**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙢🕮🙠---



**BÁO CÁO**

**ĐỒ ÁN**

**LẬP TRÌNH PYTHON**

**XÂY DỰNG NHẬN DIỆN BIỂN BÁO**

**GIAO THÔNG BẰNG YOLOV5**

**SVTH** : Võ Tuấn Anh MSSV: 61130038

Lớp : 61.CNTT-CLC

GVHD : ThS. Nguyễn Hải Triều

*Khánh Hòa, tháng 4 năm 2022*

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN 3](#_Toc106121850)

[1.1 Giới thiệu chung 3](#_Toc106121851)

[1.2 Mục tiêu đề tài 3](#_Toc106121852)

[Chương 2. CƠ SỞ LÝ THUYẾT 4](#_Toc106121853)

[2.1 Sơ lược về YOLO 4](#_Toc106121854)

[2.2 Yêu cầu mạng YOLO 4](#_Toc106121855)

[2.3 Kiến trúc mạng YOLO 4](#_Toc106121856)

[Chương 3. CHUẨN BỊ DỮ LIỆU 4](#_Toc106121857)

[3.1 Tập hợp ảnh 5](#_Toc106121858)

[3.2 Gán nhãn 5](#_Toc106121859)

[3.3 Sắp xếp thư mục ảnh và nhãn 7](#_Toc106121860)

[Chương 4. HUẤN LUYỆN MÔ HÌNH 9](#_Toc106121861)

[4.1 Công nghệ sử dụng và thư viện 9](#_Toc106121862)

[4.2 Huấn luyện mô hình 9](#_Toc106121863)

[4.3 Kết quả đạt được 9](#_Toc106121864)

[Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 10](#_Toc106121865)

[5.1 Kết Luận 10](#_Toc106121866)

[5.1.1 Ưu điểm 10](#_Toc106121867)

[5.1.2 Nhược điểm 10](#_Toc106121868)

[5.2 Hướng Phát Triển 10](#_Toc106121869)

CHƯƠNG 1. TỔNG QUAN

* 1. Giới thiệu chung

Với sự phát triển hết sức mạnh mẽ của công nghệ 4.0, AI (trí tuệ nhân tạo) hay cụ thể là Machine Learning (học máy) cũng đang từng bước phát triển không ngừng. Các ứng dụng về lĩnh vực này ngày càng được áp dụng nhiều vào đời sống con người như: Robot giúp việc, hệ thống dịch thuật, chatbox…

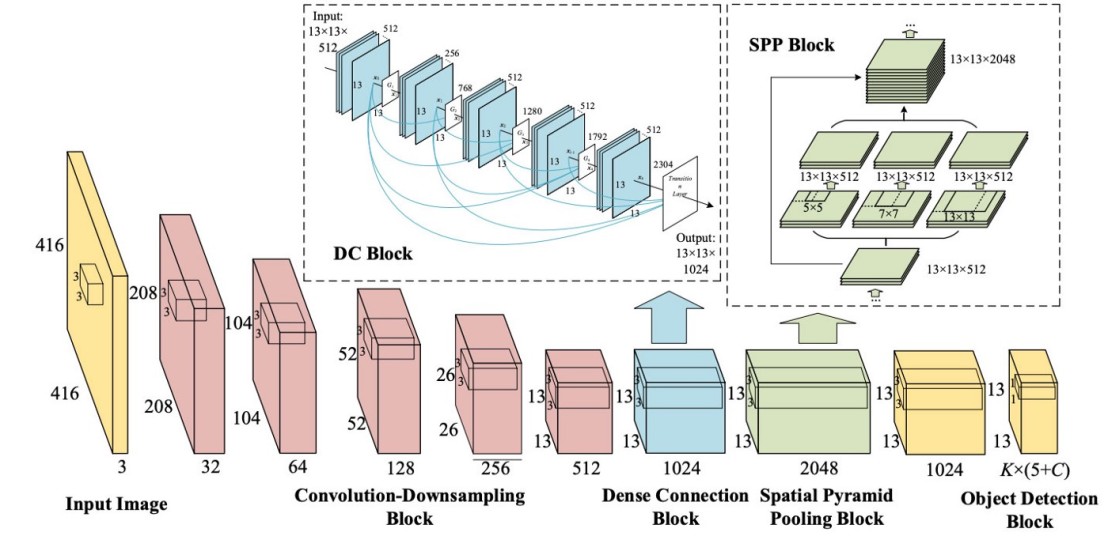
Mạng Nowrowrron học sâu (Deep Learning Network) là lĩnh vực nghiên cứu các thuật toán, chương trình máy tính để máy tính có thể học tập và đưa ra những dự đoán như con người. Nó được ứng dụng vào nhiều ứng dụng về phân loại và phát hiện đối tượng. Một ví dụ điển hình là CNN (Convolutional Neural Network) áp dụng để nhận dạng tự động, tìm hiểu các mẫu phân biệt từ ảnh bằng cách xếp chồng liên tiếp các lớp lên nhau và trong nhiều ứng dụng, CNN hiện nay được coi là trình phân loại ảnh mạnh và thúc đẩy các công nghệ trong lĩnh vực thị giác máy tính, làm đòn bẩy cho quá trình học máy. Nhưng bên cạnh đó, để phân loại được một đối tượng thì công nghệ CNN tiêu tốn cực lớn về tài nguyên như băng thông, bộ nhớ và khả năng xử lý của phần cứng

Để giảm thiểu những tài nguyên tiêu hao này, những thuật toán mô hình giải thuật theo thời gian được ra đời ngày càng nhiều và trong đó mô hình YOLOv5 cho bài toán phân loại vật thể, cụ thể là được ứng dụng vào đề tài “ Phân loại biển báo giao thông Việt Nam”.

* 1. Mục tiêu đề tài
* Tìm hiểu về Deep Learning và cá ứng dụng
* Hiểu rõ được cơ sở lý thuyết, kiến trúc của mô hình YOLO cho bài toán nhận diện vật thể.
* Sử dụng các thư viện hỗ trợ, môi trường ảo đẻ thực thi mô hình

1. CƠ SỞ LÝ THUYẾT
   1. Sơ lược về YOLO

YOLOv5 là một mô hình Object Detection thuộc họ mô hình YOLO. Nếu các bạn chưa biết thì 3 phiên bản YOLO đầu tiên được phát triển bởi Joseph Redmon. Sau đó, Alexey Bochkovskiy cho ra mắt YOLOv4 với sự cải thiện cả về tốc độ cũng như độ chính xác. Và rồi YOLOv5 được công bố gần đây với những so sánh ban đầu cho thấy độ chính xác tương đương YOLOv4 và có tốc độ nhanh hơn khi thực hiện dự đoán (tuy nhiên vẫn có rất nhiều hoài nghi về độ tin cậy của những so sánh này vì YOLOv5 mới được ra mắt trên GitHub chứ chưa có bài báo chính thức nào cả).

* 1. Yêu cầu mạng YOLO ?
* Kết hợp với Maxpooling để trích xuất dặc trưng của ảnh tốt hơn
* Khung hình bao quanh vật thể, khung hình có kích thước xác định trước
* Một khối output mà ta sẽ chia nó thành một lưới ô vuông, phát hiện trên từng cell.
* Phương pháp giúp tối thiểu nhiều bounding box overlap
  1. Kiến trúc mạng YOLO

Mô hình mạng YOLO

Kiến trúc YOLO bao gồm: base network là các mạng convolution làm nhiệm vụ trích xuất đặt trưng. Phần phía sau là những Extra Layer được áp dụng để phát hiện vật thể trên feature map của base network.

1. CHUẨN BỊ DỮ LIỆU
   1. Tập hợp ảnh

* Trước hết chúng ta cần phải có dữ liệu ảnh và nhãn (images & labels). Dữ liệu đối với mô hình cũng giống những viên gạch cho việc xây nhà, không có dữ liệu thì không xây dựng được mô hình nào cả.
* Có vài cách để thu thập ảnh cho việc huấn luyện mô hình:
* Tìm trên Open Images Dataset, đây là tập hợp hơn 9 triệu ảnh với 6000 nhóm khác nhau.
* Tìm với công cụ tìm kiếm (Google, Bing, …) với lưu ý cần kiểm tra kỹ về bản quyền sử dụng ảnh.
* Trích xuất các khung hình từ Videos
* Tự chụp ảnh
* Trong bài nhận diện này em chủ yếu ảnh thực tế từ Google.
  1. Gán nhãn

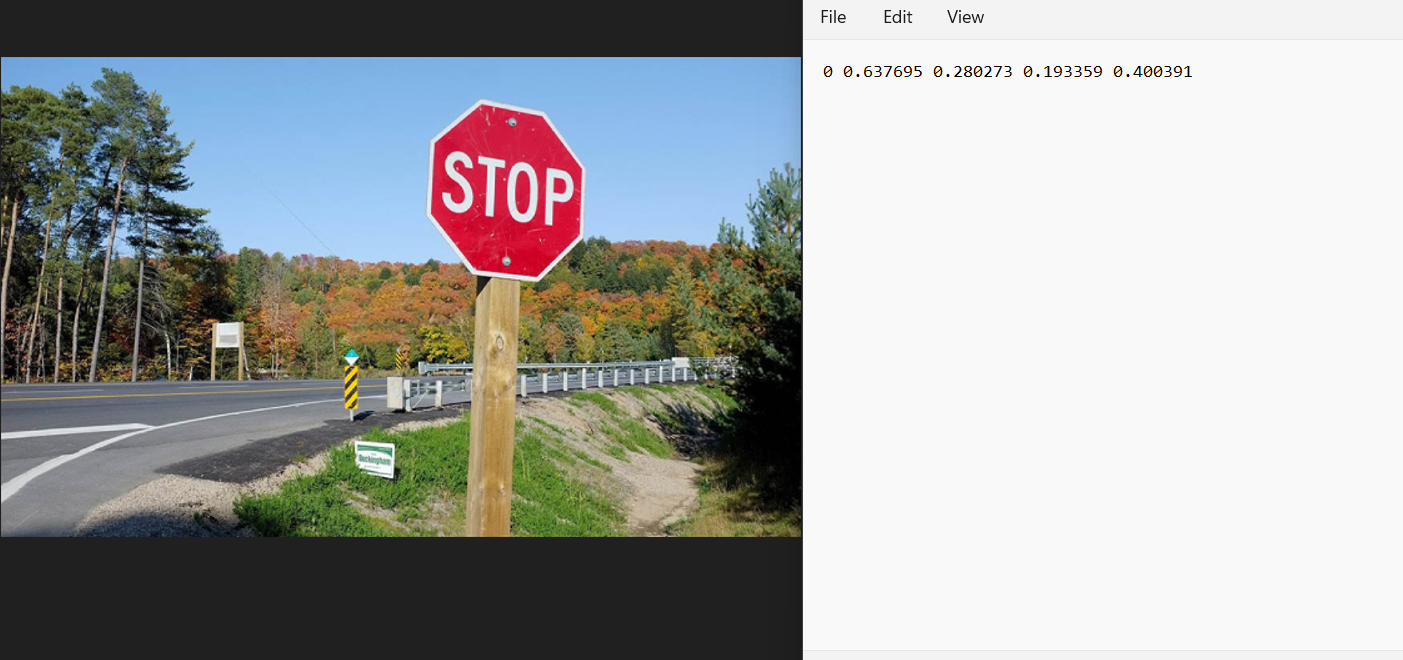
Trong bài này em dùng phần mềm Makesense để gán nhãn các ảnh.



Phần mềm Makesense

Mỗi ảnh sẽ có một nhãn (file txt) tương ứng để chứa thông tin về đường bao của đối tượng:

* Mỗi dòng chứa thông tin của một đối tượng
* Mỗi dòng có 5 giá trị chứa thông tin về đối tượng và toạ độ đường bao: đối tượng, toạ độ trung tâm x, toạ độ trung tâm y, chiều rộng, chiều dài.
* Lưu ý là tọa độ đường bao đã được chuẩn hóa về khoảng (0,1) đây là yêu cầu của mô hình YOLO



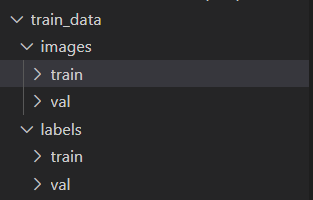
Mô tả

Tọa độ đường bao YOLO

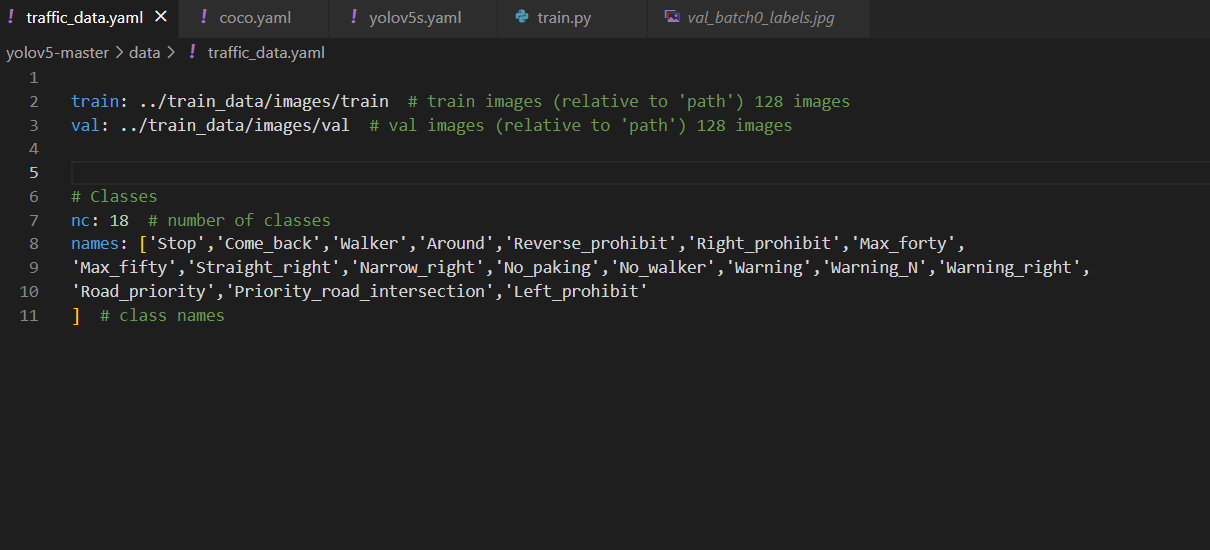
* 1. Sắp xếp thư mục ảnh và nhãn

Sau khi đã có ảnh và nhãn, chúng ta có thể sắp xếp thư mục như sau:

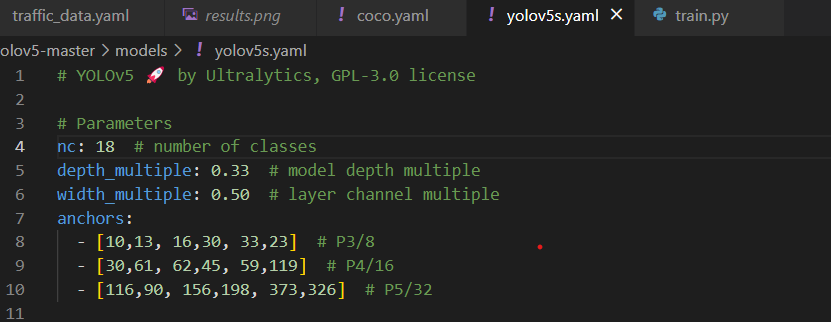
* Thư mục images và labels chứa ảnh và nhãn tương ứng
* Ảnh và nhãn được chia thành hai thư mục train và validation dùng để huấn luyện và kiểm định mô hình.



* File cấu hình của tập dữ liệu traffic\_data.yaml chứa đường dẫn đến thư mục ảnh và thông tin đối tượng.



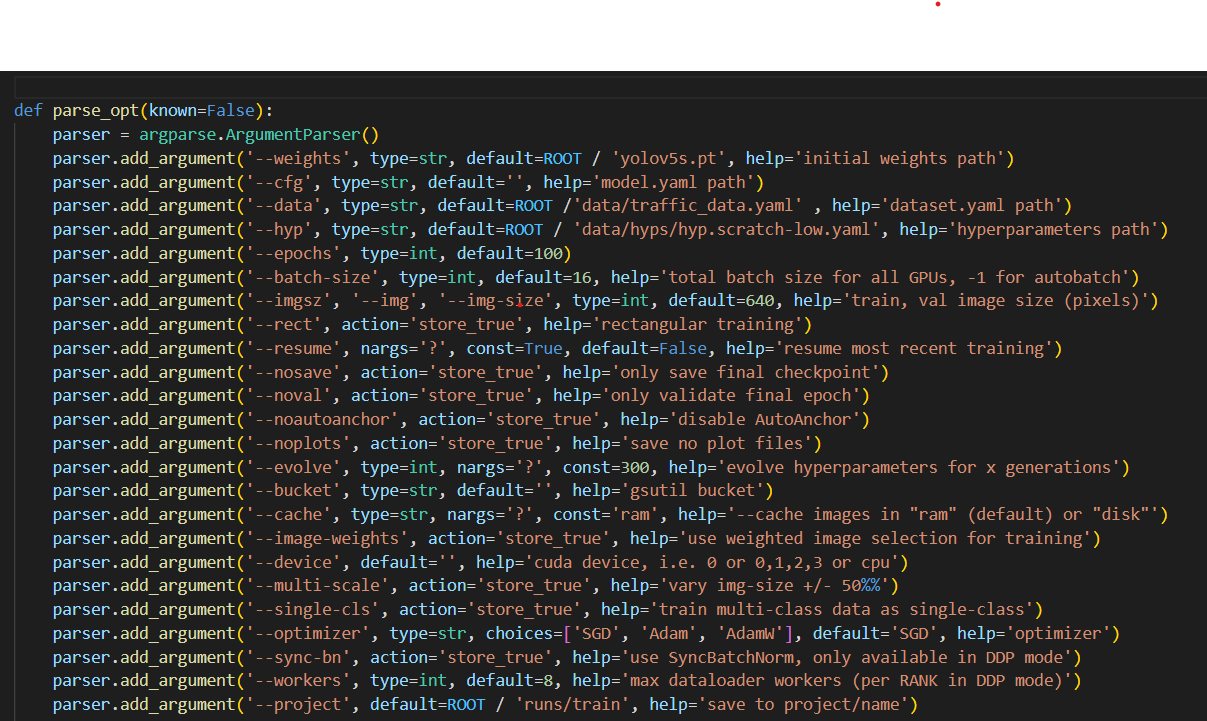
* File cấu hình model/yolov5s.yaml chứa thông tin cấu trúc mạng, ở đây chúng ta dùng cấu trúc mô hình YOLOv5s (cỡ nhỏ) với mục tiêu rút ngắn thời gian huấn luyện và kích thước mô hình.



1. HUẤN LUYỆN MÔ HÌNH
   1. Công nghệ sử dụng và thư viện

Sử dụng VSCode , python version(310) và import các thư viện liên quan.

* 1. Huấn luyện mô hình
* Img: kích thước ảnh(độ phân giải)
* Batch size: số ảnh dùng để huấn luyện trong mỗi lượt
* Epochs : số lượt huấn luyện cho tất cả các ảnh trong tập dữ liệu
* Cfg : đường dẫn đến file cấu hình mô hình
* Cache : dùng bộ nhớ đệm nhanh hơn
* Weights : đường dẫn đến file weight chứa độ liên kết các neuron( để ‘’ là huấn luyện từ đầu)
* Workers : số lõi CPU sử dụng



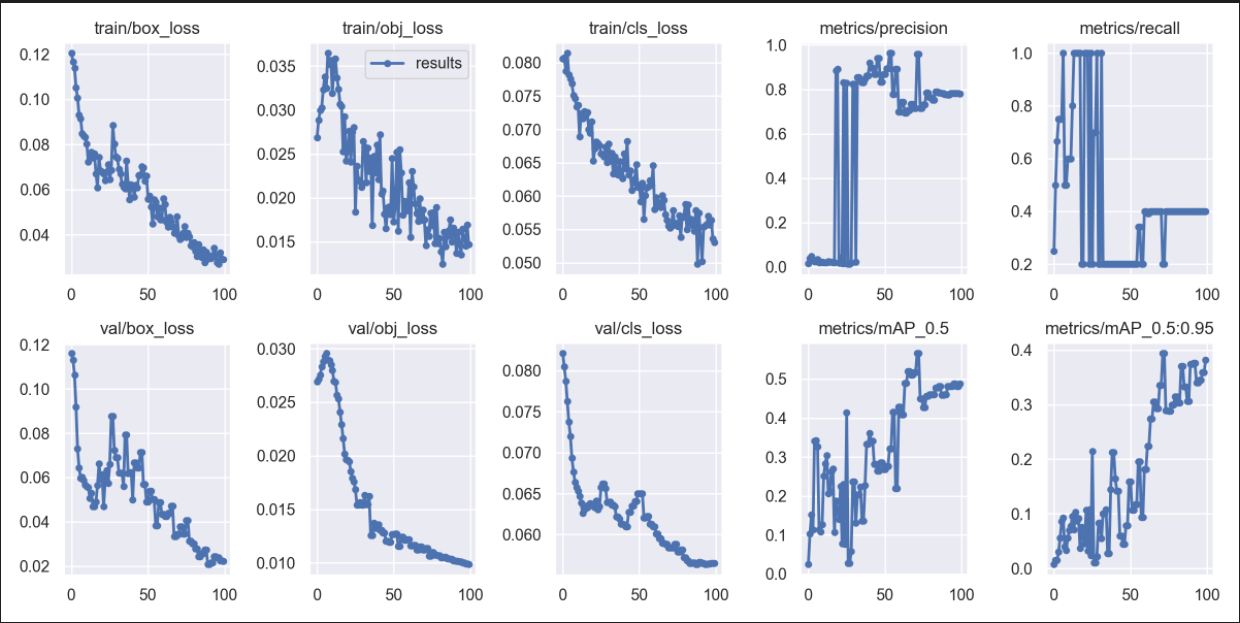
* 1. Kết quả đạt được
* Gán nhãn cho dữ liệu đầu vào



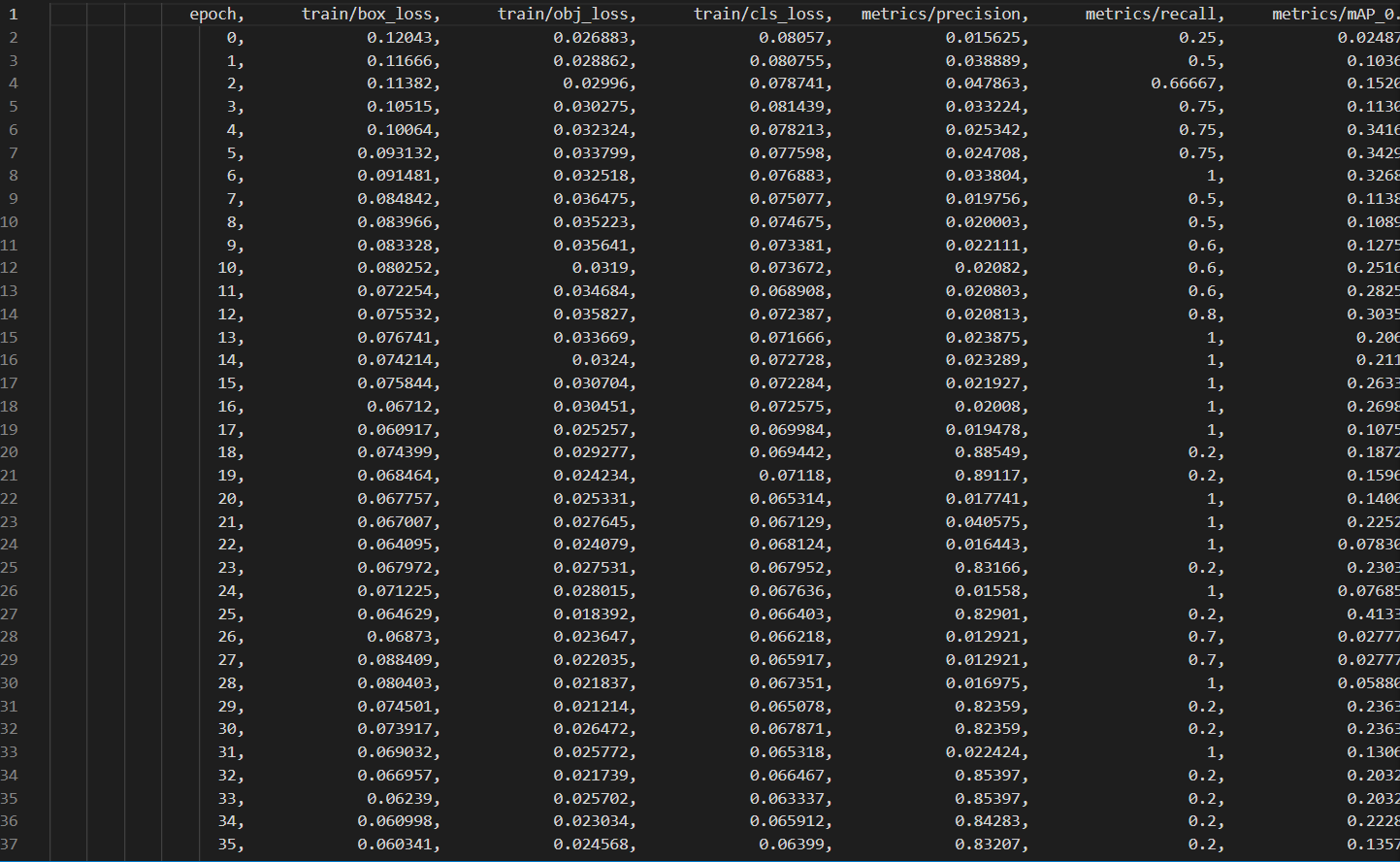
* Nhận diện các class dữ liệu đầu vào



* Sơ đồ kết quả đạt được



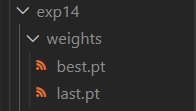
* Kết quả các train từng epoch được lưu lại ở file result.csv



* Kết quả đạt được : Vì train trên VS code cấu hình máy yếu nên chưa nhận diện được tối ưu kết quả.



* Sau khi train xong trong mục weights có 2 file là best.pt và last.pt. Đó là weights tốt nhất epoch và best.pt và cuối cùng last.pt



* Để phát hiện đối tượng cụ thể ta có thể nhận diện bằng video. Camera, file ảnh..

1. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN
   1. Kết Luận
      1. Ưu điểm

* Nhận diện tương đối các biển báo
* Nắm được các bước huấn luyện một mô hình nhận diện
  + 1. Nhược điểm
* Chưa đạt được kết quả như mong muốn.
* Chưa đạt hiệu quả nhận diện tối ưu.
* Tốc độ còn chậm, cấu hình máy thấp.
* Dữ liệu còn ít, chưa thiết kế giao diện.
  1. Hướng Phát Triển
* Xây dựng giao diện hoàn thiện, thêm các chức năng liên quan như đếm số lượng biển báo.
* Tối ưu hóa về hiệu suất nhận diện.
* Thêm bộ dữ liệu lớn hơn hoàn thiện để thành dự án lớn chạy thực tế.

**LỜI CẢM ƠN**

Trước tiên em xin dành lời cảm ơn chân thành đến Thầy Cô khoa Công Nghệ Thông Tin đã truyền đạt cung cấp cho em những kiến thức cần thiết để vận dụng trong thực tiễn cũng như đề tài này.

Và đặc biệt cảm ơn Thầy Nguyễn Hải Triều và các bạn đã tận tình hướng dẫn, giúp đỡ cho em trong suốt quá trình thực hiện. Do kiến thức còn hạn chế trong quá trình thực hiện em không tránh những sai sót. Kính mong thầy có thể chỉ dẫn thêm để rút kinh nghiệm và hoàn thành tốt hơn ở những bài tập lớn tiếp theo.

Em xin chân thành cảm ơn !