi Google
- Hiện thực được chương trình xử lý nhận dạng ảnh tĩnh bằng Python

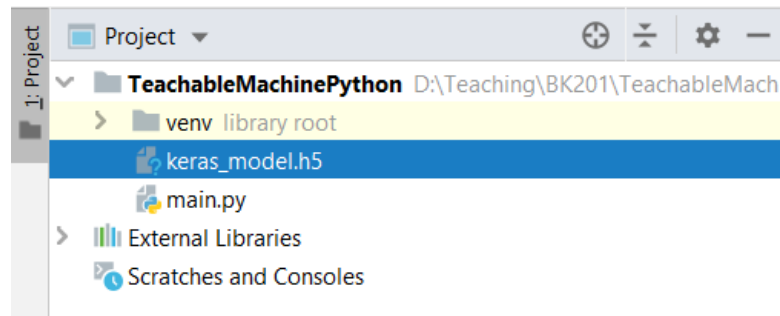
2 Tích hợp học máy Google vào PyCharm

Bước 1: Khởi động phần mềm PyCharm, tạo một dự án mới bằng cách chọn **File/New Project**, như minh họa ở hình bên dưới:



Hình 2.1: Tạo một dự án mới trên PyCharm

Bước 2: Chép file h5 đã tải ở bài trước vào cùng thư mục với file main.py. Để làm được việc này, chúng ta nhấn chuột phải vào file main.py, rồi chọn **Show in Explorer**. Trình duyệt thư mục sẽ được mở ra, chúng ta chép file h5 vào thư mục này. Kết quả của bước này, chúng ta sẽ có thêm thư viện **keras_model.h5** cùng cấp với file main.py, như minh họa ở hình bên dưới:



Hình 2.2: Chép file h5 vào cùng thư mục

Bước 3: Chép chương trình sau đây vào file main.py. Thực ra, đây là chương trình mẫu được hỗ trợ trên trang chủ Học máy với Google, như sau:

```

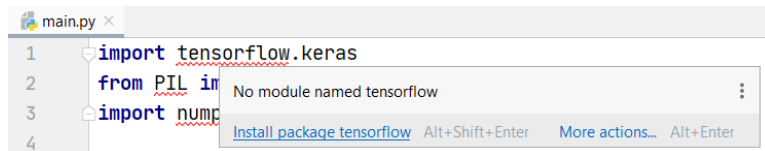
1 import tensorflow.keras
2 from PIL import Image, ImageOps
3 import numpy as np
4
5 # Disable scientific notation for clarity
6 np.set_printoptions(suppress=True)
7
8 # Load the model
9 model = tensorflow.keras.models.load_model('keras_model.h5'
10 )
11
12 # Create the keras model
13 data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
14
15 # Replace this with the path to your image
16 image = Image.open('test_photo.jpg')
17
18 #resize the image to a 224x224
19 size = (224, 224)
20 image = ImageOps.fit(image, size, Image.ANTIALIAS)
21
22 #turn the image into a numpy array
23 image_array = np.asarray(image)
24
25 # display the resized image
26 image.show()
27
28 # Normalize the image
29 normalized_image_array = (image_array.astype(np.float32) /
30 127.0) - 1
31
32 # Load the image into the array
33 data[0] = normalized_image_array
34
35 prediction = model.predict(data)

```

34 `print(prediction)`

Chương trình 2.1: Chương trình mẫu từ trang chủ Học máy với Google

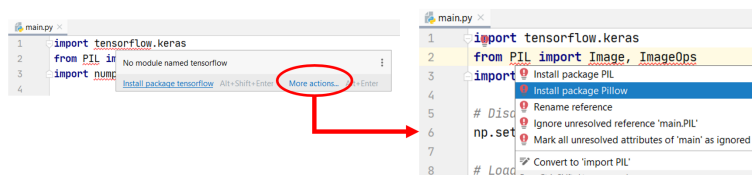
Bước 4: Cài đặt thêm thư viện cho chương trình. Thực ra, để chạy chương trình này, chúng ta phải cài đặt thêm 3 thư viện **tensorflow**, **PIL** và **numpy**, như chỉ định ở 3 dòng đầu tiên của chương trình. Bằng cách rê chuột vào thư viện đang bị báo lỗi, chương trình sẽ tự động gợi ý như minh họa ở hình bên dưới. Chúng ta chỉ cần chọn vào **Install package tensorflow** để cài đặt thêm.



Hình 2.3: Cài đặt thư viện theo gợi ý của PyCharm

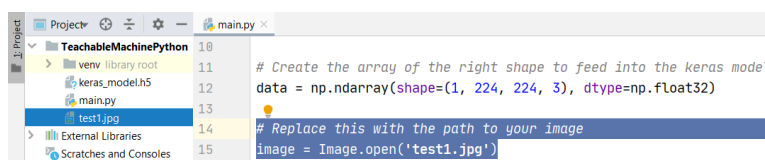
Lưu ý: Trong quá trình cài đặt thư viện, hãy chú ý tới các thông báo của PyCharm. Nếu quá trình cài đặt thành công, phần import thư viện sẽ không còn báo lỗi và sẽ có thông báo **Packages installed successfully**. Khi cài đặt xong thư viện tensorflow, thì **thư viện numpy cũng đã tự động cài đặt thêm**.

Bước 5: Chúng ta tiếp tục làm tương tự để cài đặt thư viện PIL. Tuy nhiên, lần này việc cài đặt sẽ bị báo lỗi. Đây cũng là điều thường xuyên xảy ra khi làm việc với mã nguồn mở trên Python, do việc xung đột phiên bản hoặc phiên bản cũ không còn tồn tại. Đối với trường hợp này, chúng ta sẽ chọn **More actions** để xem các gợi ý khác của PyCharm, như minh họa ở hình bên dưới:



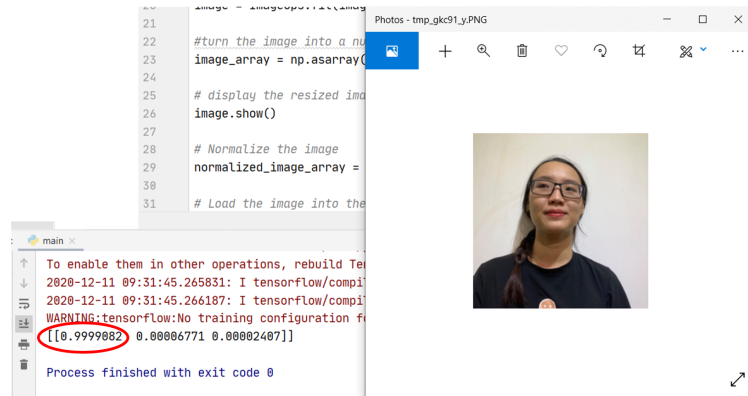
Hình 2.4: Cài đặt lại thư viện PIL bằng cách chọn More actions...

Bước 6: Thêm một ảnh vào cùng thư mục với main.py và sửa lại câu lệnh ở dòng 15 của chương trình. Trong hướng dẫn bên dưới, chúng tôi chép vào một hình có tên là **test1.jpg** và sửa lại chương trình trong file main.py, như minh họa ở hình bên dưới:



Hình 2.5: Đặt file ảnh test1.jpg vào để nhận diện

Đến đây, chúng ta đã có thể chạy chương trình. Ở lần đầu tiên, việc cài đặt sẽ tốn thời gian. Sau khi chạy thành công, hình ảnh nhận diện sẽ hiện lên cùng với độ tin cậy của việc nhận dạng, được tính theo phần trăm, như minh họa ở hình bên dưới:



Hình 2.6: Kết quả thực thi của chương trình

Ở đây, mô hình Học máy với Google của chúng ta có 3 đối tượng là "Quỳnh Anh", "Phan Phan" và "Ảnh nền". Do đó, kết quả xuất ra của việc nhận dạng là một mảng có 3 phần tử, tương ứng với xác suất nhận dạng của 3 đối tượng. Chúng ta sẽ hiện thực thêm một đoạn chương trình nhỏ để tìm ra tỉ lệ nhận dạng tốt nhất trong các đối tượng đã huấn luyện, và in ra kết quả thân thiện hơn. Bạn đọc có thể tham khảo đoạn chương trình bên dưới để tích hợp vào trong chương trình của mình.

```
1 name = ["Quynh Anh", "Phan Phan", "Anh nen"]
2 index = -1
3 max_value = -1
4 for i in range(0, len(prediction[0])):
5     if max_value < prediction[0][i]:
6         max_value = prediction[0][i]
7         index = i
8 print("Ket Qua: ", name[index])
9 print("Chinh Xac: ", max_value)
```

Chương trình 2.2: Chương trình mẫu từ trang chủ Học máy với Google

Chương trình ở trên thực chất chỉ là tìm số lớn nhất trong mảng. Tuy nhiên ở đây chỉ có thêm một chức năng là số lớn nhất đó đang nằm ở đâu trong mảng (lưu trong biến **index**) để chúng ta có thể lấy ra nhãn của đối tượng tương ứng (lưu trong biến **name**).

3 Câu hỏi ôn tập

1. Mô hình nhận dạng ảnh có tên là gì?
 - A. Tensorflow
 - B. Keras
 - C. Techable machine
 - D. Machine Learning
2. Kết quả trả về (biến prediction) là:
 - A. Biến đơn kiểu số thực
 - B. Biến đơn kiểu số nguyên
 - C. Mảng 1 chiều
 - D. Mảng 2 chiều
3. Số lượng phần tử chiều thứ nhất của prediction là bao nhiêu?
 - A. 0
 - B. 1
 - C. Tùy thuộc vào số lượng đối tượng huấn luyện
 - D. Tất cả đều sai
4. Số lượng phần tử chiều thứ hai của prediction là bao nhiêu?
 - A. 0
 - B. 1
 - C. Tùy thuộc vào số lượng đối tượng huấn luyện
 - D. Tất cả đều sai
5. Độ tin cậy được tính theo đơn vị gì?
 - A. Phần trăm
 - B. Độ chính xác
 - C. Không đơn vị
 - D. Tất cả đều sai
6. Trước khi bắt đầu việc nhận dạng, hình ảnh được đổi thành kích thước bao nhiêu?
 - A. 100x100
 - B. 224x224
 - C. Tùy thuộc vào ảnh đầu vào
 - D. Tất cả đều sai
7. Ảnh đầu vào để nhận dạng cần có kích thước bao nhiêu?
 - A. 224x224
 - B. Lớn hơn 224x224
 - C. Bao nhiêu cũng được
 - D. Tất cả đều đúng

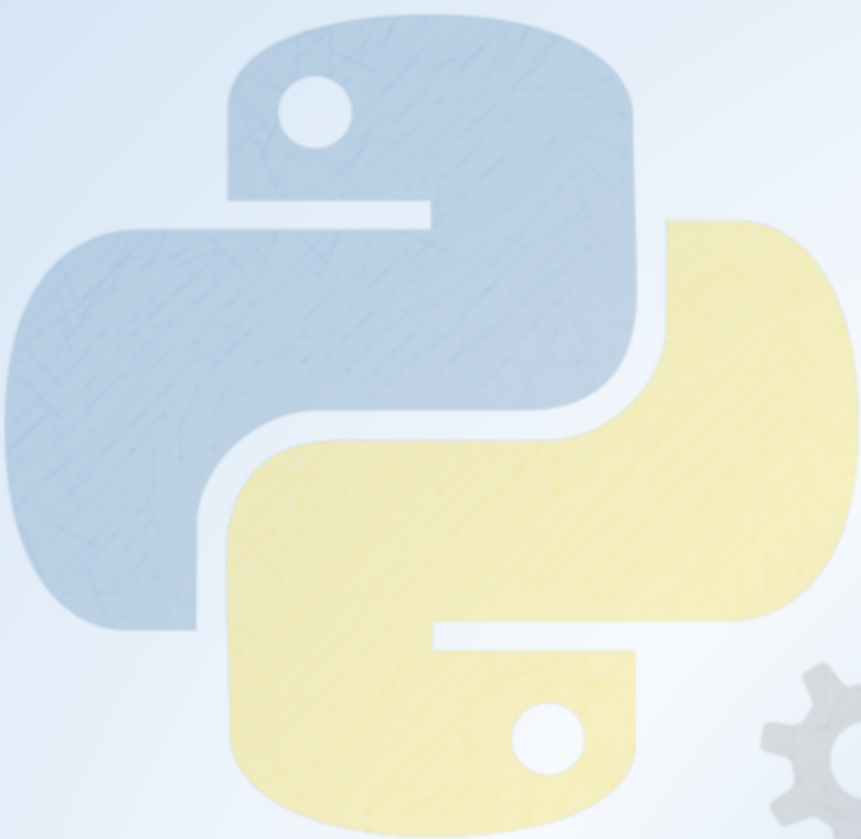
Đáp án

1. B 2. D 3. B 4. C 5. A 6. B 7. B



CHƯƠNG 3

Tích hợp Webcam để nhận diện hình ảnh trên Python



Pandas



1 Giới thiệu

Trong bài hướng dẫn trước, chúng ta đã có thể chạy chương trình nhận dạng từ một ảnh tĩnh. Tuy nhiên để có thể xây dựng nên một dự án sinh động hơn, chúng ta cần phải lấy ảnh động từ webcam có sẵn trên máy tính. Ý tưởng ở đây là định kì chụp ảnh rồi mới nhận dạng. Quá trình này sẽ diễn ra liên tục cho đến khi hệ thống bị tắt hoặc dừng lại. Thêm nữa, khi nhận dạng thành công một đối tượng đã được huấn luyện, chúng ta sẽ dùng loa của máy tính để thông báo, thay vì chỉ in kết quả ra màn hình.

Bài hướng dẫn này sẽ tập trung vào 2 kĩ thuật chính. Thứ nhất là tích hợp việc chụp ảnh động từ Webcam cho việc nhận dạng. Tiếp theo, là hiện thực chương trình phát ra giọng nói bằng tiếng Việt, để tăng tính thân thiện của chương trình với người dùng. Các kĩ năng chính của bài hướng dẫn này được tóm tắt như sau:

- Thêm thư viện - Cài đặt thư viện cho dự án
- Đóng gói chương trình thành hàm
- Tích hợp chụp ảnh từ Webcam
- Tích hợp giọng nói tiếng Việt

2 Tích hợp Webcam

Bước 1: Thêm thư viện cv2 và hiện thực hàm `captureImage`, như gợi ý sau đây:

```
1 import cv2
2
3 cam = cv2.VideoCapture(0)
4
5 def capture_image():
6     ret, frame = cam.read()
7     cv2.imwrite('img_detect.png', frame)
```

Chương trình 3.1: Thêm thư viện cv2 và chụp ảnh từ Webcam

Để có thể thêm thư viện cv2 thành công, bạn cũng cần phải cài đặt nó. Bằng cách rê chuột vào tên thư viện trong câu lệnh `import`, gợi ý cài đặt sẽ được PyCharm hiện ra. Cũng như việc cài đặt PIL ở bài trước, với cv2 bạn cũng phải chọn vào More actions để có thể cài đặt một thư viện khác tương đương.

Tiếp theo, hàm `captureImage()` chỉ đơn giản là chụp một hình từ webcam và lưu nó lại dưới tên **img_detect.png**.

Bước 2: Đóng gói chương trình ở bài trước thành hàm `face_detection`, kết quả của chương trình sẽ như sau:

```
1 def face_detection():
2     # Disable scientific notation for clarity
3     np.set_printoptions(suppress=True)
```

```

4
5     # Load the model
6     model = tensorflow.keras.models.load_model('keras_model
7         .h5')
8
9     # Create the keras model
10    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.
11        float32)
12
13    # Replace this with the path to your image
14    image = Image.open('img_dectect.png')
15
16    #resize the image to a 224x224
17    size = (224, 224)
18    image = ImageOps.fit(image, size, Image.ANTIALIAS)
19
20    #turn the image into a numpy array
21    image_array = np.asarray(image)
22
23    # display the resized image
24    image.show()
25
26    # Normalize the image
27    normalized_image_array = (image_array.astype(np.float32
28        ) / 127.0) - 1
29
30    # Load the image into the array
31    data[0] = normalized_image_array
32
33    prediction = model.predict(data)
34    print(prediction)
35
36    name = ["Quynh Anh", "Phan Phan", "Anh Nen"]
37    index = -1
38    max_value = -1
39    for i in range(0, len(prediction[0])):
40        if max_value < prediction[0][i]:
41            max_value = prediction[0][i]
42            index = i
43    print("Ket Qua: ", name[index])
44    print ("Chinh Xac: ", max_value)

```

Chương trình 3.2: Đóng gói chương trình thành hàm face_detection

Lưu ý: Trong chương trình trên, chúng ta cần phải sửa lại tên hình ảnh để nhận dạng thành **img_detect.png**

Bước 3: Hiện thực thêm chương trình thực thi, gồm 2 bước là chụp hình và nhận dạng, như sau:

```

1 while True:
2     capture_image()
3     face_detection()

```

Chương trình 3.3: Hoàn thiện chương trình

Ở bước này, chúng ta có thể thấy một đặc trưng vô cùng quan trọng của một dự án: nó cần một **vòng lặp vô tận**. Bởi đơn giản, chương trình kết thúc cũng có nghĩa là ứng dụng đã dừng. Trong các dự án thực tế, chương trình phải luôn luôn chạy cho đến khi nào người dùng chủ động tắt nó đi (ví dụ như tắt nguồn điện chẳng hạn). Đây là điều khác biệt rất lớn đối với các chương trình mang tính chất nghiên cứu, khi **chương trình chỉ cần chạy một lần** rồi thôi.

3 Tích hợp trợ lý ảo

Bước 1: Cài đặt thư viện hỗ trợ. Để có thể phát ra giọng nói tiếng Việt, chúng ta phải cài đặt khá nhiều thư viện, như liệt kê bên dưới:

```

1 import os
2 import playsound
3 import wikipedia
4 from gtts import gTTS
5 from webdriver_manager.chrome import ChromeDriverManager

```

Chương trình 3.4: Thư viện cho giọng nói tiếng Việt

Trừ thư viện cuối cùng, các thư viện còn lại bạn đọc có thể tận dụng gợi ý từ Py-Charm. Sau đó chuyển sang cửa sổ Terminal để cài đặt thư viện cuối cùng bằng câu lệnh **pip install webdriver-manager**. Bạn đọc hãy lưu ý dấu - trong câu lệnh pip install.

Bước 2: Hiện thực các câu lệnh khởi tạo cho giọng nói tiếng Việt. Các câu lệnh này có thể đặt phía sau câu lệnh khởi tạo Camera ở phần trước, như sau:

```

1 wikipedia.set_lang('vi')
2 language = 'vi'
3 path = ChromeDriverManager().install()

```

Chương trình 3.5: Khởi tạo giọng nói tiếng Việt

Bước 3: Hiện thực hàm speak, nhận tham số đầu vào là một chuỗi, và phát âm ra tiếng Việt, như sau:

```

1 tts = gTTS(text=text, lang=language, slow=False)
2 tts.save("sound.mp3")
3 playsound.playsound("sound.mp3", True)
4 os.remove("sound.mp3")

```

Chương trình 3.6: Hiện thực hàm phát âm tiếng Việt

Ở đây, phần phát ra âm tiếng Việt sử dụng thư viện của Google (thư viện gtts - Google Text To Speech). Cuối cùng, chúng ta sẽ gọi hàm này ở cuối hàm face_detect,

sau khi nhận dạng ra vật thể. Ở lần đầu tiên chạy chương trình, có thể sẽ phải mất thời gian để PyCharm thiết lập môi trường ảo. Sau đó, chương trình sẽ chạy nhanh hơn. Hình ảnh toàn bộ chương trình lúc này của chúng ta như sau:

```
1 import tensorflow.keras
2 from PIL import Image, ImageOps
3 import numpy as np
4
5 import cv2
6
7 import os
8 import playsound
9 import wikipedia
10 from gtts import gTTS
11 from webdriver_manager.chrome import ChromeDriverManager
12
13 wikipedia.set_lang('vi')
14 language = 'vi'
15 path = ChromeDriverManager().install()
16
17 def speak(text):
18     tts = gTTS(text=text, lang=language, slow=False)
19     tts.save("sound.mp3")
20     playsound.playsound("sound.mp3", True)
21     os.remove("sound.mp3")
22
23
24 cam = cv2.VideoCapture(0)
25
26 def capture_image():
27     ret, frame = cam.read()
28     cv2.imwrite('img_detect.png', frame)
29
30
31 def face_detection():
32     # Disable scientific notation for clarity
33     np.set_printoptions(suppress=True)
34
35     # Load the model
36     model = tensorflow.keras.models.load_model('keras_model.h5')
37
38     # Create the array of the right shape to feed into the
39     # keras model
40     data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
41
42     # Replace this with the path to your image
43     image = Image.open('img_detect.png')
```

```

44     #resize the image to a 224x224 with the same strategy
    as in TM2:
45     #resizing the image to be at least 224x224 and then
    cropping from the center
46     size = (224, 224)
47     image = ImageOps.fit(image, size, Image.ANTIALIAS)
48
49     #turn the image into a numpy array
50     image_array = np.asarray(image)
51
52     # display the resized image
53     image.show()
54
55     # Normalize the image
56     normalized_image_array = (image_array.astype(np.float32)
    / 127.0) - 1
57
58     # Load the image into the array
59     data[0] = normalized_image_array
60
61     # run the inference
62     prediction = model.predict(data)
63     print(prediction)
64
65     name = ["Quynh Anh", "Phan Phan", "Anh nen"]
66     index = -1
67     max_value = -1
68     for i in range(0, len(prediction[0])):
69         if max_value < prediction[0][i]:
70             max_value = prediction[0][i]
71             index = i
72     print("Ket Qua: ", name[index])
73     print("Chinh Xac: ", max_value)
74
75     speak("Xin chao " + name[index])
76
77 while True:
78     capture_image()
79     face_detection()

```

Chương trình 3.7: Dự án nhận dạng khuôn mặt - Giọng nói tiếng Việt

Bạn đọc có thể dựa vào dự án này để phát triển thêm nhiều dự án tương tự chẳng hạn như phân loại cà chua đã chín và còn sống, rau sạch và rau đang bị sâu. Độ chính xác của hệ thống sẽ phụ thuộc vào tập dữ liệu nhận dạng. Bạn đọc cần lưu ý khi chọn lọc dữ liệu cho việc huấn luyện, đặc biệt là dữ liệu sai.

4 Câu hỏi ôn tập

1. Việc nhận dạng khuôn mặt từ Webcam được xây dựng dựa trên ý tưởng nào?
 - A. Nhận dạng trực tiếp từ video của Webcam
 - B. Tận dụng khả năng tự focus vào khuôn mặt của Webcam
 - C. Chụp từng hình rồi nhận dạng
 - D. Tất cả đều sai
2. Thư viện cho việc lấy hình ảnh từ Webcam là gì?
 - A. webcam2
 - B. cv2
 - C. opencv
 - D. gtts
3. Thư viện cho việc chính cho việc phát ra âm tiếng Việt là gì?
 - A. webcam2
 - B. cv2
 - C. opencv
 - D. gtts
4. Nguyên nhân chính ảnh hưởng đến tốc độ thực thi của chương trình là gì?
 - A. Máy tính cấu hình thấp
 - B. Camera độ phân giải thấp
 - C. Môi trường ảo PyCharm có hiệu suất thấp
 - D. Tất cả các nguyên nhân trên
5. Các ứng dụng tiềm năng của kỹ thuật phân loại hình ảnh là gì?
 - A. Phân biệt cà chua sống và cà chua chín
 - B. Phân biệt rau sạch và rau bị sâu
 - C. Phân biệt củ khoai tây tới hạn và khoai tây còn non
 - D. Tất cả các ứng dụng trên

Đáp án

1. C 2. B 3. D 4. C 5. D

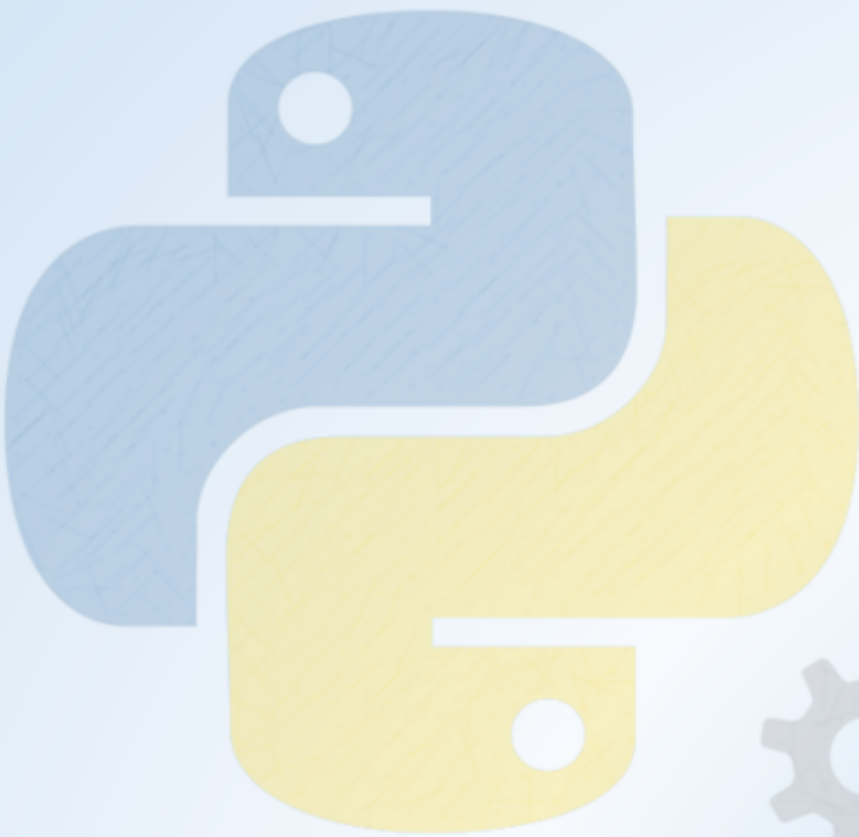
Phần II

Nhận dạng âm thanh với Teachable Machine



CHƯƠNG 4

Huấn luyện âm thanh trên Teachable Machine



Pandas



1 Giới thiệu

Công cụ thứ 2 trong bộ công cụ Học máy với Google liên quan đến nhận dạng âm thanh. Đây thực sự là một công nghệ rất cập nhật mà Google cung cấp cho chúng ta. Sử dụng công cụ này, chúng ta có thể cho hệ thống nhận dạng những lệnh ngắn, dưới 2 giây, để làm một tác vụ nào đó.

Ban đầu, bạn đọc có thể nghĩ rằng 2 giây là quá ít để ghi âm điều gì đó. Tuy nhiên, đây là công nghệ để xây dựng nên từ lệnh cho các hệ thống nhà thông minh (Google Home) hoặc trợ lý ảo (Google Assistant) trên điện thoại. Khi cần kích hoạt những hệ thống này, bạn thường hay nói "OK Google".

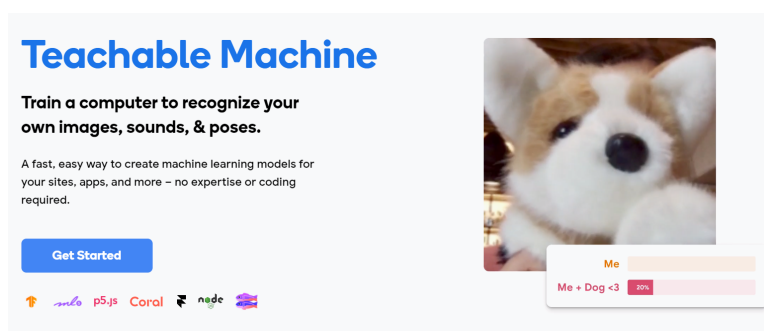
Hiện tại, đa phần các thiết bị điện tử, từ máy tính cho tới tivi thông minh, đều rất ít khi được tắt nguồn hoàn toàn. Nó chỉ ở chế độ chờ (stand by) và sẽ kích hoạt trở lại khi có tín hiệu gửi đến. Do đó, giọng nói theo kiểu ra lệnh sẽ là hướng tiếp cận cho các ứng dụng như vậy.

Bài hướng dẫn này sẽ tập trung vào các mục tiêu sau đây:

- Sử dụng được công cụ Học máy với Google liên quan tới âm thanh
- Hiểu được các khái niệm âm thanh nền, âm thanh lệnh
- Huấn luyện và chạy thử nghiệm một hệ thống trí tuệ nhân tạo về âm thanh

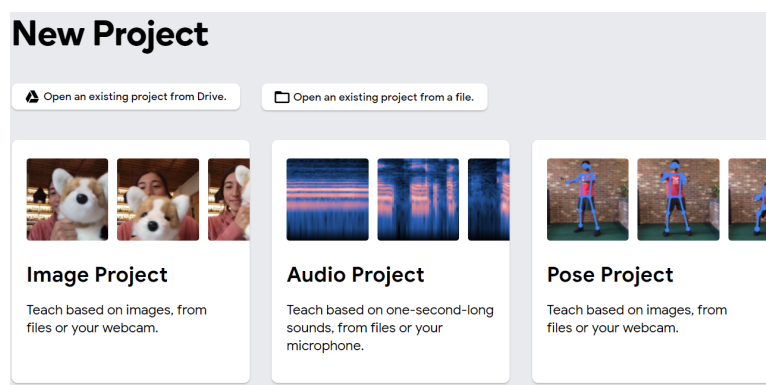
2 Huấn luyện hệ thống âm thanh

Học Máy Google đã được xây dựng thành một công cụ trực tuyến. Chúng ta bắt đầu bằng việc vào đường dẫn <https://teachablemachine.withgoogle.com/>, giao diện sau đây sẽ hiện ra:



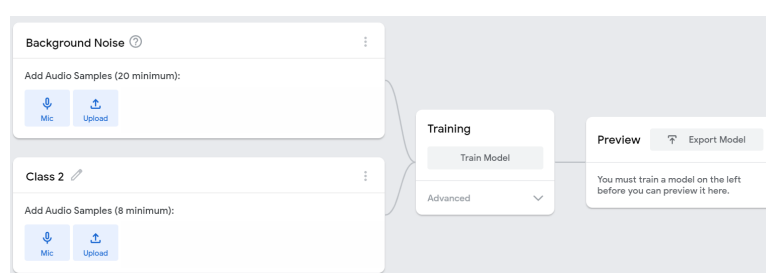
Hình 4.1: Trang web online của Học máy Google

Tại trang web này, chúng ta có thể xem trước một số hình ảnh cũng như video giới thiệu của Google về công nghệ Học máy. Tiếp theo, nhấn vào nút **Get Started** để bắt đầu, giao diện của công cụ học máy với Google được hiện ra như sau:



Hình 4.2: Công cụ Học máy với Google - Audio Project

Lần này, chúng ta sẽ chọn vào **Audio Project** để bắt đầu xây dựng hệ thống trí tuệ nhân tạo cho âm thanh. Giao diện sau đây sẽ hiện ra:



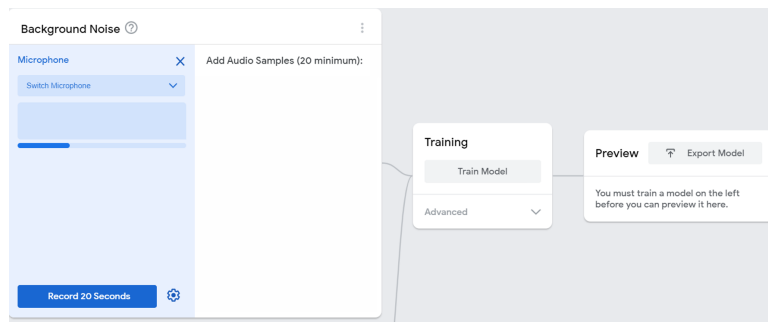
Hình 4.3: Giao diện để huấn luyện âm thanh

Bước 1: Cung cấp âm thanh nền

Ở giao diện này, trước khi huấn luyện âm thanh, chúng ta cần phải cung cấp âm thanh nền (Background Noise), hay còn gọi là âm nhiễu. Thực sự đây là điều không dễ dàng vì âm nhiễu thường không thể xác định và cũng không có quy luật. Tuy nhiên đó là trường hợp khi chúng ta muốn phát hiện ra âm khi nói nhỏ. Trong trường hợp nói lớn hơn một chút thì ảnh hưởng của âm nền sẽ giảm đi.

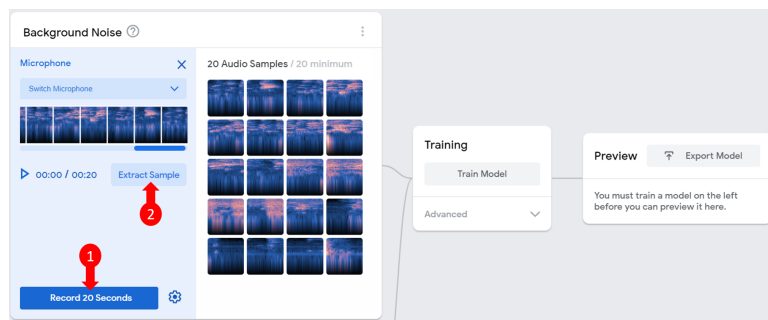
Chúng ta cũng có 2 cách để cung cấp âm thanh nền: Thu âm trực tiếp từ micro hoặc là tải âm nền đã có lên hệ thống huấn luyện. Đối với âm thanh, việc **tải một âm có sẵn lên hệ thống sẽ không tối ưu để huấn luyện**, đơn giản nó sẽ khác rất nhiều với âm thu từ micro của hệ thống. Với cách thứ nhất, giao diện sau đây sẽ hiện lên như sau:

Mặc định, hệ thống sẽ ghi âm 20 giây cho âm nền. Bạn chỉ cần nhấn vào nút **Record for 20 second** là hệ thống bắt đầu ghi âm. Trong trường hợp muốn ghi âm nhiều hơn, bạn có thể nhấn vào nút Setting ở bên cạnh để thay đổi.



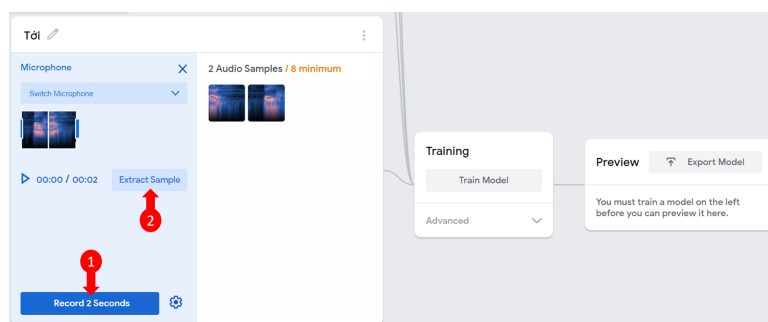
Hình 4.4: Cung cấp dữ liệu cho âm thanh nền

Sau khi ghi âm xong, bạn phải nhấn thêm vào nút Extract Sample để phân tách âm nền ra nhiều mẫu âm thanh nhỏ, mỗi âm thanh là 1 giây, như kết quả ở hình bên dưới:



Hình 4.5: Hoàn thành dữ liệu cho âm nền

Tiếp theo, chúng ta sẽ cung cấp âm thanh cho từ lệnh cần huấn luyện. Các thao tác ở bước này hoàn toàn tương tự như âm thanh nền, chỉ khác là lần này chúng ta sẽ thu mỗi 2 giây, và lưu ý là phải nhấn vào nút **Extract Sample** thì dữ liệu mới được lưu lại cho việc huấn luyện, như minh họa ở hình bên dưới:

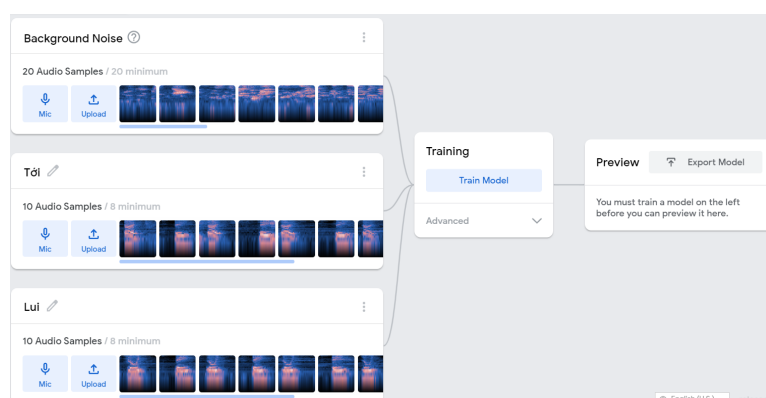


Hình 4.6: Cung cấp dữ liệu cho từ khóa Tôi

Bước 2: Cung cấp dữ liệu cho từ lệnh

Bạn đọc có thể làm tương tự, để xây dựng hệ thống âm thanh gồm có 4 từ lệnh là Tôi, Lui, Trái và Phải. Ở đây, theo kinh nghiệm của chúng tôi, bạn hãy **nói từ lệnh khớp trong khung hình 1 giây** để việc nhận dạng sau này sẽ chính xác và nhạy hơn. Khác với hình ảnh, âm thanh có sự liên quan giữa dữ liệu trước và sau. Do đó, việc thu từ lệnh ngắn sẽ giảm tải rất nhiều cho hệ thống lúc vận hành. Kết quả cho

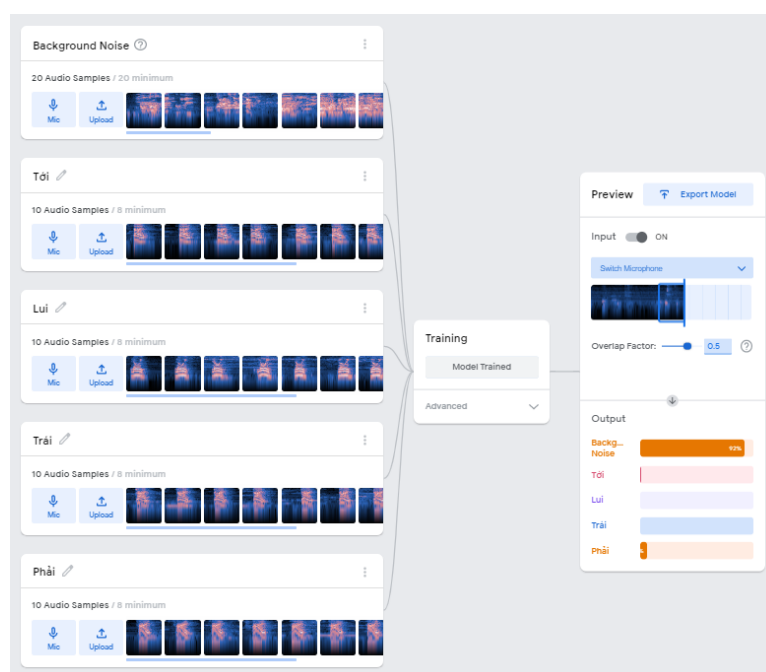
đến bước này sẽ như minh hoạt ở hình bên trên.



Hình 4.7: Toàn bộ hệ thống với 4 từ lệnh Tới, Lui, Trái và Phải

Bước 3: Huấn luyện hệ thống

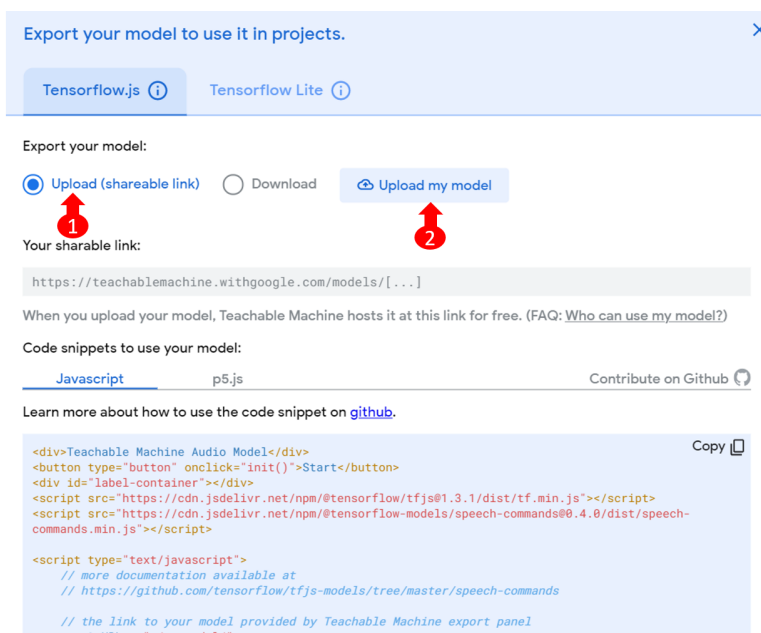
Nhấn nút Train để bắt đầu huấn luyện tập dữ liệu âm thanh. Sau khi quá trình này kết thúc, bạn đã có thể dùng thử hệ thống, như minh họa ở hình bên dưới:



Hình 4.8: Kiểm tra hệ thống nhận diện âm thanh

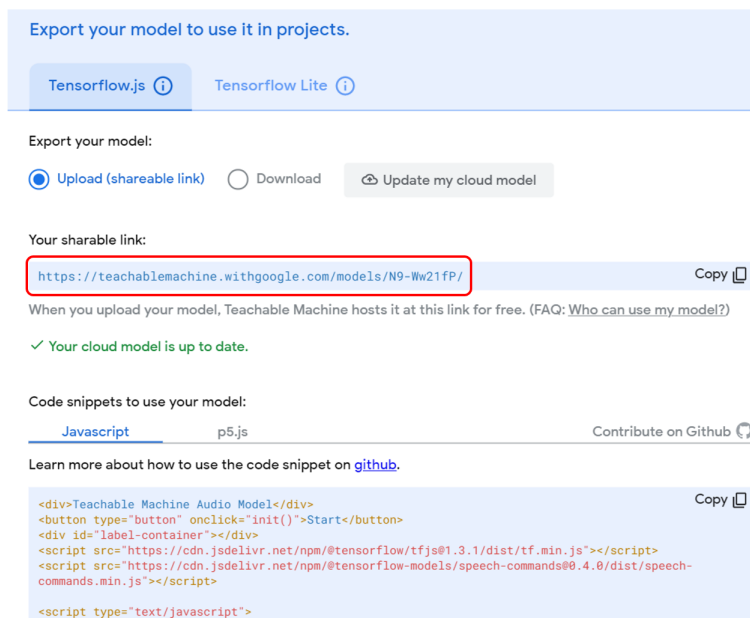
3 Xuất kết quả huấn luyện

Tương tự như phần huấn luyện hình ảnh, chúng ta cũng cần phải xuất kết quả huấn luyện ra những định dạng khác để có thể tích hợp vào nhiều nền tảng khác nhau, như lập trình trên Scratch hoặc Python chẳng hạn. Bằng cách nhấn vào nút **Export Model**, giao diện sau đây sẽ hiện ra:



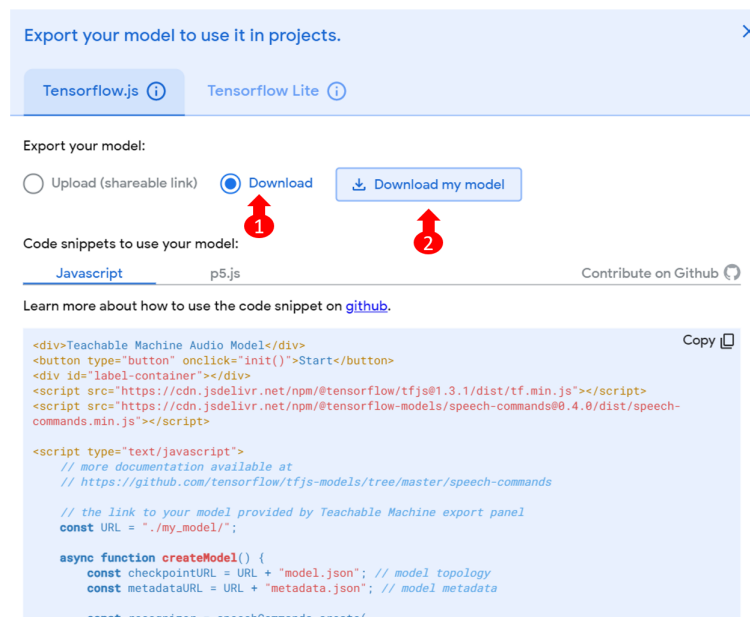
Hình 4.9: Tải kết quả huấn luyện lên server Google

Chúng ta sẽ chọn vào mục và chọn Upload (shareable link) và chọn tiếp vào **Upload my model** như minh họa ở hình bên dưới. Chức năng này sẽ tải toàn bộ kết quả huấn luyện lên server của Google và trả về cho chúng ta một đường dẫn để lập trình trong môi trường Scratch. Khi việc tải lên hoàn tất, màn hình dưới đây sẽ hiện ra và bạn có thể sao chép lại đường dẫn ở mục **Your shareable link**.



Hình 4.10: Hệ thống đã được tải thành công lên servero Google

Cuối cùng, để có thể sử dụng mô hình vào các ngôn ngữ lập trình cấp cao khác như Python, chúng ta cần phải tải về dưới dạng file và dùng offline. Hướng dẫn cho quá trình này được minh họa như hình bên dưới:

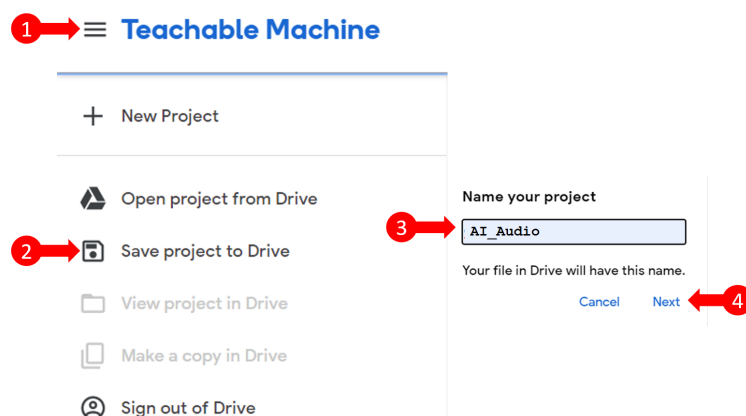


Hình 4.11: Tải mô hình về dưới dạng file nén

File nén tải về sẽ được sử dụng khi chúng ta lập trình nhận dạng âm thanh bằng Python. Cần lưu ý là với dự án Audio, Google không hỗ trợ định dạng Tensorflow. Do đó, khi sử dụng với ngôn ngữ Python, chúng ta phải chuyển file nén này thêm một bước nữa. Các thao tác này sẽ được hướng dẫn sau.

4 Lưu dự án vào Google drive

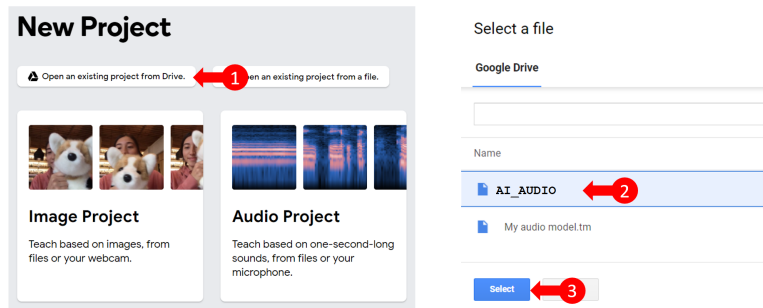
Với những hỗ trợ toàn diện từ phía Google, hệ thống AI của bạn có thể được lưu lại trực tuyến trên Google drive. Bằng các trở lại màn hình chính và làm theo 4 bước hướng dẫn ở hình bên dưới như: Chọn Save project to Drive, Đặt tên cho project và cuối cùng nhấn nút Next.



Hình 4.12: Lưu dự án vào Google drive

5 Mở dự án từ Google drive

Trong trường hợp muốn mở lại dự án cũ để cập nhật hoặc làm tiếp, từ trang chủ của Học Máy Google, chúng ta chọn **Open existing project from Drive** rồi lựa chọn trong danh sách các dự án đã lưu, như minh họa ở hình bên dưới.



Hình 4.13: Mở dự án từ Google drive

6 Câu hỏi ôn tập

1. Công cụ huấn luyện âm thanh có tên là gì?
A. Teachable Machine
B. Artificial Intelligence
C. Video Project
D. Audio Project
2. Đối tượng đầu tiên trong công cụ huấn luyện audio là gì?
A. Âm thanh nền (Background Noise)
B. Class 1
C. Class 2
D. Tất cả đều sai
3. Dữ liệu để huấn luyện cho hệ thống nhận dạng bằng ảnh có thể được lấy từ nguồn nào?
A. Trực tiếp từ micro của máy tính
B. Tải âm thanh từ máy tính lên
C. Ảnh từ Google Drive của người dùng
D. Tất cả các nguồn âm thanh trên
4. Trong các nguồn âm thanh dưới đây, nguồn âm nào là tốt nhất để huấn luyện hệ thống?
A. Micro của máy tính
B. Tải âm thanh từ máy tính lên
C. Ảnh từ Google Drive của người dùng
D. Không thể so sánh được
5. Đường dẫn trực tuyến của mô hình huấn luyện âm thanh hỗ trợ cho môi trường lập trình nào sau đây?
A. Scratch 2.0
B. Scratch 3.0
C. Python
D. Scratch 3.0 và Python
6. Các từ lệnh trong hệ thống huấn luyện âm thanh có độ dài tốt nhất là:
A. 1 giây
B. 2 giây
C. 20 giây
D. Tất cả đều sai
7. Khi huấn luyện âm thanh nền, mặc định hệ thống cần bao nhiêu giây?
A. 1 giây
B. 2 giây
C. 20 giây
D. Tất cả đều sai

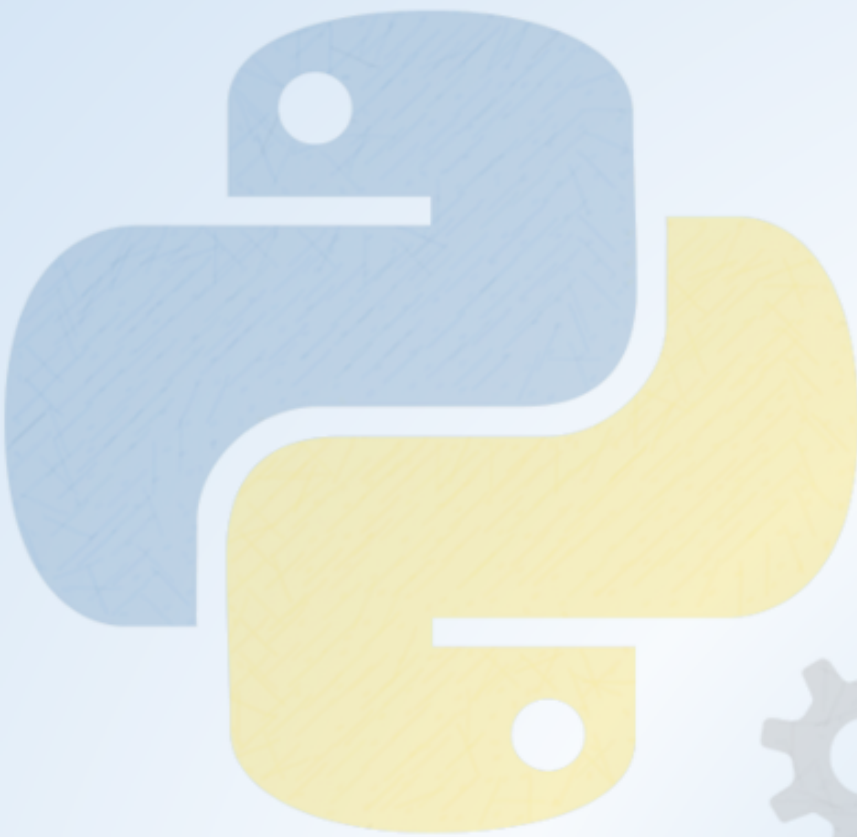
Đáp án

1. D 2. A 3. D 4. A 5. B 6. A 7. C



CHƯƠNG 5

AI và Robot trên Scratch 3.0



Pandas



1 Giới thiệu

Mô hình huấn luyện với Google có độ tương thích tốt với nhiều hệ thống lập trình, đặc biệt là trong giáo dục. Trong bài hướng dẫn này, chúng ta sẽ kết hợp mô hình huấn luyện Google với Scratch 3.0 phiên bản trực tuyến. Thực ra, Scratch 3.0 đang là mã nguồn mở cho rất nhiều tổ chức và cá nhân tùy biến lại môi trường lập trình của nó, để tích hợp thêm nhiều công cụ mới theo thời đại 4.0.

Trong bài hướng dẫn này, chúng tôi chọn Scratch 3.0 được phát triển bởi học viện MIT, với những tích hợp sẵn về Học máy với Google và Robot sử dụng mạch Microbit. Chúng ta sẽ viết một chương trình điều khiển bằng giọng nói để điều khiển việc di chuyển của một Robot ảo trên môi trường lập trình và thậm chí là một robot thật. Các mục tiêu của bài hướng dẫn này như sau:

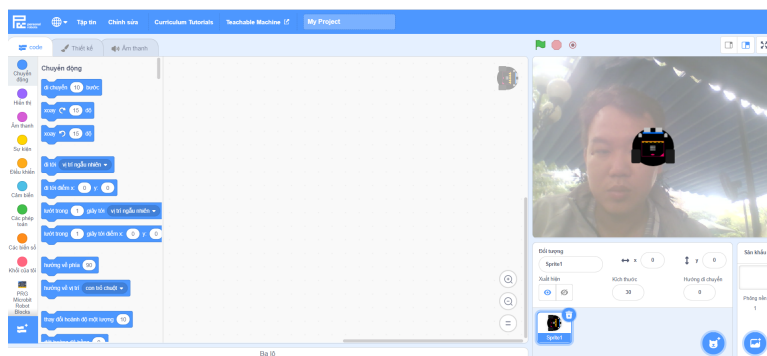
- Làm quen với môi trường Scratch 3.0 trực tuyến của MIT
- Tìm hiểu và áp dụng công cụ Học máy với Google
- Kết nối với Robot Microbit
- Viết chương trình tổng hợp các công nghệ trên

2 Scratch 3.0 trực tuyến

Lý do chính để chúng tôi giới thiệu với bạn đọc về phần mềm trực tuyến là vì các hỗ trợ thêm của nó về các công nghệ mới liên quan đến Robot và Trí tuệ nhân tạo. Để bắt đầu, chúng ta sẽ vào đường dẫn sau đây:

<https://mitmedialab.github.io/prg-extension-boilerplate/http/>

Giao diện về phần mềm Scratch 3.0 sẽ được hiện ra như sau:

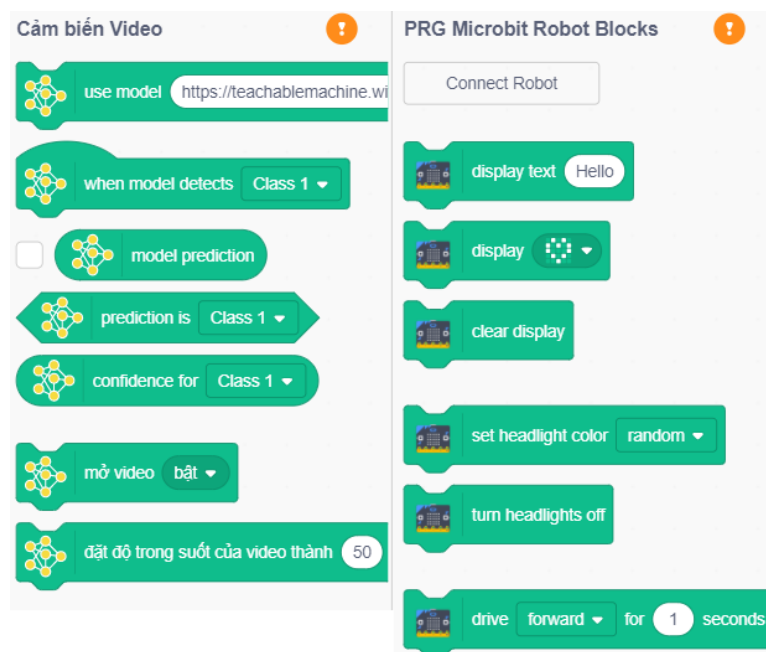


Hình 5.1: Giao diện lập trình Scratch 3.0 trực tuyến

Điều khác biệt đầu tiên là giao diện này sẽ **truy cập trực tiếp vào webcam và cả micro** của máy tính. Do đó, trong lần đầu mở lên, sẽ có thông báo để người dùng cho phép quyền truy cập vào thiết bị này. Trong trường hợp bạn lỡ tay không cho phép, chỉ cần tắt trang web này đi và truy cập lại lần nữa. Hiện tại, trang lập trình

trực tuyến này hỗ trợ tốt nhất trên trình duyệt Chrome.

Thứ 2, bên cạnh những khối lập trình cơ bản của Scratch 3.0, giao diện lập trình này tự động thêm vào 2 khối lập trình mới, là **Cảm biến Video** và **Microbit Robot**, như minh họa ở hình bên dưới:



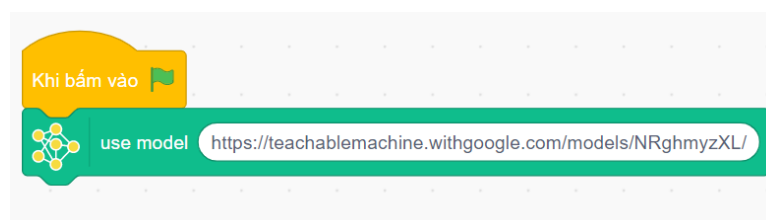
Hình 5.2: Các câu lệnh liên quan tới Trí tuệ nhân tạo và Robot

Đây là những cập nhật rất mới, giúp chúng ta tiếp cận với công nghệ trí tuệ nhân tạo cũng như Robotics. Thuật ngữ Cảm biến Video thực chất là Học máy với Google mà chúng ta đã huấn luyện ở các bài trước.

3 Trí tuệ nhân tạo trên Scratch 3.0

3.1 Kích hoạt trí tuệ nhân tạo

Với tiêu chí lập trình đơn giản, Scratch 3.0 hỗ trợ tích hợp hệ thống Trí tuệ nhân tạo chỉ với 1 câu lệnh là use model, với tham số là đường link của hệ thống mà bạn đã huấn luyện ở bài trước. Chúng ta hãy bắt đầu với sự kiện lá cờ màu xanh, như minh họa ở hình bên dưới:



Hình 5.3: Liên kết với mô hình trí tuệ nhân tạo

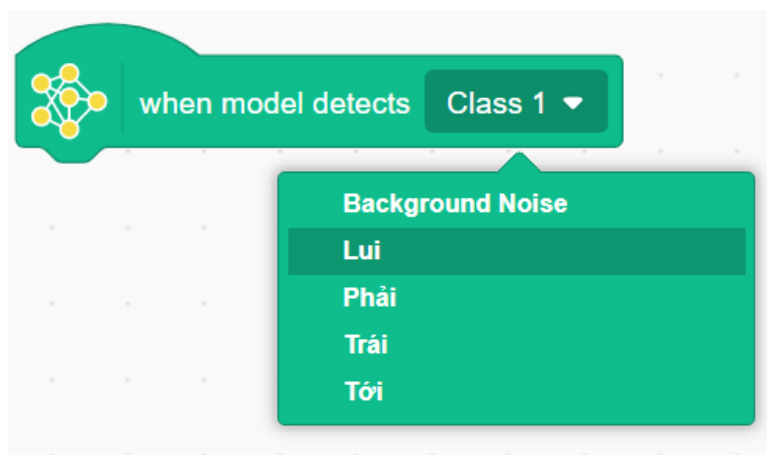
Ở đây, chúng tôi xài mô hình đã được huấn luyện sẵn và lưu trữ ở đường dẫn sau đây, với 4 từ khóa là Tới, Lui, Trái và Phải đã được huấn luyện:

<https://teachablemachine.withgoogle.com/models/NRghmyzXL/>

Một đặc điểm vô cùng thuận tiện là bạn có thể huấn luyện lại mô hình này và đường dẫn của nó vẫn không thay đổi. Bạn hãy nhấn vào lá cờ để thực thi câu lệnh này. Khi câu lệnh này thực thi, bạn sẽ thấy hình ảnh trên giao diện bị dừng lại một chút. Khoảng dừng này là để tải mô hình huấn luyện này về máy. Đây là điều quan trọng để chúng ta bước sang phần thứ 2.

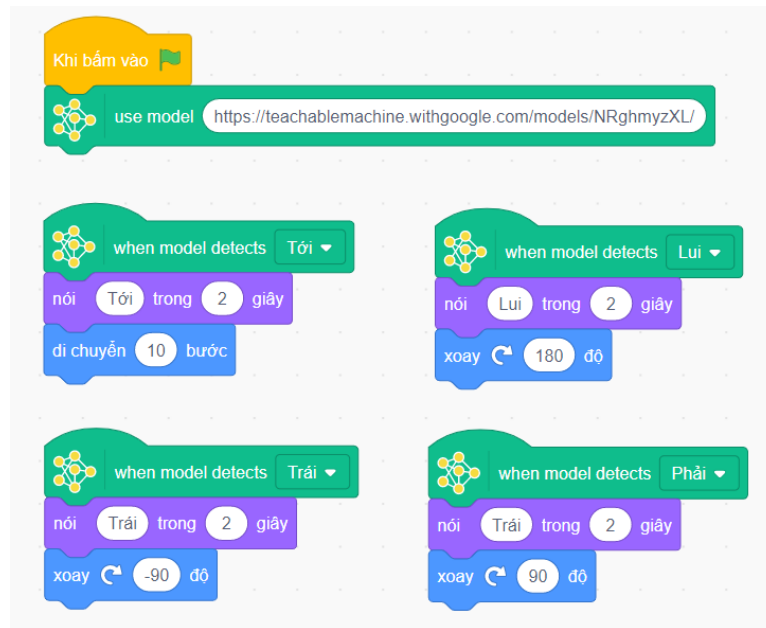
3.2 Bắt đầu việc nhận dạng

Khi câu lệnh ở phần trên được thực thi thành công, câu lệnh tiếp theo mới có thể được sử dụng. Các đối tượng được huấn luyện lúc này mới xuất hiện trong tùy chọn của câu lệnh, như minh họa ở hình bên dưới:



Hình 5.4: Các đối tượng được huấn luyện xuất hiện trong tùy chọn

Trong trường hợp mà câu lệnh của bạn chưa đúng với mong muốn, hãy kiên nhẫn khoảng 5 giây rồi nhấn vào lá cờ màu xanh để liên kết lại. Cuối cùng, một chương trình gợi ý cho việc di chuyển của Robot bằng giọng nói sẽ như sau:



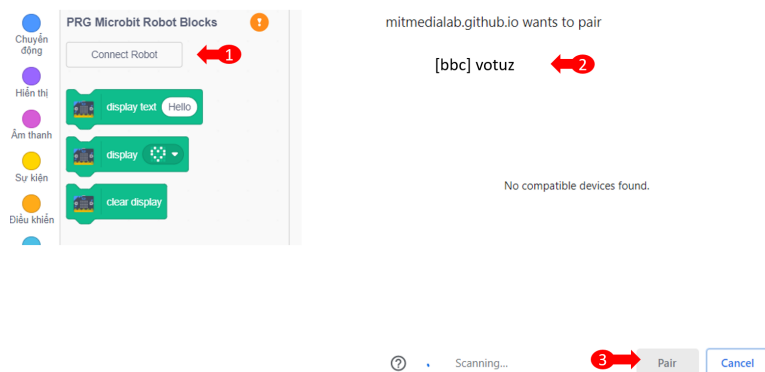
Hình 5.5: Điều khiển di chuyển của Robot bằng giọng nói

Đến đây, bạn có thể thử nói để điều khiển Robot di chuyển. Trong trường hợp cảm thấy việc nhận dạng không tốt, bạn đọc có thể huấn luyện lại mô hình.

4 Kết nối với Robot Microbit

Scratch 3.0 hỗ trợ kết nối với một Robot sử dụng mạch Microbit qua kết nối Bluetooth. Chuẩn kết nối Bluetooth 4.0 này sẽ đặc biệt hơn nhiều so với kết nối cũ: bạn không cần phải ghép đôi 2 thiết bị. Thoạt nhìn ban đầu, bạn sẽ nghĩ điều này không bảo mật. Tuy nhiên, mức độ bảo mật của nó cao hơn rất nhiều và chỉ có thể kết nối được bằng phần mềm chuyên dụng. Thêm nữa, cơ chế giao tiếp mới cho phép việc gửi dữ liệu không dây xa hơn và ổn định hơn.

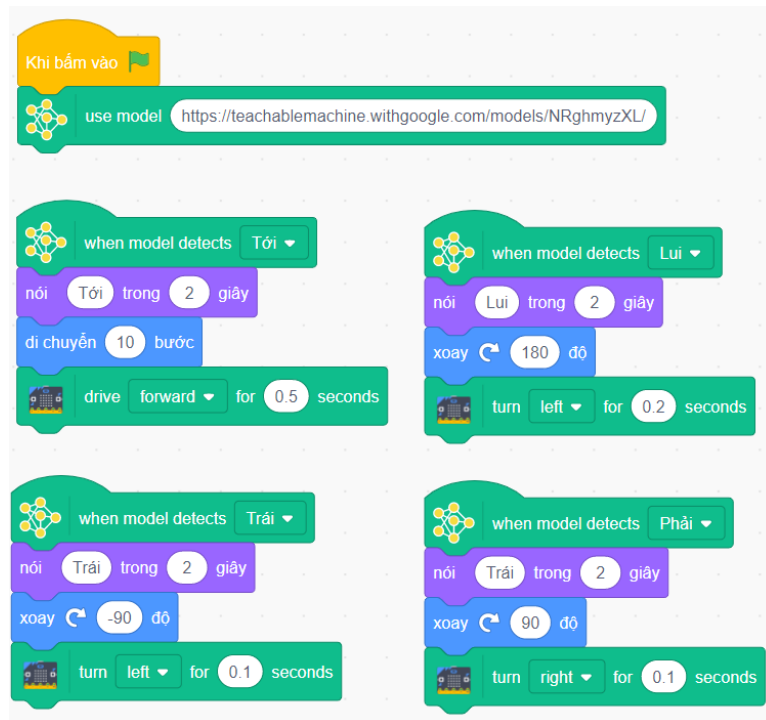
Để kết nối với Robot, chúng ta sẽ nhấn vào nút **Connect Robot** sau đó chọn và danh sách tên của Robot và nhấn nút Pair, như hướng dẫn bên dưới:



Hình 5.6: Kết nối với Robot Microbit

Lưu ý: Khi vừa khởi động Robot, tên của nó sẽ xuất hiện trên màn hình 25 đèn. Bạn hãy ghi lại tên của nó để kết nối cho đúng trong trường hợp có nhiều Robot xuất hiện trong danh sách.

Cuối cùng, chúng ta sẽ bổ sung thêm các lệnh di chuyển liên quan đến Robot, để một Robot thật có thể di chuyển mỗi khi chúng ta ra lệnh bằng giọng nói. Gọi ý chương trình cho bạn đọc sẽ như sau:



Hình 5.7: Thêm các câu lệnh di chuyển cho Robot

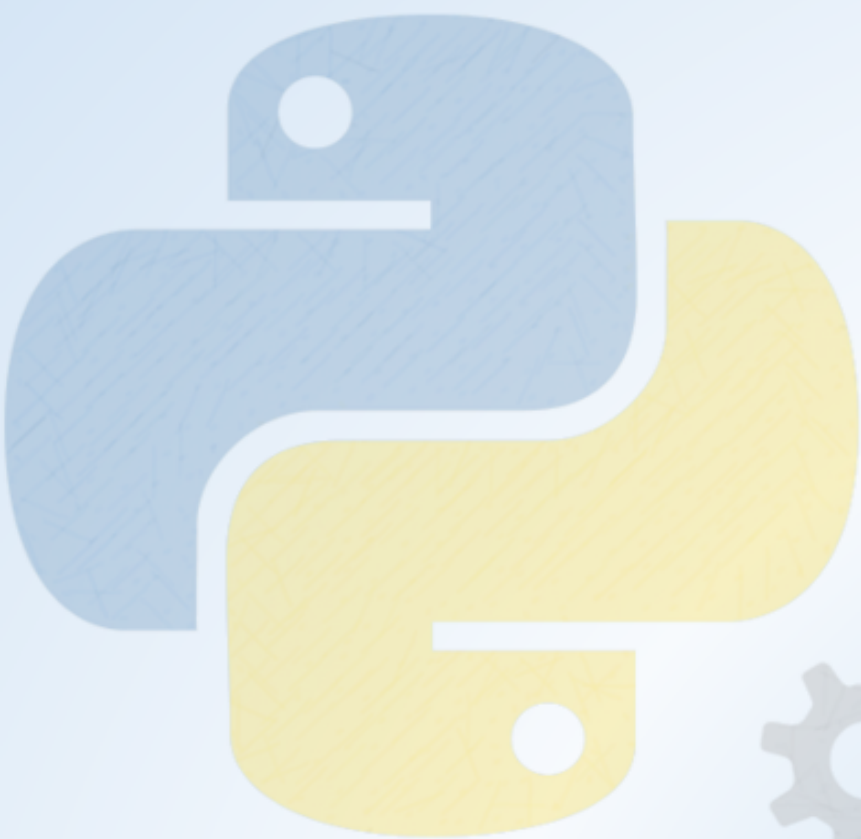
Tất nhiên, việc di chuyển của Robot có chính xác hay không phụ thuộc rất nhiều vào kết cấu cơ khí của nó. Ở đây, chúng tôi chỉ tinh chỉnh việc di chuyển của nó bằng yếu tố thời gian. Ý nghĩa ở đây là robot sẽ xoay trái trong 0.1 giây, và chúng ta mong đợi là sau 0.1 giây, nó sẽ hướng về bên trái. Những sai số chắc chắn sẽ có và sẽ thay đổi. Chỉ với những cơ khí Robot phức tạp mới có những cảm biến, ví dụ như la bàn, hay cảm biến góc xoay, thì lúc đó nó mới có thể di chuyển chính xác được.

Chương trình minh họa trong bài hướng dẫn này chủ yếu để bạn đọc có thể nắm được kiến trúc của toàn hệ thống. Thực ra, việc xử lý trí tuệ nhân tạo sẽ được xử lý trên máy tính và gửi dữ liệu Bluetooth tới mạch Microbit. Dựa trên kiến trúc này, không phải là một Robot, mà là một thiết bị thông minh, chẳng hạn như đèn hay quạt, có thể điều khiển được thông qua giọng nói.



CHƯƠNG 6

Tích hợp Micro để nhận dạng âm thanh trên Python



Pandas



1 Giới thiệu

Trong bài hướng dẫn này, chúng ta sẽ sử dụng kết quả huấn luyện ở bài trước và hiện thực một chương trình nhận dạng âm thanh bằng ngôn ngữ Python. Tương tự như bài toán ở chương 3, trong bài này, chúng ta sẽ lấy audio động từ Micro có sẵn trên máy tính. Ý tưởng ở đây là định kì ghi âm 2s rồi mới nhận dạng. Quá trình này sẽ diễn ra liên tục cho đến khi hệ thống bị tắt hoặc dừng lại. Khi nhận dạng thành công một đối tượng đã được huấn luyện, kết quả sẽ được xuất ra màn hình.

Tuy nhiên việc nhận dạng âm thanh là tương đối phức tạp so với việc nhận dạng hình ảnh, cụ thể ở các điểm như sau:

1. Khác với hệ thống nhận diện hình ảnh, Teachable Machine không cung cấp sẵn các đoạn lệnh phục vụ cho việc nhận dạng âm thanh trên python. Do đó, chúng ta cần phải hiểu được mô hình và tự đưa dữ liệu vào.
2. Đối với hệ thống nhận diện hình ảnh, chúng ta đơn giản chỉ cần đưa một bức hình vào mô hình là nó có thể nhận dạng được ngay. Còn đối với âm thanh, do một số đặc thù về tính chất nên chúng ta cần phải thực hiện thêm nhiều thao tác tiền xử lý dữ liệu. Mục đích để dữ liệu đầu vào được tốt hơn, từ đó mô hình có thể dự đoán chính xác hơn.
3. Ngoài ra, mô hình kết quả của hệ thống nhận diện âm thanh trên Teachble machine chỉ hỗ trợ cho ngôn ngữ web (.json) mà không hỗ trợ cho ngôn ngữ Python. Nên chúng ta cần phải thêm một bước chuyển đổi mô hình về đúng dạng file ".h5" để các thư viện trên Python có thể hiểu và chạy được.

Các kĩ năng chính của bài hướng dẫn này được tóm tắt như sau:

- Thêm thư viện - Cài đặt thư viện cho dự án.
- Đóng gói chương trình thành hàm.
- Tích hợp ghi âm từ Micro.

2 Tích hợp Micro

Bước 1: Thêm thư viện sounddevice và thư viện write từ scipy.io.wavfile và thực hiện hàm recordAudio, như gợi ý sau đây:

```
1 import sounddevice
2 from scipy.io.wavfile import write
3
4 def recordAudio():
5     fs = 44100
6     seconds = 2
7     print("Bat dau")
8     myrecording = sounddevice.rec(int(seconds * fs),
9     samplerate=fs, channels=2)
10    sounddevice.wait()
```

```

10 print("Ket thuc")
11 write('audio.wav', fs, myrecording)

```

Chương trình 6.1: "Thêm các thư viện cần thiết để tiến hành thu âm từ Micro"

Nếu bạn chưa có thư viện `sounddevice` và `scipy.io.wavfile` thì bạn có thể cài nó bằng cách rê chuột vào tên thư viện trong câu lệnh `import`, gợi ý cài đặt sẽ được PyCharm hiện ra.

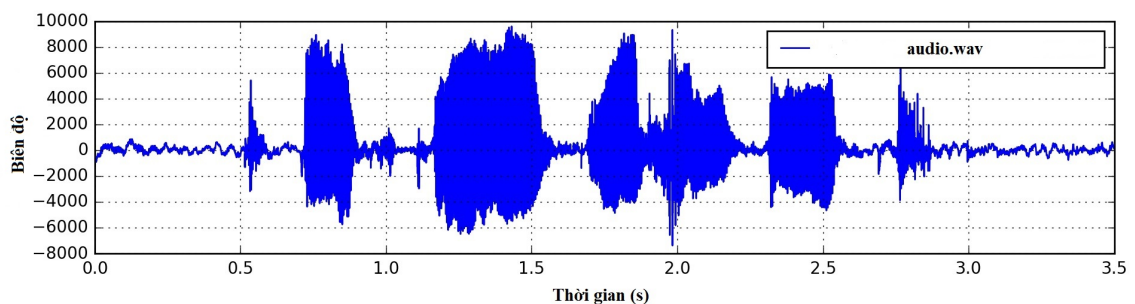
Tiếp theo, hàm `recordAudio()` sẽ được thực thi như sau:

- Đầu tiên chúng ta phải cung cấp thông tin về tần số lấy mẫu (số lần lấy mẫu trên một giây, có đơn vị là Hz) thông qua biến `fs`. Và thời gian ghi âm (giây) thông qua biến `seconds`.
- Trước khi bắt đầu ghi âm, chương trình sẽ xuất hiện dòng chữ "**Bat dau**" để bạn biết được khi nào bắt đầu ghi âm. Và chờ đến khi việc ghi âm hoàn tất thì thông báo cho người dùng biết bằng cách hiển thị dòng "**Ket thuc**" ra màn hình.
- Cuối cùng là lưu đoạn ghi âm đó lại dưới tên **audio.wav**.

Bước 2: Tiền xử lý dữ liệu:

Không đơn giản như dữ liệu cho hệ thống nhận diện hình ảnh, đoạn audio thu được ở Bước 1 được biểu diễn dưới dạng biên độ dao động theo miền thời gian (như hình 6.1). Đối với dạng dữ liệu này thì mô hình không thể đọc và phân loại được. Cho nên, chúng ta cần áp dụng biến đổi Fourier để chuyển đổi dữ liệu về miền tần số trước khi đưa dữ liệu vào mô hình để nhận diện.

Ngoài ra, do đặc tính của âm thanh mà dữ liệu rất dễ bị nhiễu. Điều đó làm cho mô hình nhận diện âm thanh khó có thể nhận diện một cách chính xác được. Vậy nên, trong quá trình chuyển đổi dữ liệu về miền tần số, chúng ta phải áp dụng một số kỹ thuật tiền xử lý tín hiệu âm thanh để lọc nhiễu và làm cho dữ liệu thu được tốt hơn.



Hình 6.1: Dữ liệu âm thanh được biểu diễn dựa trên biên độ theo thời gian

Sau đây là đoạn chương trình dùng để tiền xử lý dữ liệu như lọc nhiễu âm thanh, tách phần âm im lặng ra khỏi đoạn audio và chuyển đổi dữ liệu về miền tần số.

```

1 def read_mfcc(input_filename):
2     sample_rate, audio = wf.read(input_filename)
3     audio = (audio[:, 0] + audio[:, 1]) / 2 # Convert to
Mono
4     energy = np.abs(audio)
5     silence_threshold = np.percentile(energy, 95)
6     offsets = np.where(energy > silence_threshold)[0]
7
8     vad = VoiceActivityDetection()
9     vad.process(audio)
10    voice_samples = vad.get_voice_samples()
11    audio_voice_only = voice_samples
12
13    mfcc = mfcc_fbank(audio_voice_only, sample_rate)
14    return mfcc
15
16
17 # Apply to remove silence
18 class VoiceActivityDetection:
19
20     def __init__(self):
21         self.__step = 160
22         self.__buffer_size = 160
23         self.__buffer = np.array([], dtype=np.int16)
24         self.__out_buffer = np.array([], dtype=np.int16)
25         self.__n = 0
26         self.__VADthd = 0.
27         self.__VADn = 0.
28         self.__silence_counter = 0
29
30     def vad(self, _frame):
31         frame = np.array(_frame) ** 2.
32         result = True
33         threshold = 0.1
34         thd = np.min(frame) + np.ptp(frame) * threshold
35         self.__VADthd = (self.__VADn * self.__VADthd + thd)
36         / float(self.__VADn + 1.)
37         self.__VADn += 1.
38
39         if np.mean(frame) <= self.__VADthd:
40             self.__silence_counter += 1
41         else:
42             self.__silence_counter = 0
43
44         if self.__silence_counter > 20:
45             result = False
46         return result
47
48     # Push new audio samples into the buffer.

```

```

48     def add_samples(self, data):
49         self.__buffer = np.append(self.__buffer, data)
50         result = len(self.__buffer) >= self.__buffer_size
51         return result
52
53     # Pull a portion of the buffer to process
54     # (pulled samples are deleted after being
55     # processed
56     def get_frame(self):
57         window = self.__buffer[:self.__buffer_size]
58         self.__buffer = self.__buffer[self.__step:]
59         return window
60
61     # Adds new audio samples to the internal
62     # buffer and process them
63     def process(self, data):
64         if self.add_samples(data):
65             while len(self.__buffer) >= self.__buffer_size:
66                 # Framing
67                 window = self.get_frame()
68                 # print('window size %i'%window.size)
69                 if self.vad(window): # speech frame
70                     self.__out_buffer = np.append(self.
__out_buffer, window)
71                 # print('__out_buffer size %i'%self.
__out_buffer.size)
72
73     def get_voice_samples(self):
74         return self.__out_buffer
75
76
77 def pad_mfcc(mfcc, max_length): # num_frames, nfilt=64.
78     if len(mfcc) < max_length:
79         mfcc = np.vstack((mfcc, np.tile(np.zeros(mfcc.shape
[1]), (max_length - len(mfcc), 1))))
80     return mfcc
81
82
83 def mfcc_fbanks(signal: np.array, sample_rate: int): # 1D
signal array.
84     # Returns MFCC with shape (num_frames, n_filters, 3).
85     filter_banks, energies = fbanks(signal, samplerate=
sample_rate, nfilt=NUM_FBANKS)
86     frames_features = normalize_frames(filter_banks)
87     return np.array(frames_features, dtype=np.float32) #
Float32 precision is enough here.
88
89
90 def normalize_frames(m, epsilon=1e-12):

```

```
91     return [(v - np.mean(v)) / max(np.std(v), epsilon) for
           v in m]
```

Chương trình 6.2: Đoạn chương trình hỗ trợ việc đọc và tiền xử lý dữ liệu

Bởi vì đoạn chương trình trên dùng khá nhiều kỹ thuật phức tạp liên quan đến xử lý tín hiệu, các phép toán học máy để tiền xử lý dữ liệu. Để có thể hiểu được hết các kỹ thuật này, đòi hỏi phải nắm vững kiến thức về học máy, mô hình học máy cũng như sử dụng thuần thục các thư viện tính toán trong Python.

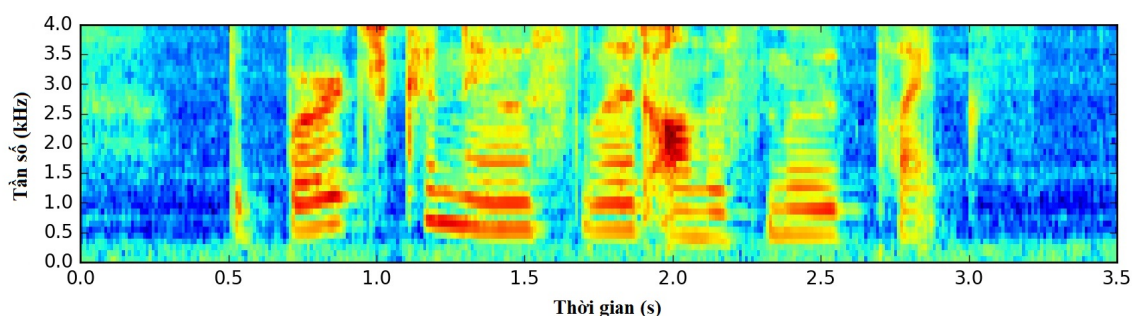
Do đó, trong phạm vi của bài viết này, chúng ta sẽ không đi sâu vào tìm hiểu chi tiết các kỹ thuật đó được hiện thực như thế nào. Mà chỉ cần nắm được cách gọi hàm để tiền xử lý dữ liệu và chuyển đổi dữ liệu về đúng định dạng để mô hình nhận diện âm thanh có thể nhận diện được.

Bạn chỉ cần gọi hàm **read_mfcc()** với tham số truyền vào chính là đường dẫn đến file audio mà chúng ta cần xử lý dữ liệu. Gán kết quả trả về vào một biến cụ thể để bước tiếp theo có thể dễ dàng truyền dữ liệu vào mô hình.

```
1 audio = read_mfcc("audio.wav")
```

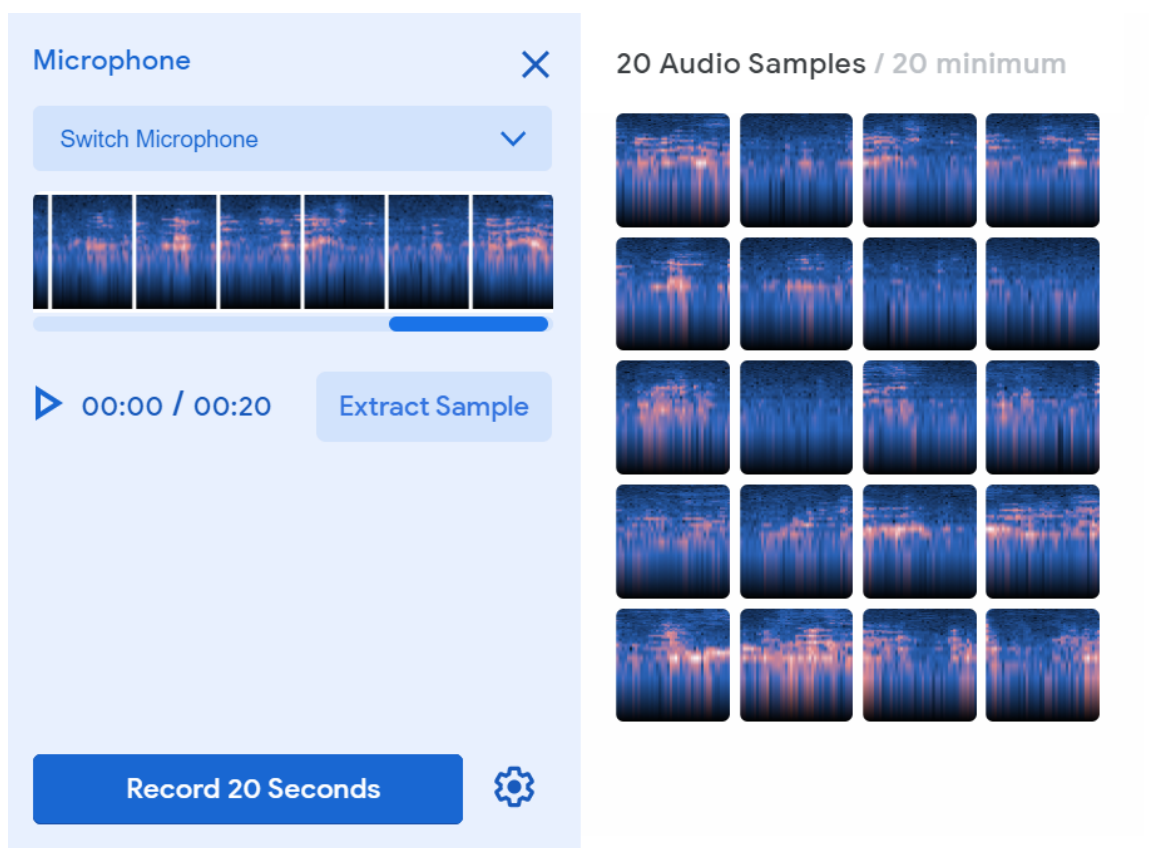
Chương trình 6.3: Câu lệnh tiền xử lý dữ liệu

Và dữ liệu nhận được sẽ như sau:



Hình 6.2: Kết quả dữ liệu âm thanh sau khi đã qua bước tiền xử lý

Còn đây là dữ liệu khi chúng ta đưa vào để xây dựng hệ thống nhận diện âm thanh trên Teachable Machine. Có thể thấy, hình ảnh về dữ liệu sau khi thực hiện lệnh **read_mfcc()** và hình ảnh trên Teachable Machine khá tương đồng với nhau. Tất cả đều được biểu diễn dưới dạng tần số.



Hình 6.3: Dữ liệu khi huấn luyện mô hình nhận dạng âm thanh

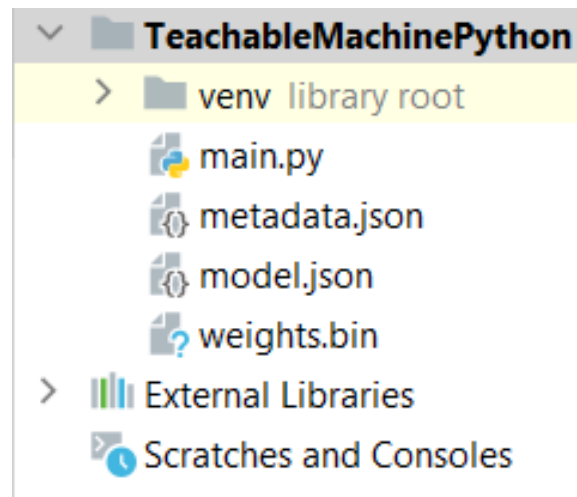
Bước 3: Load mô hình nhận dạng âm thanh từ Teachable Machine

Hiện tại, TensorFlow chính là thư viện mã nguồn mở cho các dự án về trí tuệ nhân tạo nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Thư viện này giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều.

Bên cạnh đó, ở chương trình mẫu từ trang chủ Học máy với Google cho hệ thống hệ thống nhận diện hình ảnh ở chương 3 cũng sử dụng thư viện TensorFlow này. Và mô hình được dùng để tích hợp với Tensorflow trong mô hình có kiểu **.h5**, cụ thể là `keras_model.h5`.

Tuy nhiên, mô hình kết quả của hệ thống nhận diện âm thanh từ Teachable Machine lại có kiểu là **.json**, một kiểu dữ liệu hỗ trợ lập trình trên nền tảng web. Do vậy, để có thể sử dụng được mô hình nhận diện âm thanh này, chúng ta cần phải cài đặt và thêm một thư viện có tên là **tensorflowjs** vào chương trình. Cách cài đặt được thực hiện tương tự như các thao tác cài đặt thư viện ở các bước trên.

Sau đó, chúng ta chép cả 3 file: **metadata.json**, **model.json** và **weights.bin** đã tải ở bài trước vào cùng thư mục với file `main.py`. Kết quả của bước này, chúng ta sẽ có thêm các thư viện cùng cấp với file `main.py`, như minh họa ở hình 6.4:



Hình 6.4: Chép các file của mô hình nhận diện âm thanh vào cùng thư mục

Sau đây là đoạn lệnh mẫu để tải mô hình có kiểu **.json** vào chương trình:

```
1 import tensorflowjs
2 model = tensorflowjs.converters.load_keras_model("model.
  json")
```

Chương trình 6.4: Tải mô hình nhận diện âm thanh vào chương trình

Bước 4: Đưa dữ liệu về đúng kích thước đầu vào của mô hình

Sau khi đã chuyển đổi dữ liệu về dạng tần số ở bước 2. Để mô hình có thể nhận diện được, chúng ta còn cần phải đưa dữ liệu về đúng kích thước dữ liệu đầu vào của mô hình.

Để tìm được đúng kích thước của dữ liệu đầu vào, chúng ta phải xem tóm lược các thông số của mô hình nhằm tìm ra được các dữ liệu cần thiết phục vụ cho bài toán. Cụ thể ta sẽ tải mô hình lên theo cách ở bước 3 và dùng hàm `summary()` kết hợp với hàm `print()` để hiển thị các thông số của mô hình:

```
1 import tensorflowjs
2 model = tensorflowjs.converters.load_keras_model("model.
  json")
3 print(model.summary())
```

Chương trình 6.5: Các thông số của

Trong phần kết quả hiển thị ra màn hình, chúng ta chỉ cần quan tâm đến giá trị "Output Shape" ở dòng "InputLayer" được khoanh đỏ như trong hình 6.6. **None** có nghĩa là giá trị nào cũng được nhưng thường khi chuyển đổi, ta hay để giá trị là 1.

Layer (type)	Output Shape	Param #
conv2d_1_input (InputLayer)	[(None, 43, 232, 1)]	0
conv2d_1 (Conv2D)	(None, 42, 225, 8)	136

Hình 6.5: Một phần kết quả hiển thị sau khi thực hiện câu lệnh summary()

Giả sử chúng ta đã có đoạn code tiền xử lý dữ liệu ở bước 2, và chỉ cần gọi hàm read_mfcc() để thực hiện việc tiền xử lý dữ liệu. Sau đó, để chuyển đổi kích thước của file audio và lưu kết quả nhận được vào biến batch. Chúng ta hiện thực như sau:

```

1 import numpy as np
2
3 audio = read_mfcc("audio.wav")
4 print(audio.shape)
5
6 input_data = np.expand_dims(audio[:LENGTH], -1)
7 batch = np.empty((1, 43, 232, 1), dtype=np.float32)
8 batch[0] = input_data
9 print(batch.shape);

```

Chương trình 6.6: Các thông số của

Trong đoạn code trên, chúng ta dùng thêm hàm print() kết hợp với **shape** để xem kích thước trước và sau khi chuyển đổi của file audio. Kết quả được hiển thị như sau:

(197, 232)
(1, 43, 232, 1)

Hình 6.6: Kích thước của file audio trước và sau khi chuyển đổi

Dòng đầu tiên là kích thước của file audio trước khi chuyển đổi. Dòng thứ hai là kích thước của file audio sau khi chuyển đổi. So sánh với kích thước đầu vào mà mô hình yêu cầu ở hình 6.6, chúng ta có thể thấy 2 kích thước đã khớp với nhau. Tối đây chúng ta đã có thể đưa dữ liệu vào mô hình để mô hình nhận ra kết quả.

Bước 5: Hiện thực thêm chương trình thực thi, gồm 2 bước là ghi âm và nhận dạng, như sau:

```

1 import sounddevice
2 import tensorflowjs
3 from scipy.io.wavfile import write
4 import scipy.io.wavfile as wf
5 import numpy as np

```

```

6 from python_speech_features import fbank
7
8 LENGTH = 43
9 NUM_FBANKS = 232
10 EXT = 'wav'
11
12 def recordAudio():
13     fs = 44100 # Sample rate
14     seconds = 2 # Duration of recording
15     print("start")
16     myrecording = sounddevice.rec(int(seconds * fs),
17     samplerate=fs, channels=2)
18     sounddevice.wait() # Wait until recording is finished
19     print("finish")
20     write('audio.wav', fs, myrecording) # Save as WAV file
21
22 def read_mfcc(input_filename):
23     sample_rate, audio = wf.read(input_filename)
24     audio = (audio[:, 0] + audio[:, 1]) / 2 # Convert to
25     Mono
26     energy = np.abs(audio)
27     silence_threshold = np.percentile(energy, 95)
28     offsets = np.where(energy > silence_threshold)[0]
29
30     vad = VoiceActivityDetection()
31     vad.process(audio)
32     voice_samples = vad.get_voice_samples()
33     audio_voice_only = voice_samples
34
35     mfcc = mfcc_fbank(audio_voice_only, sample_rate)
36     return mfcc
37
38 # Apply to remove silence
39 class VoiceActivityDetection:
40
41     def __init__(self):
42         self.__step = 160
43         self.__buffer_size = 160
44         self.__buffer = np.array([], dtype=np.int16)
45         self.__out_buffer = np.array([], dtype=np.int16)
46         self.__n = 0
47         self.__VADthd = 0.
48         self.__VADn = 0.
49         self.__silence_counter = 0
50
51     def vad(self, _frame):
52         frame = np.array(_frame) ** 2.
53         result = True

```

```

53         threshold = 0.1
54         thd = np.min(frame) + np.ptp(frame) * threshold
55         self.__VADthd = (self.__VADn * self.__VADthd + thd)
/ float(self.__VADn + 1.)
56         self.__VADn += 1.
57
58         if np.mean(frame) <= self.__VADthd:
59             self.__silence_counter += 1
60         else:
61             self.__silence_counter = 0
62
63         if self.__silence_counter > 20:
64             result = False
65         return result
66
67     # Push new audio samples into the buffer.
68     def add_samples(self, data):
69         self.__buffer = np.append(self.__buffer, data)
70         result = len(self.__buffer) >= self.__buffer_size
71         # print('__buffer size %i'%self.__buffer.size)
72         return result
73
74     # Pull a portion of the buffer to process
75     # (pulled samples are deleted after being
76     # processed
77     def get_frame(self):
78         window = self.__buffer[:self.__buffer_size]
79         # print(self.__buffer)
80         self.__buffer = self.__buffer[self.__step:]
81         # print('__buffer size %i'%self.__buffer.size)
82         return window
83
84     # Adds new audio samples to the internal
85     # buffer and process them
86     def process(self, data):
87         if self.add_samples(data):
88             while len(self.__buffer) >= self.__buffer_size:
89                 # Framing
90                 window = self.get_frame()
91                 # print('window size %i'%window.size)
92                 if self.vad(window): # speech frame
93                     self.__out_buffer = np.append(self.
__out_buffer, window)
94                     # print('__out_buffer size %i'%self.
__out_buffer.size)
95
96     def get_voice_samples(self):
97         return self.__out_buffer
98

```

```

99
100 def pad_mfcc(mfcc, max_length): # num_frames, nfilt=64.
101     if len(mfcc) < max_length:
102         mfcc = np.vstack((mfcc, np.tile(np.zeros(mfcc.shape
103             [1]), (max_length - len(mfcc), 1))))
104     return mfcc
105
106 def mfcc_fbank(signal: np.array, sample_rate: int): # 1D
107     signal array.
108     # Returns MFCC with shape (num_frames, n_filters, 3).
109     filter_banks, energies = fbank(signal, samplerate=
110     sample_rate, nfilt=NUM_FBANKS)
111     frames_features = normalize_frames(filter_banks)
112     # delta_1 = delta(filter_banks, N=1)
113     # delta_2 = delta(delta_1, N=1)
114     # frames_features = np.transpose(np.stack([filter_banks
115     , delta_1, delta_2]), (1, 2, 0))
116     return np.array(frames_features, dtype=np.float32) #
117     Float32 precision is enough here.
118
119 def normalize_frames(m, epsilon=1e-12):
120     return [(v - np.mean(v)) / max(np.std(v), epsilon) for
121     v in m]
122
123 def audio_detection():
124     audio = read_mfcc("audio.wav")
125     model = tensorflowjs.converters.load_keras_model("model
126     .json")
127     input_data = np.expand_dims(audio[:LENGTH], -1)
128     batch = np.empty((1, 43, 232, 1), dtype=np.float32)
129     batch[0] = input_data
130
131     prediction = model.predict(batch)
132     name = ["Tap am", "Tam Bang", "Nhan le"]
133     index = -1
134     max_value = -1
135     for i in range(0, len(prediction[0])):
136         if max_value < prediction[0][i]:
137             max_value = prediction[0][i]
138             index = i
139
140     print("Ket Qua: ", name[index])
141     print("Chinh Xac: ", max_value)
142
143 while (True):
144     recordAudio()

```

Chương trình 6.7: Hoàn thiện chương trình

3 Câu hỏi ôn tập

1. Việc nhận dạng âm thanh từ Micro được xây dựng dựa trên ý tưởng nào?
 - A. Nhận dạng trực tiếp âm thanh từ micro.
 - B. Ghi âm từng đoạn âm thanh với thời gian cho trước rồi nhận dạng.
 - C. Ghi âm một đoạn âm thanh bất kỳ với thời gian bất kỳ.
 - D. Tất cả đều sai
2. Thư viện cho việc thu âm từ Micro và ghi dữ liệu xuống máy tính là gì?
 - A. webcam2
 - B. cv2
 - C. sounddevice
 - D. sounddevice và thư viện write từ scipy.io.wavfile
3. Thư viện cho việc tải mô hình nhận dạng âm thanh từ Teachable Machine là gì?
 - A. tensorflow
 - B. keras
 - C. tensorflowjs
 - D. numpy
4. Tại sao chúng ta cần tiền xử lý dữ liệu âm thanh trước khi đưa vào mô hình?
 - A. Dữ liệu âm thanh thường bị nhiễu.
 - B. Dữ liệu âm thanh thu được có chất lượng kém
 - C. Dữ liệu âm thanh không phù hợp với mô hình nhận dạng âm thanh.
 - D. Tất cả đều đúng.
5. Nguyên nhân chính ảnh hưởng đến tốc độ thực thi của chương trình là gì?
 - A. Máy tính cấu hình thấp
 - B. Micro thu âm dữ liệu không tốt
 - C. Môi trường ảo PyCharm có hiệu suất thấp
 - D. Tất cả các nguyên nhân trên
6. Để xem kích thước dữ liệu đầu vào của một mô hình chúng ta dùng lệnh print() kết hợp với lệnh nào?
 - A. summary()
 - B. predict()
 - C. sum()
 - D. range()
7. Để mô hình có thể tính toán và nhận dạng được dữ liệu đầu vào, chúng ta dùng lệnh nào?
 - A. summary()
 - B. predict()
 - C. sum()
 - D. range()

Đáp án

1. B 2. B 3. C 4. D 5. C 6. A 7. B

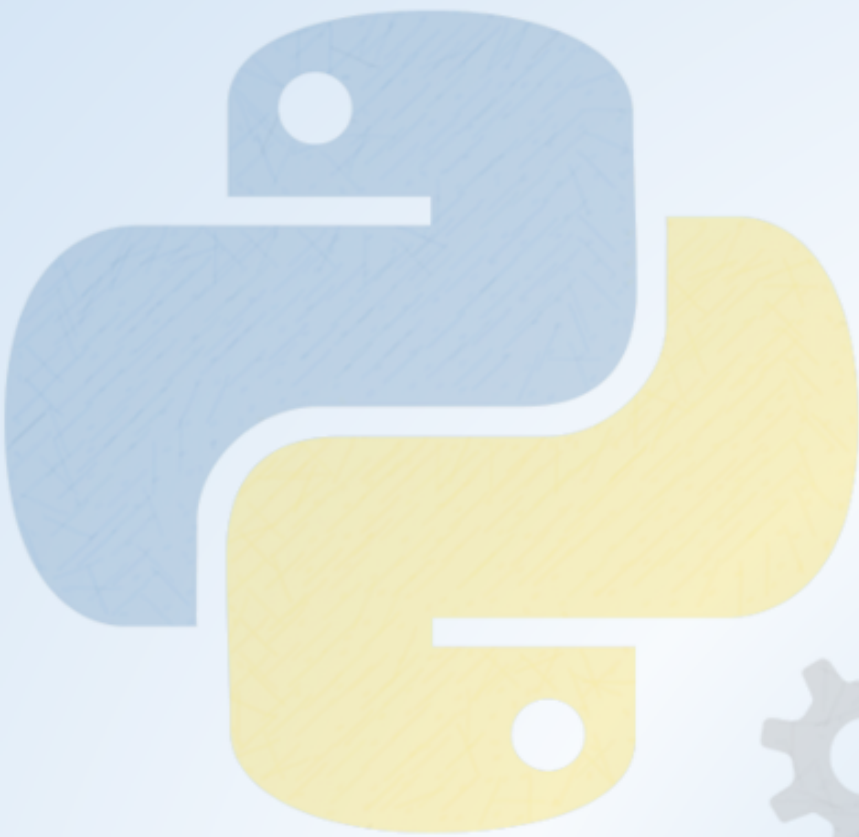
Phần III

Khoa học dữ liệu



CHƯƠNG 7

Khoa học Dữ liệu cùng Python



Pandas



1 Giới thiệu

Thế giới đang trải qua cuộc cách mạng công nghiệp 4.0 với đặc trưng là Dữ liệu lớn (Big data), Trí tuệ nhân tạo (AI), Internet vạn vật (IoT), Điện toán đám mây (Cloud computing) và đặc biệt là Khoa học dữ liệu (Data Science). Hiểu một cách đơn giản, Khoa học dữ liệu là khoa học về việc quản trị và phân tích dữ liệu, trích xuất các giá trị từ dữ liệu để tìm ra các hiểu biết, các tri thức hành động, các quyết định dẫn dắt hành động. Mục đích chính của khoa học dữ liệu là biến đổi một lượng lớn dữ liệu chưa qua xử lý, làm thế nào để định vị được thành mô hình kinh doanh, từ đó giúp đỡ các tổ chức tiết giảm chi phí, gia tăng hiệu quả làm việc, nhìn nhận cơ hội, rủi ro trên thị trường và làm gia tăng lợi thế cạnh tranh của doanh nghiệp.

Khoa học dữ liệu gồm có ba phần chính: Tạo ra và quản trị dữ liệu, phân tích dữ liệu và chuyển kết quả phân tích thành giá trị của hành động. Việc phân tích và dùng dữ liệu lại dựa vào ba nguồn tri thức: toán học (thống kê toán học), công nghệ thông tin (máy học) và tri thức của lĩnh vực ứng dụng cụ thể.

Trong lĩnh vực công nghệ thông tin, mặc dù có rất nhiều ngôn ngữ lập trình cao cấp, Python vẫn là ngôn ngữ lập trình tiêu biểu đối với các chuyên gia làm việc trong lĩnh vực khoa học dữ liệu (Data Science). Nhu cầu ngày càng tăng đối với các Nhà khoa học dữ liệu có tay nghề cao trong ngành công Nghệ Thông Tin và Python đã phát triển như ngôn ngữ lập trình ưa thích nhất. Với sự trợ giúp của hướng dẫn này về Python cho khoa học dữ liệu, bạn sẽ hiểu tại sao Python được coi là ngôn ngữ ưa thích nhất.

Tại sao sử dụng Python cho Khoa học Dữ liệu? Vì sự sẵn có của nhiều thư viện Khoa học dữ liệu / Phân tích dữ liệu như Pandas, StatsModels, NumPy, SciPy và Scikit-Learn, là một số thư viện nổi tiếng có sẵn cho những người tham gia trong cộng đồng Khoa học dữ liệu.

2 Thư viện thường được sử dụng

2.1 Numpy

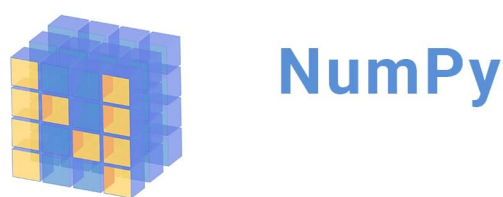
NumPy là một thư viện Python cơ bản dành cho khoa học điện toán. Nó đi kèm với sự hỗ trợ cho một đối tượng mảng N-Chiều mạnh mẽ và các chức năng broadcasting.

Mục đích của thư viện NumPy:

- Mục đích chính: Dành cho Học máy.
- Mục đích phụ: Biểu thị hình ảnh, sóng âm thanh dưới dạng một mảng các số thực trong mảng N chiều, binary raw stream.

2.2 Pandas

Pandas cung cấp các cấu trúc dữ liệu có ý nghĩa, nhanh chóng và linh hoạt. Pandas nhằm mục đích thực hiện phân tích dữ liệu trong thế giới thực bằng Python. Một



Hình 7.1: Thư viện Numpy trong Python.

trong những tính năng mạnh mẽ nhất của Pandas là đơn giản hóa các thao tác dữ liệu phức tạp chỉ bằng một hoặc hai lệnh. Ngoài ra, thư viện này có rất nhiều các phương thức tích hợp dành cho kết hợp, lọc và nhóm dữ liệu. Nó cũng có tính năng time-series.



Hình 7.2: Thư viện Pandas trong Python.

Mục đích chính: Phân tích dữ liệu trong thế giới thực thành dữ liệu có ích.

Điểm nổi bật của thư viện Pandas:

- Khả năng thực hiện các loại thao tác tùy chỉnh.
- Đảm bảo rằng toàn bộ quá trình thao tác dữ liệu dễ dàng hơn.
- Cung cấp chức năng và tính linh hoạt cao khi được sử dụng với các công cụ và thư viện Python khác.
- Có thể chọn đầu ra phù hợp nhất cho phương thức áp dụng.
- Hỗ trợ các phép gộp, ghép, lặp, lặp chỉ mục lại và các thao tác trực quan hóa.

2.3 Matplotlib

Matplotlib là một thư viện mô phỏng hai chiều cho ngôn ngữ lập trình Python. Nó có khả năng tạo ra các số liệu có giá trị cao và môi trường đa nền tảng tương tác. Theo trang web chính thức của Matplotlib mô tả thư viện Python này cố gắng

"Làm những thứ dễ dàng thành dễ làm và những thứ khó khăn thành những thứ có thể". Thư viện này cho phép tạo biểu đồ thanh, biểu đồ lỗi, biểu đồ phân tán... với ít dòng code hơn.



Hình 7.3: Thư viện Matplotlib trong Python.

Mục đích của thư viện Matplotlib:

- Mục đích chính: Vẽ đồ thị 2D.
- Mục đích phụ: Trực quan hóa dữ liệu.

Điểm nổi bật của Matplotlib:

- Kiểm soát hoàn toàn các thuộc tính trục, thuộc tính phông chữ, kiểu đường kẻ, v.v. thông qua interface hướng đối tượng.
- Thư viện huyền thoại cho sơ đồ phân tán.
- Hỗ trợ trục x / y thứ cấp.
- Hoạt động rất tốt với một số hệ thống đồ họa và hệ điều hành.

2.4 SciPy

SciPy là thư viện Python đi kèm với một lượng module cho đại số tuyến tính, và số liệu thống kê. Các nguồn mở thư viện Python cho phép các lập trình viên và kỹ sư làm việc với biến đổi Fourier, ODE solvers, xử lý tín hiệu và xử lý ảnh...

NumPy arrays được dùng như một cấu trúc dữ liệu cơ bản của SciPy. Tất cả các chức năng cung cấp bởi các module con của SciPy cũng có tài liệu rất tốt. Do đó, rất dễ dàng để bắt đầu với thư viện máy học này.

Mục đích của thư viện SciPy:

- Mục đích chính: Học máy, nghiên cứu khoa học.
- Mục đích phụ: Giải Quyết toán học chức năng.

Điểm nổi bật của thư viện SciPy:

- Dễ xử lý các hoạt động toán học.
- Cung cấp giải pháp hiệu quả cho tích phân, nội suy, tối ưu hóa...
- Hỗ trợ xử lý tín hiệu.



Hình 7.4: Thư viện SciPy trong Python.

2.5 Scikit-Learn

Được coi là một trong những thư viện Python tốt nhất để làm việc với dữ liệu phức tạp, Scikit-Learn được xây dựng dựa trên các thư viện Matplotlib, NumPy và SciPy. Thư viện học máy này có những tính năng đơn giản đơn giản nhưng hiệu quả để hoàn thành các nhiệm vụ phân tích và khai phá dữ liệu. Scikit-Learn là một trong những thư viện Python phát triển nhanh nhất. Ngoài ra, Scikit-Learn có một số thuật toán để thực hiện các nhiệm vụ khai thác dữ liệu và học máy, đáng chú ý là phân loại, phân cụm, lựa chọn mô hình, reduce và hồi quy.



Hình 7.5: Thư viện Scikit-Learn trong Python.

Mục đích chính của Scikit-Learn: Dành cho Phân tích và Khai phá dữ liệu.

Điểm nổi bật của Scikit-Learn:

- Khả năng trích xuất các đặc trưng từ hình ảnh và văn bản.
- Tái sử dụng trong một số hoàn cảnh.
- Một số phương thức để kiểm tra tính chính xác của các mô hình được giám sát trên dữ liệu chưa thấy.
- Một loạt các thuật toán, bao gồm phân cụm, phân tích nhân tố, phân tích thành phần chính cho các mạng thần kinh không giám sát.

3 Thu thập và xử lý dữ liệu

3.1 Thu thập dữ liệu

Thực ra, thu thập dữ liệu là một trong những phần khó nhất và cũng là phần mang lại nhiều giá trị nhất của lĩnh vực khoa học dữ liệu. Điểm khó ở đây, là dữ liệu của bạn sẽ cần phải thu thập từ nhiều nguồn khác nhau được chia sẻ trên mạng, chứ nó thường không nằm tập trung tại một server. Thêm nữa, bạn cần phải trích lọc thông tin cần thiết, và loại bỏ các thông tin thừa để tăng tốc cho quá trình xử lý trong tương lai.

Trong phạm vi hướng dẫn của giáo trình này, chúng tôi cung cấp dữ liệu đã thông qua xử lý cho bạn đọc. Điều này sẽ tạo ra cảm giác đơn giản cho các bạn mới bắt đầu, và có thể tập trung vào chức năng phân tích dữ liệu có sẵn. Dữ liệu sử dụng là kết quả thi đại học năm 2020, có thể được tải ở đường dẫn sau đây:

<https://ubc.sgp1.cdn.digitaloceanspaces.com/PythonData/diemthidhqg2020.csv>

Bạn đọc có thể tự tìm dữ liệu cho riêng mình, ví dụ như kết quả thi học kì ở trường, để áp dụng vào các kĩ thuật phân tích dữ liệu được trình bày trong bài hướng dẫn này.

3.2 Xử lý dữ liệu

3.2.1 Thêm dữ liệu vào chương trình

Trước khi bắt đầu xử lý dữ liệu, ta tiến hành thêm các thư viện cần thiết như Numpy, Pandas như trong đoạn code sau:

```
1 import pandas as pd
2 import numpy as np
3 import warnings
4 warnings.filterwarnings("ignore")
```

Chương trình 7.1: Thêm thư viện

Sau đó, ta sẽ đọc dữ liệu từ tập tin csv. Để kiểm tra nội dung dữ liệu, ta sẽ in ra 5 dòng đầu của dữ liệu.

```
1 data = pd.read_csv('diemthidhqg2020.csv')
2 print(data.head(5))
```

Chương trình 7.2: Đọc dữ liệu từ tập tin CSV

Chúng ta sẽ thấy được kết quả như hình dưới, Python sẽ xuất ra 5 dòng dữ liệu đầu tiên và các cột dữ liệu.

Ngoài ra, ta có thể kiểm tra các thông tin khác của dữ liệu bằng lệnh

```
1 print(data.info())
```

Kết quả sẽ in ra bao gồm:

- Cột đầu tiên: số thứ tự các trường dữ liệu (cột) trong dữ liệu.
- Cột thứ hai: các tên của từng trường dữ liệu (tên của từng cột trong dữ liệu)

```

      Unnamed: 0      sbd      ho      ...      sinh_hoc      lich_su      tieng_anh
0              0      2000001  PHẠM      ...      NaN      5.75      5.2
1              1      2000002  ĐẶNG      ...      NaN      7.00      7.0
2              2      2000003  LÂM      ...      NaN      4.75      6.0
3              3      2000004  LÊ      ...      7.0      NaN      5.6
4              4      2000005  LƯ      ...      5.5      NaN      8.2

```

```
[5 rows x 19 columns]
```

Hình 7.6: Lấy 5 dòng dữ liệu đầu tiên.

- Cột thứ ba: là cột đếm số lượng các phần tử không phải là phần tử rỗng (tức là các phần tử có giá trị xác định)
- Cột cuối cùng: đây là cột kiểu dữ liệu của mỗi trường, đa phần sẽ có các kiểu dữ liệu thường gặp như int64, float64, object ...

```

Data columns (total 19 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0  74453 non-null  int64
1   sbd         74453 non-null  int64
2   ho          74453 non-null  object
3   lot         73977 non-null  object
4   ten         74436 non-null  object
5   dd          74453 non-null  int64
6   mm          74453 non-null  int64
7   yyyy       74453 non-null  int64
8   vat_ly     42855 non-null  float64
9   dia_ly     29604 non-null  float64
10  khtn       41950 non-null  float64
11  gdcđ       25239 non-null  float64
12  ngu_van    72648 non-null  float64
13  hoa_hoc    43144 non-null  float64
14  toan_hoc   74205 non-null  float64
15  khxh       25222 non-null  float64
16  sinh_hoc   42608 non-null  float64
17  lich_su    30320 non-null  float64
18  tieng_anh  65898 non-null  float64
dtypes: float64(11), int64(5), object(3)

```

Hình 7.7: Thông tin của dữ liệu.

3.2.2 Trích xuất, chia nhỏ dữ liệu

Sau khi đã thêm dữ liệu thành công vào chương trình, biến *pd* sẽ chứa toàn bộ dữ liệu. Tuy nhiên, để thuận tiện cho việc xử lý, phân tích, đánh giá dữ liệu thì ta cần phải chia nhỏ dữ liệu ra nhiều phần khác nhau, tùy theo mục đích của việc đánh giá.

Trong bài này, ta sẽ đánh giá sự phân bố điểm của từng môn thi (ví dụ 3 môn: Toán, Ngữ Văn, Tiếng Anh).

Ta sẽ sử dụng câu lệnh bên và gán cho biến *toan* tất cả các dữ liệu về điểm thi môn Toán. Để lấy chính xác cột dữ liệu, ta sẽ dựa vào các tên trường dữ liệu mà ta tìm thấy được ở cột thứ hai như Hình 7.7.

```
1 toan = data['toan_hoc']
```

Tuy nhiên, sẽ có một vấn đề là do một số em học sinh không thi môn Toán nên trong dữ liệu sẽ bị cập nhật rỗng (NaN). Các dữ liệu này sẽ ảnh hưởng rất nhiều, thậm chí làm sai lệch kết quả phân tích, đánh giá dữ liệu. Vậy nên, ta sẽ giải quyết bằng cách sử dụng hàm:

```
1 toan = data['toan_hoc'].dropna()
```

Sau khi loại bỏ các phần tử rỗng, ta sẽ kiểm tra lại số lượng phần tử hiện tại (74205) có giá trị khác rỗng so với ban đầu là 74453 (bao gồm cả phần tử rỗng).

```
1 toan = data['toan_hoc']
2 print(len(toan)) #74453
3
4 toan = data['toan_hoc'].dropna()
5 print(len(toan)) #74205
```

Chương trình 7.3: Kiểm tra số lượng phần tử

Tương tự như trên, ta sẽ tạo ra 3 biến chứa dữ liệu của 3 môn Toán, Ngữ Văn, Tiếng Anh:

```
1 toan = data['toan_hoc'].dropna()
2 print(len(toan)) #74205
3
4 van = data['ngu_van'].dropna()
5 print(len(van)) #72648
6
7 anh = data['tieng_anh'].dropna()
8 print(len(anh)) #65898
```

Chương trình 7.4: Tạo các biến chứa dữ liệu

4 Câu hỏi ôn tập

- Để hiển thị 10 dòng đầu của dữ liệu *data*, ta dùng câu lệnh nào?
A. `print(data(10))`
B. `print(data.head())`
C. `print(data.head(10))`
D. `data.head(10)`

- Câu lệnh sau đây có chức năng gì?

```
1 print(data.info())
```

- Hiển thị các thông tin thuộc tính (tên, số lượng phần tử, kiểu dữ liệu) của dữ liệu
 - Hiển thị dữ liệu
 - Hiển thị dòng chữ "data.info()"
 - Hiển thị 5 dòng đầu của dữ liệu
- Để đọc nội dung từ file csv, thư viện nào sẽ được sử dụng?
A. pandas
B. numpy
C. warnings
D. Tất cả các thư viện trên
 - Để xóa các phần tử rỗng, ta dùng câu lệnh nào?
A. `dropna()`
B. `print()`
C. `head()`
D. `info()`
 - Để kiểm tra số lượng phần tử của biến **toan**, chúng ta sẽ dùng câu lệnh gì?
A. `print(toan)`
B. `lenght(toan)`
C. `len(toan)`
D. Tất cả đều sai
 - Kiểu dữ liệu **int64** có nghĩa là gì?
A. Kiểu số nguyên, giá trị lớn nhất là 64
B. Kiểu số nguyên, có 64 bit dữ liệu
C. Kiểu số nguyên, có 64 byte dữ liệu
D. Tùy vào ngữ cảnh, các câu trả lời trên đều đúng
 - Kiểu dữ liệu **float64** có nghĩa là gì?
A. Kiểu số thực, giá trị lớn nhất là 64
B. Kiểu số thực, có 64 bit dữ liệu
C. Kiểu số thực, có 64 byte dữ liệu
D. Tùy vào ngữ cảnh, các câu trả lời trên đều đúng

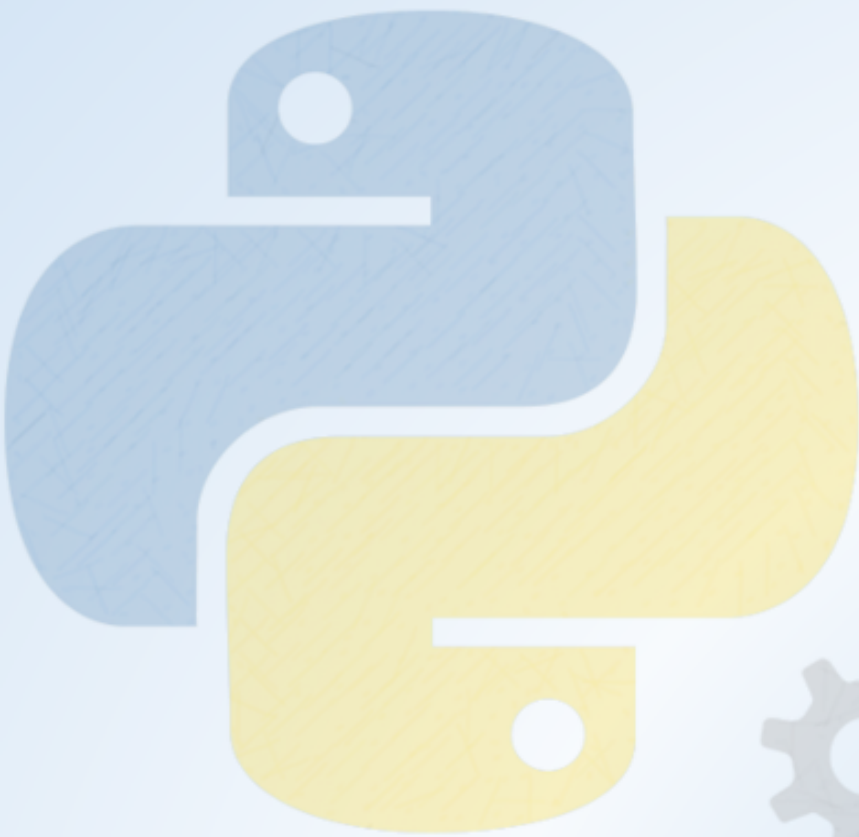
Đáp án

1. C 2. A 3. A 4. A 5. C 6. B 7. B



CHƯƠNG 8

Vẽ đồ thị biểu diễn dữ liệu trực quan



Pandas



1 Giới thiệu

Trong chương vừa rồi, ta đã tìm hiểu sơ lược về khoa học dữ liệu và các thư viện phục vụ cho việc xử lý và phân tích dữ liệu. Vấn đề dữ liệu luôn là một vấn đề nhạy cảm và để thu thập dữ liệu cũng là một thử thách khó khăn. Vậy nên, chúng ta sẽ sử dụng những dữ liệu được công khai trên mạng Internet.

Sau khi đã có được dữ liệu và xử lý chúng, để cho mọi người hình dung được sự trừu tượng và đặc tính của dữ liệu, ta nên biểu thị chúng dưới dạng hình ảnh trực quan (đồ thị). Ta sẽ sử dụng thư viện Matplotlib để vẽ các biểu đồ, và trong chương này ta sẽ hiện thực 3 loại biểu đồ cơ bản:

- Biểu đồ đường
- Biểu đồ cột
- Biểu đồ tròn

Các công cụ giới thiệu trong bài hướng dẫn này rất hữu hiệu để phân tích dữ liệu có kích thước lớn. Nó sẽ cho chúng ta những cái nhìn trực quan hơn và rút trích được nhiều thông tin ý nghĩa hơn. Dữ liệu sử dụng trong bài này là kết quả thi đại học năm 2020, đã được chúng tôi sưu tầm và có thể tải được từ đường dẫn sau đây:

<https://ubc.sgp1.cdn.digitaloceanspaces.com/PythonData/diemthidhqg2020.csv>

2 Vẽ biểu đồ bằng thư viện Matplotlib

Ở bài trước, ta đã trích xuất dữ liệu lớn thành các dữ liệu điểm nhỏ (Toán, Ngữ Văn, Tiếng Anh). Trong bài này, ta sẽ sử dụng các dữ liệu điểm nhỏ đó để tiến hành vẽ đồ thị và phân tích dữ liệu. Trước tiên cần phải thêm thư viện để vẽ biểu đồ:

```
1 import matplotlib.pyplot as plt
```

Chương trình 8.1: Thêm thư viện vẽ đồ thị

2.1 Biểu đồ đường

Ta sẽ dùng dữ liệu điểm thi của môn Toán để vẽ biểu đồ đường.

Trước tiên, ta cần thực hiện việc đếm số lần xuất hiện của dữ liệu điểm trong dữ liệu, tức là với mỗi điểm số sẽ có bao nhiêu thí sinh đạt được. Để thống kê, ta sẽ sử dụng hàm:

```
1 value_counts()
```

7.8	5208
7.6	5144
8.0	5047
8.2	5008
7.4	4829
8.4	4442
7.2	4392
8.6	4038
7.0	3848
8.8	3552
6.8	3362
6.6	2909
9.0	2820
6.4	2433
6.2	2117
9.2	1971
6.0	1847

Hình 8.1: Dữ liệu sau khi được thống kê.

Khi sử dụng hàm thống kê số lần xuất hiện của điểm, các điểm này sẽ bị sắp xếp lộn tung, không có thứ tự, để sắp xếp lại điểm theo thứ tự từ thấp đến cao, ta sẽ sử dụng hàm:

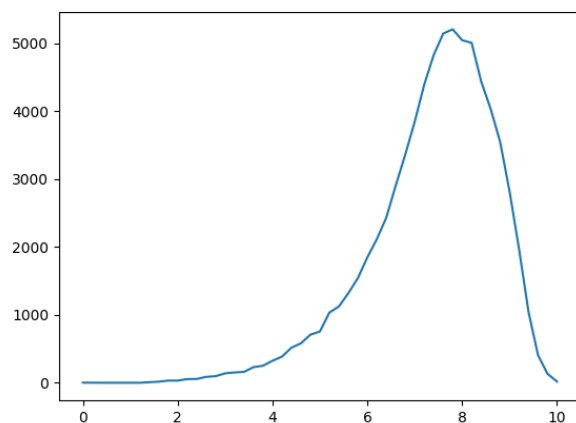
```
1 sort_index()
```

Sau khi đã sắp xếp dữ liệu điểm theo thứ tự, ta tiến hành vẽ biểu đồ đường như sau:

```
1 plt.plot(toan.value_counts().sort_index())
2 plt.show()
```

Chương trình 8.2: Vẽ biểu đồ đường

Kết quả của việc vẽ đồ thị sẽ như hình bên dưới.

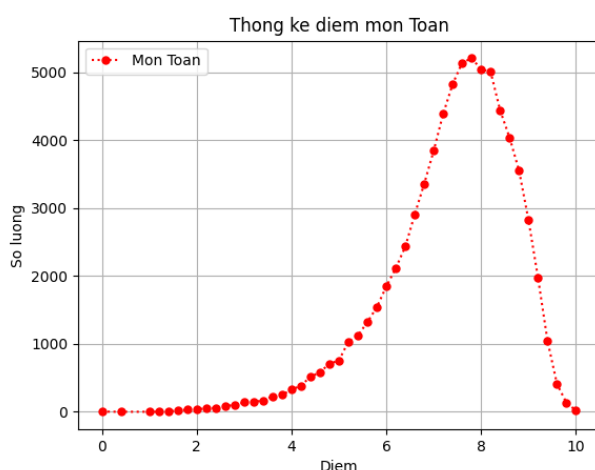


Hình 8.2: Đồ thị biểu diễn điểm thi môn Toán.

Tuy nhiên, để thêm phần sinh động cho đồ thị, ta sẽ thêm các thuộc tính khác như màu, kiểu đường đồ thị, lưới đồ thị, gắn thêm chú thích...

```
1 plt.plot(toan.value_counts().sort_index(), color='red',  
    linestyle=':', marker='.', markersize=10)  
2 plt.grid(True)  
3 plt.legend(['Mon Toan'])  
4 plt.xlabel('Diem')  
5 plt.ylabel('So luong')  
6 plt.title('Thong ke diem mon Toan')  
7 plt.show()
```

Chương trình 8.3: Vẽ biểu đồ có sử dụng thêm thuộc tính



Hình 8.3: Đồ thị hiển thị thêm thông tin, chú thích.

Từ biểu đồ trên, ta có thể đưa ra nhận xét chung rằng khoảng điểm mà nhiều thí sinh đạt được là từ 6 đến 9, và khoảng điểm được nhiều thí sinh đạt được nhất (trên 5000 thí sinh) là từ 7.5 đến 8.1.

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import warnings
5 warnings.filterwarnings("ignore")
6
7 data = pd.read_csv('diemthidhqg2020.csv')
8
9 toan = data['toan_hoc'].dropna()
10
11 plt.plot(toan.value_counts().sort_index(), color='red',
12          linestyle=':', marker='.', markersize=10)
13 plt.grid(True)
14 plt.legend(['Mon Toan'])
15 plt.xlabel('Diem')
16 plt.ylabel('So luong')
17 plt.title('Thong ke diem mon Toan')
18 plt.show()

```

Chương trình 8.4: Chương trình hoàn chỉnh của biểu đồ đường

2.2 Biểu đồ cột

Ta sẽ dùng dữ liệu điểm thi của môn Ngữ Văn để vẽ biểu đồ cột.

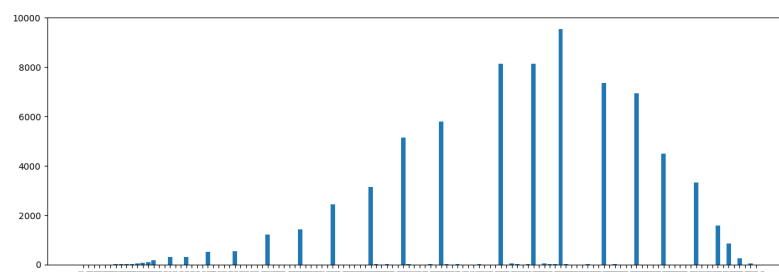
Tương tự các bước như vẽ đồ thị đường, ta sẽ sử dụng hàm `bar()` để vẽ biểu đồ cột, chương trình như sau:

```

1 plt.figure(figsize = (15,5))
2 diem_thi = van.value_counts().sort_index().index.tolist()
3 diem_thi = list(map(str, diem_thi))
4 plt.bar(diem_thi, van.value_counts().sort_index())
5 plt.show()

```

Chương trình 8.5: Vẽ biểu đồ tròn



Hình 8.4: Biểu đồ cột bị phân mảnh.

Tuy nhiên, do dữ liệu điểm khá nhiều điểm số khác nhau, nên khi biểu diễn dưới dạng đồ thị cột sẽ bị phân mảnh, khó để quan sát. Vậy nên ta cần phải gom các khoảng điểm lại để dễ dàng trong việc đánh giá, hiển thị. Trong trường hợp này,

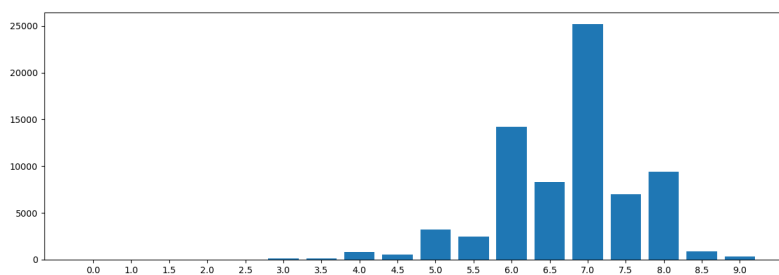
phương pháp làm tròn sẽ phù hợp hơn, mặc dù sẽ gây ra các sai số trong việc đánh giá. Ta sẽ làm tròn 0.5 đơn vị điểm (ví dụ như 6.25 sẽ làm tròn thành 6.5).

```
1 van = round(van*2)/2
```

Ta sẽ có đoạn chương trình như sau:

```
1 van = round(van*2)/2
2 plt.figure(figsize = (15,5))
3 diem_thi = van.value_counts().sort_index().index.tolist()
4 diem_thi = list(map(str, diem_thi))
5 plt.bar(diem_thi, van.value_counts().sort_index())
6 plt.show()
```

Chương trình 8.6: Vẽ biểu đồ cột

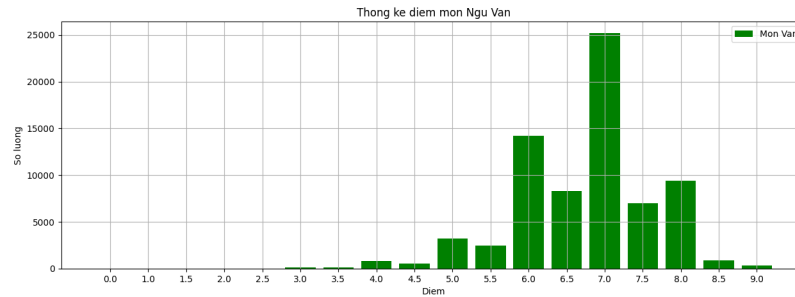


Hình 8.5: Biểu đồ tròn sau khi được làm gọn.

Để thêm phần sinh động, có thêm chú thích và phần trăm, ta thực hiện như sau:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import warnings
5 warnings.filterwarnings("ignore")
6
7 data = pd.read_csv('diemthidhqg2020.csv')
8
9 van = data['ngu_van'].dropna()
10 van = round(van*2)/2
11
12 plt.figure(figsize = (15,5))
13 diem_thi = van.value_counts().sort_index().index.tolist()
14 diem_thi = list(map(str, diem_thi))
15 plt.grid(True)
16 plt.bar(diem_thi, van.value_counts().sort_index(), color='g')
17 plt.legend(['Mon Van'])
18 plt.xlabel('Diem')
19 plt.ylabel('So luong')
20 plt.title('Thong ke diem mon Ng'u V'')
21 plt.show()
```

Chương trình 8.7: Chương trình hoàn chỉnh của biểu đồ tròn cho môn Ngữ Văn



Hình 8.6: Biểu đồ cột biểu diễn điểm môn Ngữ Văn.

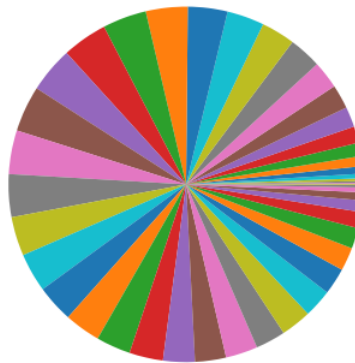
2.3 Biểu đồ tròn

Ta sẽ dùng dữ liệu điểm thi của môn Tiếng Anh để vẽ biểu đồ tròn.

Tương tự các bước như vẽ đồ thị đường, ta sẽ sử dụng hàm `pie()` để vẽ biểu đồ tròn, chương trình như sau:

```
1 anh = data['tieng_anh'].dropna()
2 plt.pie(anh.value_counts().sort_index())
3 plt.show()
```

Chương trình 8.8: Vẽ biểu đồ tròn



Hình 8.7: Biểu đồ tròn bị phân mảnh.

Tuy nhiên, do dữ liệu điểm khá nhiều điểm số khác nhau, nên khi biểu diễn dưới dạng đồ thị tròn sẽ bị phân mảnh, khó để quan sát. Vậy nên ta cần phải gom các khoảng điểm lại để dễ dàng trong việc đánh giá, hiển thị. Trong trường hợp này, phương pháp làm tròn sẽ phù hợp hơn, mặc dù sẽ gây ra các sai số trong việc đánh giá. Ta sẽ làm tròn 1 đơn vị điểm (ví dụ như 6.5 sẽ làm tròn thành 7).

```
1 anh = round(anh*1) / 1
```

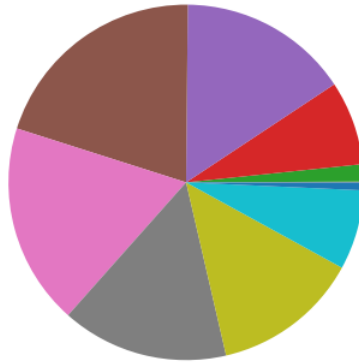
Ta sẽ có đoạn chương trình như sau:

```

1 anh = round(anh*1)/1
2 diem_thi = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10"]
3 plt.pie(anh.value_counts().sort_index(), labels=diem_thi)
4 plt.show()

```

Chương trình 8.9: Vẽ biểu đồ tròn



Hình 8.8: Biểu đồ tròn sau khi được làm gọn.

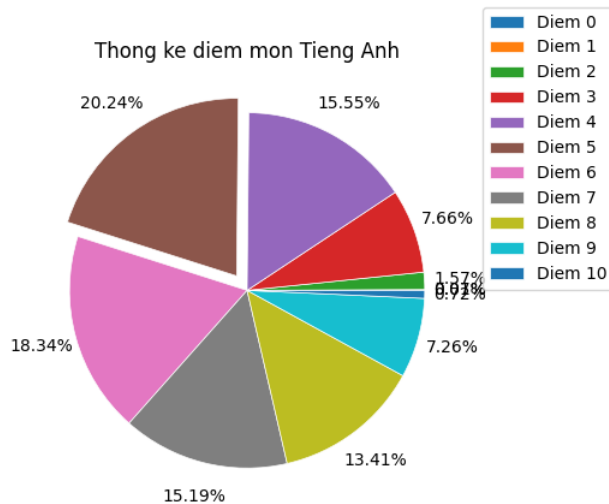
Để thêm phần sinh động, có thêm chú thích và phần trăm, ta thực hiện như sau:

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import warnings
5 warnings.filterwarnings("ignore")
6
7 data = pd.read_csv('diemthidhqg2020.csv')
8
9 anh = data['tieng_anh'].dropna()
10 anh = round(anh*1)/1
11 diem_thi = ["Diem 0", "Diem 1", "Diem 2", "Diem 3", "Diem 4",
12             "Diem 5", "Diem 6", "Diem 7", "Diem 8", "Diem 9", "Diem 10"]
13 explode = [0, 0, 0, 0, 0, 0.1, 0, 0, 0, 0, 0]
14 plt.pie(anh.value_counts().sort_index(), pctdistance=1.2,
15         autopct='%1.2f%%', wedgeprops={'edgecolor': 'white', 'linewidth': 0.5},
16         explode=explode)
17 plt.figlegend(diem_thi)
18 plt.title('Thong ke diem mon Tieng Anh')
19 plt.show()

```

Chương trình 8.10: Chương trình hoàn chỉnh của biểu đồ tròn cho môn Tiếng Anh



Hình 8.9: Biểu đồ tròn biểu diễn điểm môn Tiếng Anh.

3 Biểu diễn kết quả thống kê bằng khuôn hình

Ngoài các dạng biểu đồ trên được cung cấp bởi thư viện Matplotlib, ta có thể vẽ một số biểu đồ mới, sinh động, đa dạng hơn bằng việc sử dụng mask (khuôn). Trong phần này, ta sẽ thống kê dữ liệu những tên thường được đặt nhất.

Ta sẽ sử dụng một số thư viện cần thiết.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from wordcloud import WordCloud
5 from matplotlib import rcParams
6 from PIL import Image
```

Tương tự như những bài hướng dẫn trên, ta sẽ thực hiện việc lấy dữ liệu ở cột *ten* như sau:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from wordcloud import WordCloud
5 from matplotlib import rcParams
6 from PIL import Image
7 import warnings
8 warnings.filterwarnings("ignore")
9
10 data = pd.read_csv('diemthidhqg2020.csv')
11 ten = data['ten'].dropna()
```

Chương trình 8.11: Lấy dữ liệu tên

Sau khi lấy được dữ liệu tên, ta tiến hành gộp các tên thành một chuỗi phân biệt bởi khoảng cách. Chuẩn bị cho việc vẽ kết quả thống kê.

```
1 ten_string = ' '.join(ten)
```

Chương trình 8.12: Gộp tên thành một chuỗi

Để hiển thị các tên theo hình dạng nhất định, trước tiên ta phải kiếm một hình trắng đen, ở đây sẽ sử dụng hình đất nước Việt Nam.



Hình 8.10: Hình đất nước Việt Nam màu trắng đen.

Hình đất nước Việt Nam sẽ được sử dụng như một dạng khuôn để hiển thị các dữ liệu. Ta sẽ có chương trình đầy đủ sau:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from wordcloud import WordCloud
5 from matplotlib import rcParams
6 from PIL import Image
7 import warnings
8 warnings.filterwarnings("ignore")
9
10 data = pd.read_csv('diemthidhqg2020.csv')
11
12 ten = data['ten'].dropna()
13 ten_string = ' '.join(ten)
14
15 rcParams['figure.figsize'] = 10,15
16 mask = np.array(Image.open('vn_mask.jpg'))
17 wordcloud = WordCloud(max_words=len(ten), background_color=
18     "white", mask=mask, max_font_size=100, scale=8,
19     collocations=False).generate(ten_string)
20
21 plt.imshow(wordcloud)
22 plt.axis('off')
23 plt.show()
```

Chương trình 8.13: Chương trình hoàn thiện

4 Câu hỏi ôn tập

1. Để vẽ biểu đồ đường, ta sử dụng lệnh nào?
A. `plot()`
B. `bar()`
C. `pie()`
D. Không có lệnh nào
2. Để vẽ biểu đồ cột, ta sử dụng lệnh nào?
A. `plot()`
B. `bar()`
C. `pie()`
D. Không có lệnh nào
3. Để vẽ biểu đồ tròn, ta sử dụng lệnh nào?
A. `plot()`
B. `bar()`
C. `pie()`
D. Không có lệnh nào
4. Hàm `value_counts()` có chức năng gì?
A. Vẽ đồ thị
B. Thống kê giá trị xuất hiện bao nhiêu lần trong mảng
C. Đếm số lượng phần tử của mảng
D. Sắp xếp giá trị của mảng
5. Chức năng chính của hàm `sort_index()` là gì?
A. Tìm kiếm
B. Sao chép
C. Xoá
D. Sắp xếp
6. Hàm nào dùng để hiển thị biểu đồ?
A. `show()`
B. `plot()`
C. `bar()`
D. `pie()`
7. Để làm tròn số a ở mức 0.25 đơn vị, ta sử dụng công thức biến đổi nào?
A. $a = \text{round}(a * 1) / 1$
B. $a = \text{round}(a * 2) / 2$
C. $a = \text{round}(a * 4) / 4$
D. $a = \text{round}(a * 8) / 8$

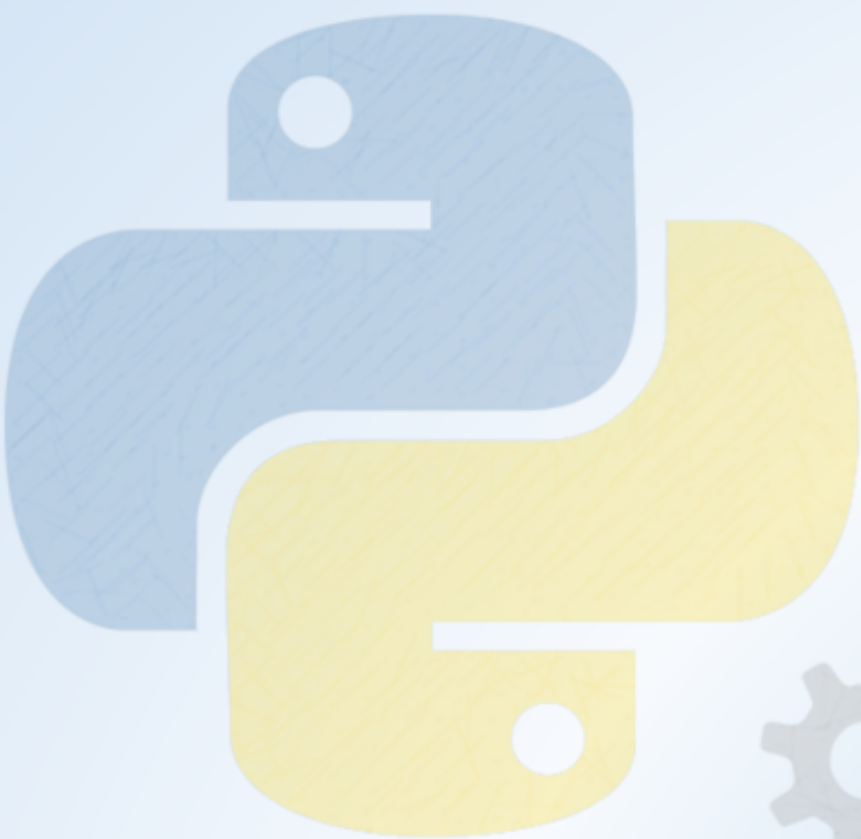
Đáp án

1. A 2. B 3. C 4. B 5. D 6. A 7. C



CHƯƠNG 9

Dự đoán tương lai gần bằng thuật toán Hồi Quy Tuyến Tính



Pandas

1 Giới thiệu

Một trong những yêu cầu của việc phân tích dữ liệu là dự đoán dữ liệu ở tương lai gần. Bằng cách xem xét giá trị lịch sử, một số giải thuật có sẵn có thể ước lượng được giá trị trong tương lai gần với độ chính xác khá cao. Các giải thuật này đã được nghiên cứu bởi các nhà khoa học phân tích dữ liệu.

Trong bài hướng dẫn này, chúng ta sẽ sử dụng dữ liệu về giá đất theo diện tích của năm 2019 và dự đoán giá đất. Chúng ta sẽ tiến hành thực hiện các bước để phân tích dữ liệu và hiện thực dự đoán giá đất theo diện tích bằng thuật toán Hồi Quy Tuyến Tính. Dữ liệu để thực hành trong bài này có thể được tải từ đường dẫn sau đây:

https://ubc.sgpl.cdn.digitaloceanspaces.com/PythonData/gia_nha_dat.csv

2 Xử lý dữ liệu

Do dữ liệu ở phần này đã được xử lý sẵn, nên ta chỉ cần sử dụng trực tiếp mà không cần thông qua các bước nào khác. Ta tiến hành thêm dữ liệu vào chương trình.

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 data = pd.read_csv('gia_nha_dat.csv').values
```

Chương trình 9.1: Thêm dữ liệu vào chương trình

Ở đây, ta sẽ tự hiểu rằng cột đầu tiên là diện tích (tính bằng mét vuông - m^2), cột thứ hai là giá tiền (nhân với 1.000vnd).

3 Biểu diễn số liệu dưới dạng đồ thị

Như những bài trước, vẽ đồ thị này sẽ khá đơn giản. Mục đích của việc vẽ đồ thị là cho chúng ta thấy được mối liên hệ giữa diện tích và giá đất.

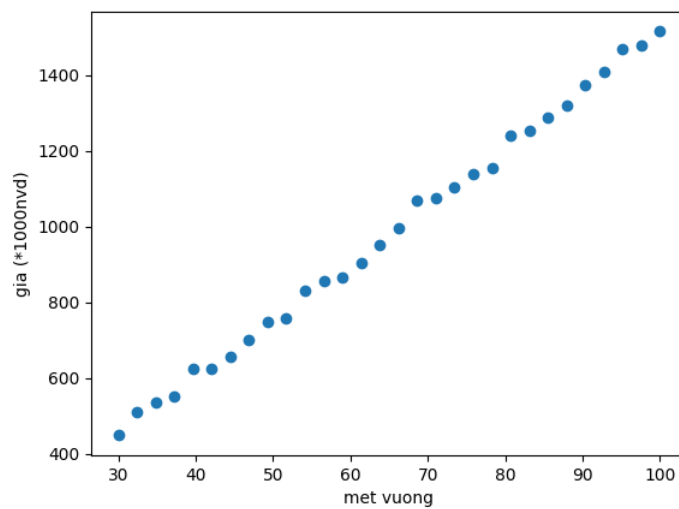
```
1 N = data.shape[0]
2 x = data[:,0].reshape(-1,1)
3 y = data[:,1].reshape(-1,1)
4 plt.scatter(x,y)
5 plt.xlabel('met vuong')
6 plt.ylabel('gia (*1000nvd)')
7 plt.show()
```

Chương trình 9.2: Vẽ đồ thị

Khi nhìn vào đồ thị, ta sẽ thấy được xu hướng tăng lên của giá đất khi diện tích tăng. Hướng tăng lên như một đường thẳng, không bị biến động đột ngột. Nên ở phần tiếp theo, ta sẽ sử dụng thuật toán Hồi Quy Tuyến Tính để tìm ra phương trình của đường thẳng đại diện cho mối liên hệ giữa giá nhà đất và diện tích đất.

Diện tích (m ²)	Giá (* 1.000vnd)	Năm
30	448.524	2019
32.4138	509.248	2019
34.8276	535.104	2019
37.2414	551.432	2019
39.6552	623.418	2019
42.069	625.992	2019
44.4828	655.248	2019
46.8966	701.377	2019
49.3103	748.918	2019
51.7241	757.881	2019
54.1379	831.004	2019
56.5517	855.409	2019
58.9655	866.707	2019
61.3793	902.545	2019
63.7931	952.261	2019
66.2069	995.531	2019
68.6207	1069.78	2019
71.0345	1074.42	2019
73.4483	1103.88	2019
75.8621	1138.69	2019
78.2759	1153.13	2019

Hình 9.1: Dữ liệu nhà đất năm 2019.



Hình 9.2: Biểu đồ về giá và diện tích nhà.

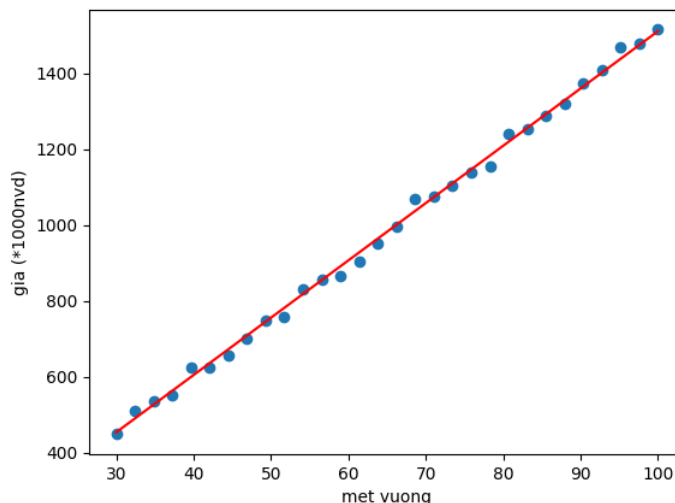
4 Sử dụng thuật toán Hồi Quy Tuyến Tính trong việc dự đoán tương lai gần

Xử lý thuật toán Hồi Quy Tuyến Tính:

```
1 x = np.hstack((np.ones((N, 1)), x))
2 w = np.array([0.,1.]).reshape(-1,1)
3 numOfIrentation = 100
4 cost = np.zeros((numOfIrentation,1))
5 learning_rate = 0.000001
6 for i in range(1, numOfIrentation):
7     r = np.dot(x,w) - y
8     cost[i] = 0.5*np.sum(r*r)
9     w[0] -= learning_rate*np.sum(r)
10    w[1] -= learning_rate*np.sum(np.multiply(r, x[:,1]).
    reshape(-1,1)))
11    print(cost[i])
12 predict = np.dot(x,w)
13 plt.plot((x[0][1], x[N-1][1]), (predict[0], predict[N-1]),
    'r')
14 plt.show()
```

Chương trình 9.3: Xử lý thuật toán

Sau khi xử lý xong, ta sẽ tìm được một đường thẳng (màu đỏ), biểu diễn cho phương trình dự đoán giá dựa theo thuật toán Hồi Quy Tuyến Tính.



Hình 9.3: Biểu đồ thể hiện phương trình để dự đoán giá.

Ta sẽ định nghĩa 1 hàm để chạy phương trình dự đoán và xuất ra kết quả:

```
1 def du_doan_gia_dat(x1):
2     y1 = w[0] + w[1] * x1
3     print('Gia nha cho ' + str(x1) + 'm2 la: ', y1)
```

Chương trình 9.4: Định nghĩa hàm dự đoán giá đất

Lấy ví dụ với mảnh đất có diện tích là 294 mét vuông, ta sẽ sử dụng như sau:

```
1 du_doan_gia_dat(294)
2
3 # Gia nha cho 294m2 la: [4442.4948527]
```

Chương trình hoàn thiện của việc sử dụng thuật toán Hồi Quy Tuyến Tính để dự đoán giá đất.

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 def du_doan_gia_dat(x1):
6     y1 = w[0] + w[1] * x1
7     print('Gia nha cho ' + str(x) + 'm2 la: ', y1)
8
9 data = pd.read_csv('gia_nha_dat.csv').values
10 N = data.shape[0]
11 x = data[:,0].reshape(-1,1)
12 y = data[:,1].reshape(-1,1)
13 plt.scatter(x,y)
14 plt.xlabel('met vuong')
15 plt.ylabel('gia (*1000nvd)')
16 x = np.hstack((np.ones((N, 1)), x))
17 w = np.array([0.,1.]).reshape(-1,1)
18 numOfIrentation = 100
19 cost = np.zeros((numOfIrentation,1))
20 learning_rate = 0.000001
21 for i in range(1, numOfIrentation):
22     r = np.dot(x,w) - y
23     cost[i] = 0.5*np.sum(r*r)
24     w[0] -= learning_rate*np.sum(r)
25     w[1] -= learning_rate*np.sum(np.multiply(r, x[:,1]).
26     reshape(-1,1)))
27     print(cost[i])
28 predict = np.dot(x,w)
29 plt.plot((x[0][1], x[N-1][1]), (predict[0], predict[N-1]),
30     'r')
31 plt.show()
32
33 du_doan_gia_dat(294)
34
35 # Gia nha cho 294m2 la: [4442.4948527]
```

Chương trình 9.5: Chương trình hoàn thiện

5 Câu hỏi ôn tập

- Mục đích của việc đánh giá trực quan của dữ liệu bằng đồ thị là gì?
 - Thấy được xu hướng thay đổi của dữ liệu
 - Đưa ra các dự đoán cho dữ liệu
 - Lựa chọn giải thuật phù hợp cho việc dự đoán dữ liệu
 - Tất cả đều đúng
- Hồi Quy Tuyến Tính sẽ vẽ ra một đồ thị hình gì?
 - Đường cong
 - Đường thẳng
 - Đường parabol
 - Đường gợn sóng
- Tại sao Hồi Quy Tuyến Tính không thể dự đoán được tương lai xa?
 - Vì dữ liệu đưa vào còn ít và đơn giản
 - Không thể dự đoán tương lai xa vì không đoán trước được những ảnh hưởng trong tương lai
 - Sai số sẽ rất lớn khi dự đoán tương lai xa vì thuật toán này chỉ phục vụ trong hoàn cảnh đơn giản
 - Tất cả đều đúng
- Trong các quá trình để đi đến việc dự đoán trong tương lai gần, đâu là quá trình khó nhất?
 - Thu thập dữ liệu
 - Xử lý dữ liệu
 - Biểu diễn dữ liệu dưới dạng đồ thị
 - Đánh giá dữ liệu để đưa ra thuật toán áp dụng phù hợp
- Tại sao quá trình **thu thập dữ liệu** là quá trình khó khăn nhất và quan trọng nhất?
 - Dữ liệu là vấn đề rất nhạy cảm
 - Dữ liệu không phù hợp sẽ gây sai lệch, ảnh hưởng đến các quyết định
 - Dữ liệu là riêng tư, không thể trực tiếp công bố
 - Tất cả đều đúng
- Có phải tất cả bài toán đều có thể giải quyết bằng Hồi Quy Tuyến Tính hay không?
 - Đúng
 - Sai
- Thuật toán Hồi Quy Tuyến Tính chỉ có thể giải quyết những bài toán mức độ nào?
 - Cực kỳ đơn giản
 - Đơn giản
 - Trung Bình
 - Phức tạp
 - Cực kỳ phức tạp

Đáp án

1. D 2. B 3. D 4. A 5. D 6. B 7. A

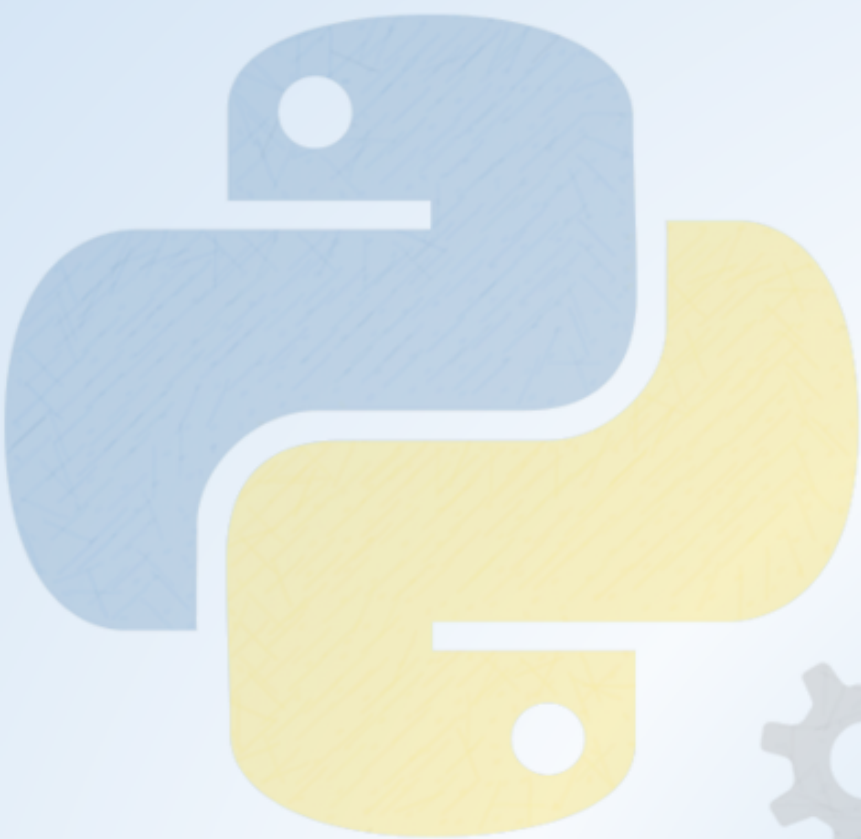
Phần IV

Robotics với Python - Ứng dụng Google Home



CHƯƠNG 10

Trợ lý ảo trên Điện thoại



Pandas

1 Giới thiệu

Ngày nay, chúng ta đã rất quen thuộc với những hệ thống trợ lý ảo. Mặc dù hiện tại, các hệ thống trợ lý ảo đa số là hoạt động đơn lẻ trên thiết bị di động, và phục vụ cho các nhu cầu về giải trí là nhiều, trong tương lai, tiềm năng của nó còn lớn hơn thế rất nhiều.

Hãy thử nhìn lại về máy tính để bàn hoặc laptop mà chúng ta đang xài, bạn sẽ thấy rằng không còn nút nguồn (Power) để tắt nguồn điện hoàn toàn của hệ thống nữa, mà thay vào đó là một nút tín hiệu, phải nhấn đè thì máy tính mới tắt nguồn. Xa hơn, điện thoại di động hay tivi thông minh, cũng dần loại bỏ nút nguồn và thay vào đó là chế độ chờ (stand-by). Lý do là các thiết bị hiện đại đa số phải dựa trên hệ điều hành để có thể đáp ứng được nhu cầu ngày càng đa dạng và phức tạp từ phía người dùng. Mà một hệ thống có hệ điều hành sẽ mất nhiều thời gian hơn để khởi động. Do đó, để tạo cảm giác nhanh với người dùng, các hệ thống này thường ở chế độ chờ chứ không tắt hoàn toàn. Thêm nữa, khi không tắt thiết bị, nó sẽ trở nên bền hơn. Ví dụ như máy tính của chúng ta, khi phải tắt nguồn nóng, hệ điều hành sẽ rất dễ bị hỏng và không thể khởi động được nữa.

Khi ở trạng thái chờ, hệ thống cần một tín hiệu để kích hoạt trở lại. Hiện tại, chúng ta đang chủ yếu phải tương tác bằng tay thông qua remote. Tuy nhiên, trong tương lai rất gần sẽ là giọng nói, mà sản phẩm Google Home là một ví dụ. Do đó, các hệ thống trợ lý ảo hiện tại là để mở đường cho các ứng dụng thông minh trong tương lai.

Trong bài hướng dẫn này, chúng tôi sẽ hỗ trợ bạn đọc cài đặt một phần mềm trên điện thoại di động Android để mô phỏng như một trợ lý ảo. Trong những bài tiếp theo, chúng ta sẽ sử dụng trợ lý ảo này để điều khiển các thiết bị khác, như bật tắt đèn, mở quạt chẳng hạn. Các mục tiêu của bài hướng dẫn này được tóm tắt như sau:

- Cài đặt công cụ phát âm tiếng Việt cho ứng dụng
- Cài đặt ứng dụng Trợ lý ảo trên Android
- Hiểu được nguyên lý vận hành của phần mềm

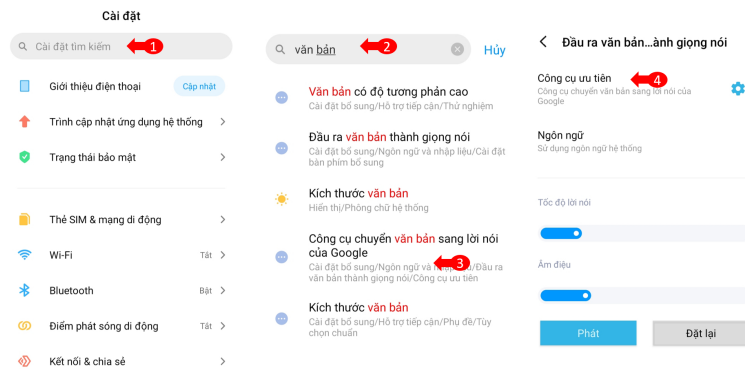
2 Cài đặt Giọng nói - Văn bản

Phần mềm trợ lý ảo hỗ trợ việc tương tác lại với chúng ta bằng giọng nói tiếng Việt. Do vậy, chúng ta phải đảm bảo công cụ này được chính xác. Để kiểm tra, bạn sẽ làm các bước như bên dưới (Xem minh họa ở Hình 10.1):

Bước 1: Vào mục Cài đặt của điện thoại.

Bước 2: Tìm kiếm công cụ Văn bản thành giọng nói.

Bước 3: Sử dụng công cụ giọng nói của Google.



Hình 10.1: Các thao tác để chỉnh công cụ văn bản thành giọng nói

Lưu ý: Trong trường hợp không có giọng nói của Google, bạn đọc có thể lên Chợ ứng dụng (CH Play) để cài thêm ứng dụng Google LLC.

3 Cài đặt ứng dụng Trợ lý ảo

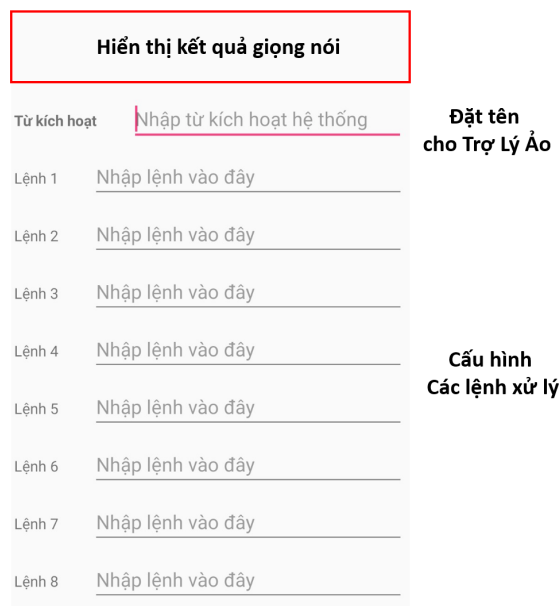
Ứng dụng trợ lý ảo có thể tải về từ đường dẫn sau đây:

<https://ubc.sgp1.cdn.digitaloceanspaces.com/AndroidAppBBC/TroLyAo.apk>

Trong quá trình sử dụng, bạn đọc có thể kiểm tra đường dẫn trên để tải các bản cập nhật mới hơn. Sau khi tải file apk, bạn cài đặt như bình thường.

Lưu ý: Trong quá trình cài đặt, nếu hệ thống cần quyền gì, bạn hãy cho phép nó.

Sau khi cài đặt xong, giao diện của ứng dụng sẽ hiện lên như sau:



Hình 10.2: Giao diện phần mềm Trợ Lý Ảo

Khi khởi động ứng dụng, phần mềm sẽ cần quyền truy cập USB (để kết nối với mạch Microbit) và quyền ghi âm (để nhận dạng giọng nói). Bạn đọc hãy lưu ý để kích hoạt các quyền này. Nếu như vì lý do nào đó, bạn không cho phép quyền, hãy vào mục Cài đặt của điện thoại để kích hoạt nó lên.

4 Hướng dẫn sử dụng

4.1 Đặt tên cho Trợ lý ảo

Cũng giống như các hệ thống thương mại có sẵn, bạn sẽ dùng một từ khóa để kích hoạt hệ thống, chẳng hạn như "OK Google" hoặc "Alexa". Ở đây, hệ thống của chúng tôi cũng hỗ trợ bạn đặt tên cho trợ lý ảo của mình. Tất nhiên, để kích hoạt nó, bạn cần nói lại tên mà mình đã đặt cho nó.

4.2 Cấu hình lệnh cho trợ lý ảo

Đây là bước cấu hình quan trọng nhất cho hệ thống này. Hiện tại, hệ thống hỗ trợ 20 lệnh tối đa, được đánh số từ 1 cho tới 20. Với một lệnh, nó có 2 chức năng, như sau:

- Nhận diện giọng nói từ người dùng: Nếu như người sử dụng ra lệnh trùng với lệnh được cài đặt, thứ tự của lệnh sẽ được gửi đến mạch Microbit trung tâm, và gửi đi tới các mạch Microbit khác có trong hệ thống. Ví dụ Lệnh 1 là "Bật đèn phòng khách". Khi chúng ta nói lệnh tương tự như vậy, số 1 sẽ được gửi đi tới các mạch Microbit.
- Thông báo bằng giọng nói với người dùng: Khi trợ lý ảo nhận được thông tin từ mạch Microbit, là các con số từ 1 tới 20. Lúc đó, trợ lý ảo sẽ nói ra câu tiếng Việt tương ứng với thứ tự của câu lệnh được thiết lập.

5 Câu hỏi ôn tập

1. Công cụ phát âm tiếng Việt trên điện thoại gọi là gì?
 - A. Công cụ nhận dạng giọng nói
 - B. Công cụ văn bản - giọng nói
 - C. Công cụ nhận dạng tiếng Việt
 - D. Công cụ cài đặt tiếng Việt
2. Công cụ phát âm ra tiếng Việt trên điện thoại Android là một ứng dụng của:
 - A. Google
 - B. Microsoft
 - C. Apple
 - D. Tất cả các phần mềm trên
3. Trợ lý ảo giới thiệu trong bài hướng dẫn thực ra là?
 - A. Dịch vụ có sẵn của Google
 - B. Ứng dụng được dựa trên dịch vụ có sẵn của Google
 - C. Dịch vụ có sẵn trên mọi hệ điều hành
 - D. Tất cả đều đúng
4. Có tất cả tối đa bao nhiêu lệnh được hỗ trợ trên phần mềm Trợ lý ảo?
 - A. 10
 - B. 20
 - C. 40
 - D. Bao nhiêu cũng được
5. Mỗi câu lệnh được cài đặt có chức năng gì?
 - A. Nhận diện giọng nói - gửi đi số thứ tự của câu lệnh
 - B. Nhận số thứ tự - phát ra giọng nói tương ứng
 - C. Tất cả đều đúng
 - D. Tất cả đều sai
6. Lý do để các trợ lý ảo bằng giọng nói sẽ thông dụng trong tương lai là gì?
 - A. Các hệ thống đa phần ở chế độ chờ
 - B. Tương tác bằng giọng nói là kĩ thuật hiện đại và tiện dụng
 - C. Từ kích hoạt hệ thống là kĩ thuật đã có sẵn của Google, Amazon hay IBM
 - D. Tất cả đều đúng
7. Các quyền cần được cho phép khi cài ứng dụng Trợ lý ảo là gì?
 - A. Quyền ghi âm
 - B. Quyền kết nối USB
 - C. Tất cả 2 quyền trên
 - D. Không cần quyền gì cả

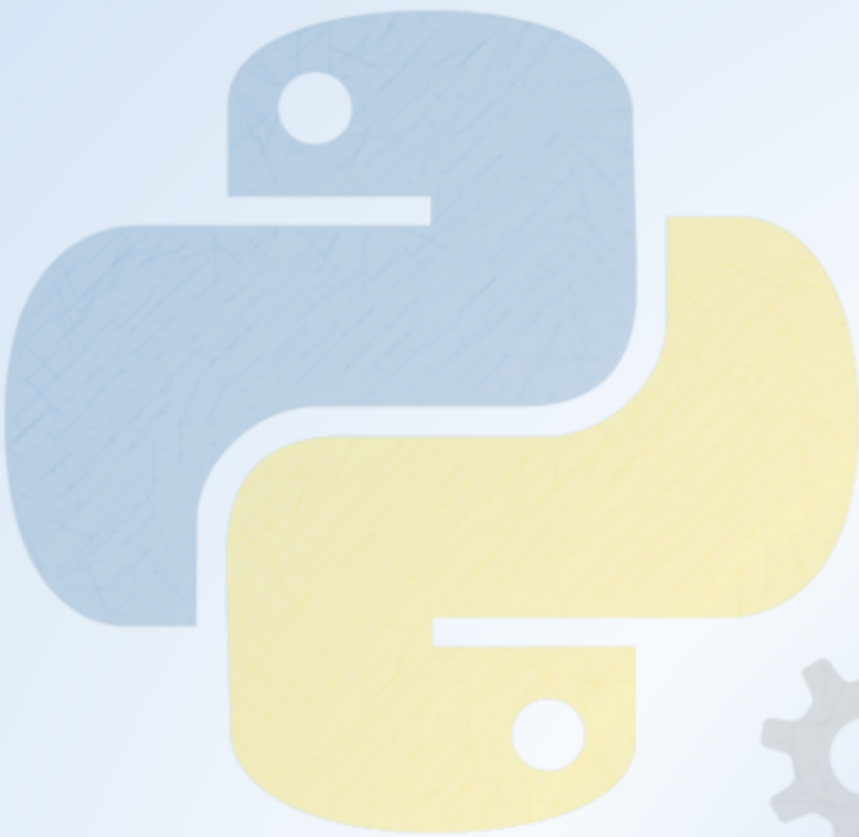
Đáp án

1. B 2. A 3. D 4. B 5. C 6. D 7. C



CHƯƠNG 11

Tích hợp Microbit vào Trợ lý ảo



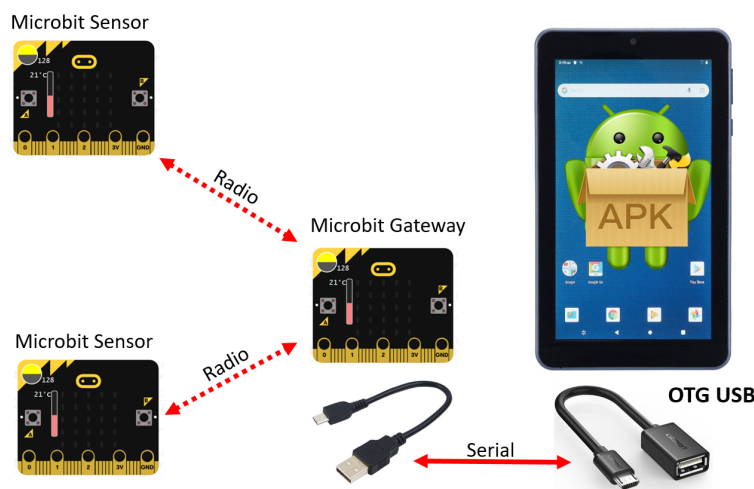
Pandas



1 Giới thiệu

Trong bài hướng dẫn này, chúng ta sẽ tích hợp mạch Microbit với Trợ lý ảo trên điện thoại. Đây sẽ là điều rất thuận tiện cho bạn đọc có thể phát triển nhiều ứng dụng phong phú. Tận dụng khả năng những công nghệ AI có sẵn trên điện thoại, một lệnh đơn giản sẽ được truyền xuống mạch Microbit cho những xử lý tiếp theo.

Để làm được điều này, chúng ta cần một kết nối giữa điện thoại và mạch Microbit. Ở đây, chúng tôi sẽ dùng một dây USB OTG để kết nối giữa điện thoại và mạch Microbit. Sở dĩ chúng tôi không sử dụng kết nối không dây là vì muốn dùng nó để gửi lệnh cho các mạch Microbit khác nữa. Mô hình kết nối giữa điện thoại và mạch Microbit được minh họa như hình bên dưới:



Hình 11.1: Mô hình kết nối giữ điện thoại và mạch Microbit trung tâm

Mạch Microbit kết nối với điện thoại, sẽ đóng vai trò là mạch trung tâm (hay còn gọi là Gateway). Mạch này sẽ có 2 nhiệm vụ chính. Đầu tiên nó sẽ nhận lệnh từ điện thoại và gửi cho các mạch Microbit khác để hình thành nên nhiều ứng dụng. Ngoài ra, nó cũng có khả năng nhận dữ liệu từ các mạch Microbit khác và gửi lên cho điện thoại.

Chúng ta cũng sẽ dùng ngôn ngữ Python để lập trình cho mạch Microbit trung tâm thay vì dùng ngôn ngữ kéo thả MakeCode. Thông qua hướng dẫn này, chúng tôi cũng muốn bạn đọc thấy được sự phong phú của ngôn ngữ Python, khi nó có thể hỗ trợ lập trình được cho nhiều nền tảng khác nhau. Tuy nhiên, chúng ta sẽ dùng phần mềm Mu để viết chương trình cho mạch Microbit. Mục tiêu của bài hướng dẫn này được tổng hợp như bên dưới:

- Lập trình Python cho Microbit bằng phần mềm Mu
- Nhận dữ liệu từ điện thoại thông qua kết nối Serial
- Giao tiếp dữ liệu không dây với các mạch Microbit khác thông qua kết nối Radio

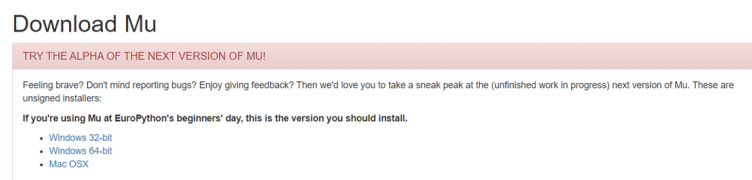
2 Cài đặt phần mềm Mu

Bước 1: Vào trang chủ <https://codewith.mu/> và chọn tiếp vào **Download**, như minh họa ở hình bên dưới:



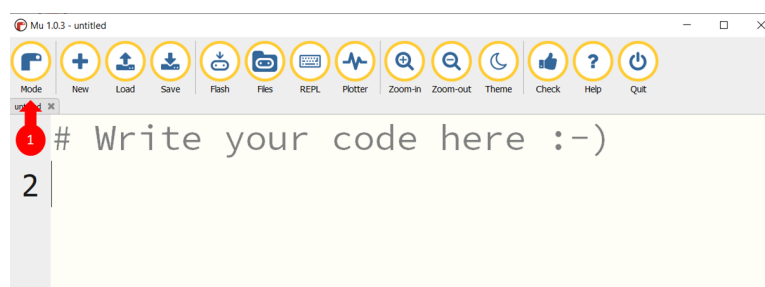
Hình 11.2: Vào trang chủ của Mu và tải phần mềm

Bước 2: Chọn phần mềm tương thích với hệ điều hành của máy tính. Với cái máy tính Windows 10 hiện tại, là dòng 64 bit.



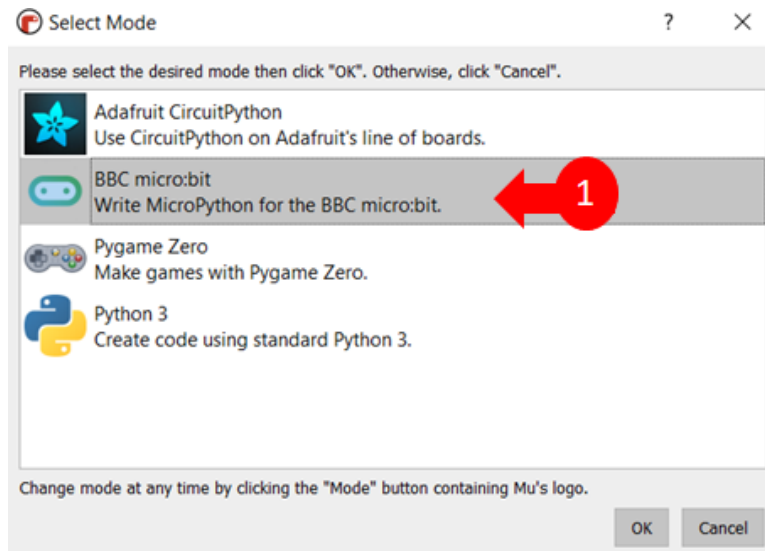
Hình 11.3: Chọn phiên bản phù hợp với cấu hình của máy tính

Bước 3: Sau khi việc cài đặt hoàn thành, giao diện sau đây của phần mềm sẽ như sau:



Hình 11.4: Giao diện của phần mềm Mu

Bước 4: Cấu hình cho phần mềm là trình soạn thảo cho mạch Microbit bằng cách nhấn vào nút **Mode** ở Hình 11.4, giao diện sau đây sẽ hiện ra:



Hình 11.5: Cấu hình cho phần mềm là trình soạn thảo cho Microbit

Bước 5: Cắm mạch Microbit vào máy tính, nút **Flash** được kích hoạt và chúng ta đã có thể lập trình cho nó. Trong bước này, có thể sẽ có thông báo hiện lên, báo hiệu việc nạp không thành công, với các lý do sau đây

- Bên trong mạch Microbit đã được nạp trước một chương trình thông qua MakeCode nên không tương thích với hệ thống Python. Bạn chỉ cần nạp lại là được.
- Mạch Microbit đang được ghép đôi với phần mềm nào đó. Điều này rất hay xảy ra khi bạn quên tắt phần mềm soạn thảo trên MakeCode. Đối với người lập trình thông thạo, sẽ thường ghép đôi phần mềm với mạch Microbit để nạp cho nhanh. Do đó, cần phải tắt phần mềm MakeCode để giành quyền kết nối cho phần mềm Mu.

3 Chương trình đầu tiên

Cũng tương tự như chương trình đầu tiên với ngôn ngữ kéo thả MakeCode, chúng ta sẽ hiện thực lại chương trình này, nhưng bằng ngôn ngữ Python trên trình soạn thảo Mu. Chương trình gợi ý cho chúng ta như sau:

```
1 from microbit import *
2
3 while True:
4     display.show(Image.HEART)
5     sleep(1000)
6     display.show(Image.HEART_SMALL)
7     sleep(1000)
```

Chương trình 11.1: Chương trình đầu tiên trên Mu

Các câu lệnh python trong chương trình này không quá khó để có thể hiểu được, nếu bạn đọc so sánh nó với chương trình bên MakeCode. Câu lệnh đầu tiên (**import**) là để khai báo các thư viện hỗ trợ cho việc lập trình trên mạch Microbit. Vòng lặp

while True: tương tự với khối **forever**. Bên trong nó, tương đương với 2 câu lệnh là **show icons** và **pause** bên môi trường MakeCode.

4 Hiện thực chương trình cho Microbit

Bây giờ, chúng ta sẽ hiện thực chương trình hoàn chỉnh cho mạch Microbit để nó có thể tích hợp vào hệ thống trợ lý ảo. Trong chương trình này, chúng ta cần 2 chức năng quan trọng:

- Radio: Để giao tiếp không dây với các mạch Microbit khác
- Serial: Để giao tiếp giữ mạch Microbit đang kết nối với điện thoại Android

Chúng ta bắt đầu chương trình bằng việc khởi tạo các khối này, như sau:

```
1 from microbit import *
2 import radio
3
4 radio.config(group=1)
5 radio.on()
6
7 uart.init(baudrate=115200)
```

Chương trình 11.2: Chương trình đầu tiên trên Mu

Chúng ta cần lưu ý rằng, các mạch Microbit chỉ có thể giao tiếp với nhau nếu chúng có cùng nhóm. Tiếp theo, khối uart dùng để giao tiếp với điện thoại, phải sử dụng tốc độ 115200, vì đây là cổng uart ảo thông qua kết nối USB.

4.1 Nhận dữ liệu từ điện thoại

Với chức năng này, mỗi khi trợ lý ảo gửi lệnh xuống, chúng ta sẽ gửi lệnh này không dây bằng khối Radio để các mạch Microbit khác có thể nhận được. Chức năng này sẽ được hiện thực trong vòng lặp vô tận, như sau:

```
1
2 from microbit import *
3 import radio
4
5 radio.config(group=1)
6 radio.on()
7
8 uart.init(baudrate=115200)
9
10 while True:
11
12     msg_bytes = uart.read()
13     if msg_bytes != None:
14         msg_str = str(msg_bytes, 'UTF-8')
15         radio.send(msg_str)
```

```
16 sleep(100)
```

Chương trình 11.3: Nhận dữ liệu từ điện thoại và phát ra Radio

4.2 Gửi dữ liệu lên điện thoại

Ngược lại, mỗi khi mạch Microbit trung tâm nhận được thông tin gì từ các mạch Microbit khác, nó cũng sẽ gửi lên trợ lý ảo để xử lý thêm, chương trình này cũng sẽ được hiện thực tiếp trong khối lặp vô tận, như sau:

```
1 from microbit import *
2 import radio
3
4 radio.config(group=1)
5 radio.on()
6
7 uart.init(baudrate=115200)
8
9 while True:
10
11     msg_bytes = uart.read()
12     if msg_bytes is not None:
13         msg_str = str(msg_bytes, 'UTF-8')
14         radio.send(msg_str)
15
16     receive = radio.receive()
17     if receive is not None:
18         uart.write(receive)
19
20     sleep(100)
```

Chương trình 11.4: Nhận dữ liệu từ Radio và gửi lên điện thoại

5 Câu hỏi ôn tập

1. Mạch Microbit trung tâm kết nối với Điện thoại thông qua:
A. USB OTG
B. USB 2.0
C. USB 3.0
D. Bluetotoh
2. Phần mềm nào sau đây hỗ trợ lập trình Python trên mạch Microbit?
A. MakeCode online
B. MakeCode offline
C. Mu
D. Tất cả các phần mềm trên
3. Công cụ gửi nhận dữ liệu giữa mạch Microbit và điện thoại thông qua kết nối USB là gì?
A. UART
B. SPI
C. I2C
D. Tất cả đều đúng
4. Tốc độ giao tiếp giữa điện thoại và mạch Microbit thông qua giao tiếp nối tiếp là bao nhiêu?
A. 9600
B. 19200
C. 115200
D. Tốc độ nào cũng được
5. Câu lệnh nào sau đây dùng để chỉnh nhóm cho giao tiếp Radio là 1?
A. `radio.config(1)`
B. `radio.group(1)`
C. `radio.config(group=1)`
D. Tất cả đều đúng
6. Câu lệnh nào dùng để bật chức năng giao tiếp Radio của mạch microbit?
A. `radio.on()`
B. `radio.turn_on()`
C. `radio.active()`
D. Tất cả đều sai
7. Câu lệnh nào sau đây để kiểm tra biến receive khác null?
A. `receive == None`
B. `receive != None`
C. `receive is not None`
D. Tất cả đều đúng

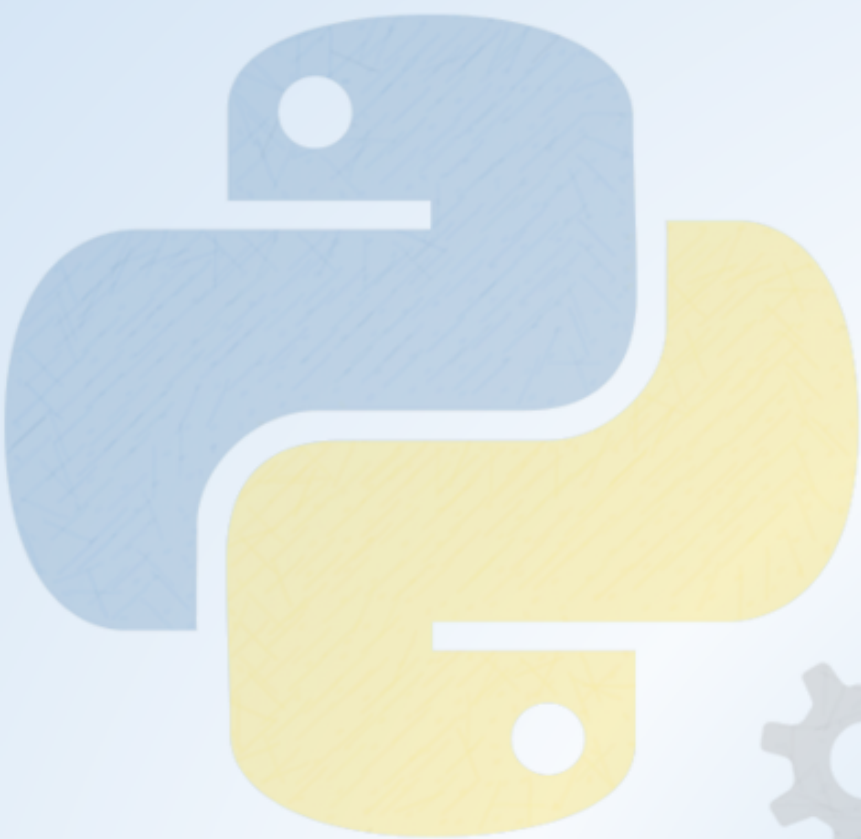
Đáp án

1. A 2. D 3. A 4. C 5. C 6. A 7. C



CHƯƠNG 12

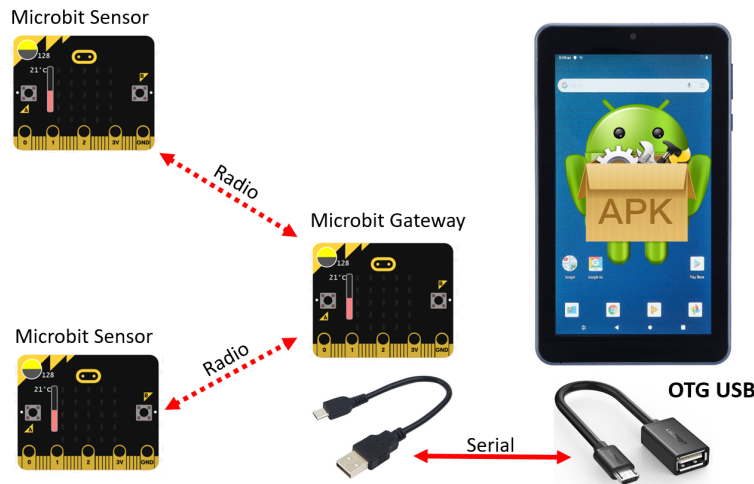
Điều khiển ngoại vi qua giao tiếp không dây



Pandas

1 Giới thiệu

Trở lại với mô hình kết nối giữa điện thoại và mạch Microbit như hình bên dưới, trong bài hướng dẫn này, chúng ta sẽ tập trung hiện thực chương trình cho các nút cảm biến (Microbit Sensor) để xử lý lệnh từ trợ lý ảo từ điện thoại.



Hình 12.1: Mô hình kết nối giữa điện thoại và mạch Microbit trung tâm

Về căn bản, một nút cảm biến sẽ có 2 chức năng như sau:

- Nhận lệnh từ trung tâm: Lệnh này sẽ được gửi không dây tới nút cảm biến. Nút cảm biến cần phải phân loại và thực hiện lệnh tương ứng mà nó được thiết lập.
- Gửi thông báo về trung tâm: Trong trường hợp giá trị giám sát vượt miền cho phép, nó có thể gửi dữ liệu không dây về nút trung tâm để trợ lý ảo thông báo cho người dùng.

Với những chức năng phân tích bên trên, rõ ràng một nút cảm biến sẽ phải cần khối Radio để nhận và gửi dữ liệu không dây. Khối Serial trong trường hợp này chưa cần dùng đến. Cuối cùng, nút cảm biến sẽ tập trung vào việc điều khiển ngoại vi và đọc dữ liệu cảm biến. Các mục tiêu của bài hướng dẫn này như sau:

- Nhận dữ liệu không dây từ trung tâm để điều khiển ngoại vi
- Giám sát cảm biến và gửi cảnh báo từ trung tâm
- Đọc được cảm biến nhiệt độ và ánh sáng trên mạch Microbit

2 Nhận dữ liệu không dây

Đây là chức năng cơ bản đầu tiên: nút cảm biến sẽ nhận lệnh từ trung tâm thông qua giao tiếp không dây. Chúng ta sẽ kế thừa một phần chương trình từ nút trung tâm để hiện thực chức năng này, như sau:

```

1 from microbit import *
2 import radio
3
4 radio.config(group=1)
5 radio.on()
6
7 def process_command(cmd):
8     display.scroll(cmd)
9     return
10
11 while True:
12
13     receive = radio.receive()
14     if receive is not None:
15         process_command(receive)
16
17     sleep(100)

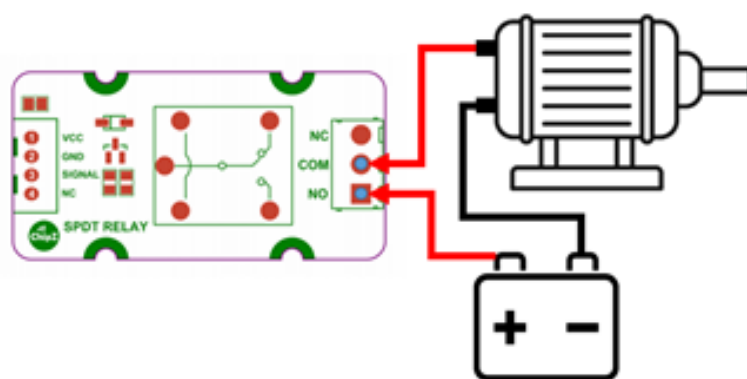
```

Chương trình 12.1: Chương trình đầu tiên trên Mu

Trong chương trình trên, chúng ta chỉ đơn giản là hiển thị những gì nhận được. Chức năng cụ thể sẽ được hiện thực chi tiết hơn trong hàm **process_command**.

3 Đóng/Mở công tắc Relay

Relay là một thiết bị sẽ mang lại nhiều hứng thú cho bạn đọc bởi nó như một công tắc mà chúng ta hay xài trong gia đình. Tuy nhiên, công tắc này chúng ta có thể đóng mở được bằng việc lập trình, thay vì phải tương tác bằng tay. Sơ đồ kết nối thiết bị có công suất lớn (như đèn, máy bơm) qua mạch Relay được minh họa như sau:



Hình 12.2: Kết nối giữa Relay và thiết bị tải

Chúng ta sẽ sử dụng 2 chân là COM và NO của relay. Ý nghĩa của NO là Normal Open, tức là bình thường công tắc sẽ hở, mạch điện sẽ không nối và không có dòng điện qua tải. Khi chúng ta lập trình kích hoạt nó, thì công tắc mới đóng và động cơ sẽ hoạt động.

Đầu vào của mạch Relay được chuẩn hóa để phù hợp với mạch mở rộng Chibi Base Shield với chuẩn cắm 4 chân VCC-GND-Sig-NC. Khi cắm vào mạch mở rộng, bạn phải lưu ý chân Sig đang được nối với chân nào của Microbit thì mới có thể lập trình được nó. Trong kết nối của chúng tôi, Sig được nối với chân P0, và chương trình để đóng mở Relay dựa vào tín hiệu từ Trọ lý ảo được gợi ý như sau:

```
1 def process_command(cmd):
2     if receive == "1":
3         display.show(Image.HEART_SMALL)
4         pin0.write_digital(0)
5     elif receive == "2":
6         display.show(Image.HEART)
7         pin0.write_digital(1)
8     return
```

Chương trình 12.2: Hiện thực đóng tắt Relay

4 Giám sát cảm biến và gửi cảnh báo

Một trong những chức năng không thể thiếu của nốt cảm biến là nó sẽ giám sát thông tin của môi trường và gửi cảnh báo tới trung tâm. Việc kết nối cảm biến với mạch Microbit sẽ tùy thuộc vào ứng dụng mà bạn đọc phải chọn lựa cho phù hợp. Trong bài hướng dẫn này, chúng tôi chỉ sử dụng các cảm biến được tích hợp sẵn trên mạch Microbit, là cảm biến nhiệt độ và cảm biến ánh sáng, để minh họa cho chức năng này.

Tuy nhiên, việc đọc giá trị từ cảm biến không cần thiết phải đọc một cách liên tục, chẳng hạn chúng ta cần 5 giây mới đọc 1 lần. Tuy nhiên, nếu chúng ta dùng câu lệnh **sleep(5000)** thì khi gửi dữ liệu từ trung tâm, có thể 5 giây sau hệ thống mới đáp ứng. Do đó, để đảm bảo việc đáp ứng của hệ thống, chúng ta không được phép xài thêm câu lệnh sleep nào nữa, mà chỉ dùng một câu lệnh **sleep(100)** ở cuối vòng lặp **while True** mà thôi.

Để tạo ra hiệu ứng 5 giây mới đọc cảm biến một lần, chúng ta tạo thêm biến **counter** và xử lý như sau:

```
1 from microbit import *
2 import radio
3
4 radio.config(group=1)
5 radio.on()
6
7 counter = 0
8
9 while True:
10
11     receive = radio.receive()
12     if receive is not None:
13         process_command(receive)
```

```

14
15     counter = counter + 1
16     if counter > 50:
17         counter = 0
18         process_sensor()
19
20     sleep(100)

```

Chương trình 12.3: Tạo hàm đọc cảm biến sau mỗi 5 giây

Bạn đọc hãy lưu ý về việc xử lý biến `counter` trong đoạn chương trình trên. Biến này sẽ tăng lên 1 đơn vị sau mỗi 100ms trong vòng lặp `while True`. Do đó, sau 50 lần của 100ms, hàm **`process_sensor`** sẽ được gọi một lần, tức là việc đọc cảm biến sẽ được thực thi sau mỗi 5 giây.

Kĩ thuật sử dụng trong chương trình trên là yêu cầu cơ bản trong việc hiện thực chương trình ngày một lớn hơn: Không sử dụng câu lệnh đợi (ở đây là `sleep`) và không sử dụng câu lệnh lặp. Nếu vi phạm quy tắc này, khi hiện thực thêm một chức năng mới, chức năng cũ có thể sẽ không chạy. Cuối cùng, chúng ta sẽ hiện thực hàm `process_sensor` như sau:

```

1 def process_sensor():
2     temp = temperature()
3     light = display.read_light_level()
4     if temp > 29:
5         radio.send("10")
6     if light < 300:
7         radio.send("11")
8     return

```

Chương trình 12.4: Hiện thực chương trình đọc cảm biến và gửi cảnh báo

Trong chương trình trên, mỗi khi nhiệt độ lớn hơn 29, chúng ta sẽ gửi cảnh báo về trung tâm là số 10. Vì vậy, bạn chỉ cần cài đặt một câu thông báo cho lệnh số 11 trên trợ lý ảo, chẳng hạn như "Nhiệt độ phòng quá nóng" thì trợ lý ảo sẽ thông báo ra bằng giọng nói. Tương tự như vậy, lệnh số 11 trên trợ lý ảo có thể là "Phòng hơi tối, xin vui lòng bật đèn".

5 Câu hỏi ôn tập

1. Microbit cảm biến kết nối với trung tâm thông qua:
A. USB OTG
B. Radio
C. Bluetooth
D. Tất cả đều sai
2. Để đọc được giá trị cảm biến nhiệt độ tích hợp bên trong Microbit, câu lệnh nào sẽ được sử dụng?
A. temp()
B. temperature()
C. getTemperature()
D. Tất cả đều sai
3. Để đọc được giá trị cảm biến ánh sáng tích hợp bên trong Microbit, câu lệnh nào sẽ được sử dụng?
A. display.read_light_level()
B. display.get_light_level()
C. display.light_level()
D. Tất cả đều sai
4. Để gửi dữ liệu không dây từ nút cảm biến sang nút trung tâm, câu lệnh nào được sử dụng?
A. radio.receive("10")
B. radio.send("10")
C. radio.sent("10")
D. Tất cả đều sai
5. Giải pháp nào để tạo hiệu ứng đợi 5 giây hiệu quả nhất?
A. Sử dụng câu lệnh sleep(5000)
B. Sử dụng câu lệnh lặp 50 lần, mỗi lần lặp đợi 100ms
C. Sử dụng biến và xử lý bằng câu lệnh if
D. Tất cả đều đúng
6. Để trợ lý ảo có thể phát ra giọng nói ở lệnh số 10, câu lệnh nào sau đây sẽ được dùng?
A. radio.send(10)
B. radio.send("10")
C. radio.receive(10)
D. radio.receive("10")
7. Giả sử công tắc Relay được nối vào chân P0, câu lệnh nào để bật Relay?
A. write_digital(P0, 1)
B. write_digital(P0, 0)
C. P0.write_digital(1)
D. P0.write_digital(0)

Đáp án

1. B 2. B 3. A 4. B 5. C 6. B 7. C