

BÀI TẬP CUỐI KỲ

Viết một chương trình AI có tên `trafficsign` để nhận diện bảng hiệu giao thông trên đường sử dụng tập dữ liệu mẫu `gtsrb` cho trước để huấn luyện. Bên dưới là một trong những kết quả sau khi chạy chương trình.

python trafficsign.py gtsrb

```
Epoch 1/10
500/500 [=====] - 5s 9ms/step - loss: 3.7139 - accuracy: 0.1545
Epoch 2/10
500/500 [=====] - 6s 11ms/step - loss: 2.0086 - accuracy: 0.4082
Epoch 3/10
500/500 [=====] - 6s 12ms/step - loss: 1.3055 - accuracy: 0.5917
Epoch 4/10
500/500 [=====] - 5s 11ms/step - loss: 0.9181 - accuracy: 0.7171
Epoch 5/10
500/500 [=====] - 7s 13ms/step - loss: 0.6560 - accuracy: 0.7974
Epoch 6/10
500/500 [=====] - 9s 18ms/step - loss: 0.5078 - accuracy: 0.8470
Epoch 7/10
500/500 [=====] - 9s 18ms/step - loss: 0.4216 - accuracy: 0.8754
Epoch 8/10
500/500 [=====] - 10s 20ms/step - loss: 0.3526 - accuracy: 0.8946
Epoch 9/10
500/500 [=====] - 10s 21ms/step - loss: 0.3016 - accuracy: 0.9086
Epoch 10/10
500/500 [=====] - 10s 20ms/step - loss: 0.2497 - accuracy: 0.9256
333/333 - 5s - loss: 0.1616 - accuracy: 0.9535
```

YÊU CẦU:

Để phát triển ô tô tự lái, một trong những thách thức chính là [thị giác máy tính](#), cho phép những ô tô này phát triển sự hiểu biết về môi trường của chúng từ hình ảnh kỹ thuật số. Đặc biệt, điều này liên quan đến khả năng nhận biết và phân biệt các biển báo trên đường – biển báo dừng, biển báo giới hạn tốc độ, biển báo nhường đường, v.v.

Trong bài tập này, các bạn sẽ sử dụng [TensorFlow](#) để xây dựng mạng nơ-ron nhằm phân loại các biển báo giao thông dựa trên hình ảnh của các biển báo đó. Để làm được như vậy, các bạn cần một bộ dữ liệu được gán nhãn trước: một tập hợp các hình ảnh đã được phân loại theo các biển báo giao thông.

Có rất nhiều bộ dữ liệu, nhưng đối với bài tập này, chúng ta sẽ sử dụng **bộ dữ liệu biển báo giao thông của Đức (GTSRB)**, chứa hàng nghìn hình ảnh của **43 loại biển báo đường bộ khác nhau**.

CHO TRƯỚC:

1. Source python mẫu: **Trafficsign**
2. Tập dữ liệu để huấn luyện: **gtsrb**
3. Danh mục thư viện cần cài:
 - a. **opencv-python**
 - b. **scikit-learn**
 - c. **tensorflow**

HƯỚNG DẪN:

Bước 1:

Xem tập dữ liệu bằng cách mở thư mục **gtsrb**. Bạn sẽ thấy 43 thư mục con trong tập dữ liệu này, được đánh số từ 0 đến 42. Mỗi thư mục con được đánh số đại diện cho một danh mục (một loại biển báo giao thông). Trong thư mục của mỗi biển báo giao thông là tập hợp các hình ảnh của loại biển báo giao thông đó.

Bước 2:

Xem file code python **trafficsign.py**. Dữ liệu và các nhãn tương ứng được tải từ thư mục dữ liệu (thông qua hàm **load_data**) và được chia thành tập dữ liệu huấn luyện và tập dữ liệu kiểm tra. Sau đó, dùng hàm **get_model** để xây dựng và biên dịch mô hình mạng nơron. Mô hình sau đó được đánh giá trên tập dữ liệu cho sẵn. Cuối cùng, mô hình sau khi được huấn luyện sẽ được lưu lại và ứng dụng trên ảnh thực tế.

Các hàm **load_data** và **get_model** là các hàm do các bạn viết theo gợi ý trong code mẫu.

THAM KHẢO THÊM:

- [Tensorflow Keras overview](#)
- [OpenCV-Python](#)

YÊU CẦU SINH VIÊN CẦN THỰC HIỆN (tối đa 2 bạn 1 nhóm):

1. Hoàn thiện code chương trình theo yêu cầu bên trên
2. Biểu diễn bằng đồ thị độ chính xác và sai số của mô hình mạng nơron bên trên
3. Dùng hình biên báo bên ngoài tập dữ liệu cho sẵn bên trên (tự chụp hoặc lấy ảnh có sẵn trên mạng), đưa vào nhận dạng và hiển thị kết quả ra màn hình (có thể thao khảo source code các bài tập ví dụ trên lớp để input ngõ vào và output kết quả nhận dạng)