**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN**





**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**MÔN KỸ THUẬT LẬP TRÌNH PYTHON**

**LỚP: IE221.L21.CNCL**

**ỨNG DỤNG LIÊN QUAN ĐẾN XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH CHO BÀI TOÁN PHÁT HIỆN PHƯƠNG TIỆN GIAO THÔNG Ở VIỆT NAM**

**GVHD:** ThS. Nguyễn Thanh Sơn

**Sinh viên thực hiện:**

NGÔ ĐÌNH LUÂN MSSV: 18521081

**Tp. Hồ Chí Minh, 5/2021**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2021*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

# BẢNG QUÁ TRÌNH LÀM VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | MSSV | Phân công | Thực hiện |
| Ngô Đình Luân | 18521064 | * Tìm hiểu gán nhãn dữ liệu * Chạy mô hình trên Collab * Chạy mô hình trên local * Demo Web * Viết báo cáo | Hoàn thành |

**LỜI MỞ ĐẦU**

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Thanh Sơn đã có những bài giảng, hướng dẫn về lý thuyết lẫn thực hành nhằm giúp cho em có những kiến thức cần thiết để thực hiện đồ án này.

Trong thời gian gần một tháng thực hiện đồ án, em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, em vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm chúng em không tránh khỏi những thiếu sót. Vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía Thầy nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện tiếp các đề tài khác trong tương lai.

Xin chân thành cảm ơn Thầy!

# MỤC LỤC

[● BẢNG QUÁ TRÌNH LÀM VIỆC 3](#_Toc72950421)

[● MỤC LỤC 5](#_Toc72950422)

[- TỔNG QUAN 5](#_Toc72950423)

[o I. GIỚI THIỆU ĐỀ TÀI 5](#_Toc72950424)

[▪ **1. Tổng quan** 5](#_Toc72950425)

[▪ 2. Giới thiệu bài toán 6](#_Toc72950426)

[o II. CƠ SỞ LÝ THUYẾT 8](#_Toc72950428)

[▪ **1. Machine learning và Deep learning** 8](#_Toc72950429)

[▪ **2. Object Detection** 9](#_Toc72950430)

[▪ **3. Phân loại các bài toán Object Detection** 10](#_Toc72950431)

[▪ **4. Mạng Neural Network** 11](#_Toc72950432)

[▪ **5. Mạng Neural Network** 12](#_Toc72950433)

[▪ **6. Mạng Neural Network** 13](#_Toc72950434)

[▪ **7. YOLO** 13](#_Toc72950435)

[o III. Quá trình thực hiện 15](#_Toc72950436)

[● TỔNG KẾT 23](#_Toc72950437)

# TỔNG QUAN

## I. GIỚI THIỆU ĐỀ TÀI

### **1. Tổng quan**

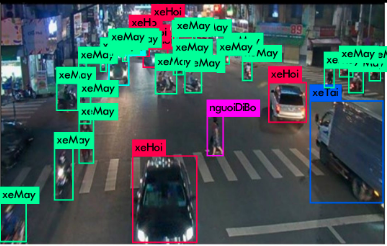
* Cùng với sự phát triển của giao thông thông minh, bài toán nhận diện phương tiện trên các nút giao lộ đông đúc từ dữ liệu thu thập bởi camera giao thông trở thành một vấn đề quan trọng, tiềm năng dễ thấy nhất của việc này có thể kể đến đó là giúp các chính quyền và người dân có thể nắm bắt được tình hình giao thông của thành phố theo thời gian thực, từ đó đưa ra những phương án để giảm thiểu ùn tắc và nâng cao hiệu quả của đèn giao thông.
* Trong những năm gần đây, với sự phát triển nhanh chóng về số lượng cùng sự đa dạng về phương tiện giao thông ở Việt Nam, các tuyến đường huyết mạch với mật độ giao thông lớn và các nút giao đông đúc dần xuất hiện tình trạng kẹt xe, điều này làm ảnh hưởng rất lớn đến môi trường sống, sự sinh hoạt - làm việc bình thường của người dân, góp phần làm chậm sự phát triển kinh tế - xã hội kéo theo rất nhiều hệ lụy sau nó. Ngoài những lý do khách quan liên quan đến cơ sở hạ tầng đường xá và sự gia tăng quá nhanh số lượng phương tiện giao thông thì còn có một lý do khác không kém phần quan trọng đó là ý thức chấp hành Luật Giao thông của người điều khiển phương tiện. Tất cả những lý do kể trên khiến cho việc điều tiết phương tiện tại các nút giao phải đối mặt với những khó khăn lớn.
* Với sự phát triển của Trí tuệ Nhân tạo, Máy học, Học sâu và Dữ liệu lớn, cùng với sự ra đời của Internet Kết nối Vạn vật (Internet of Things) và mạng 5G, giới nghiên cứu đã chứng kiến sự tiến bộ đáng kể trong các phương pháp giải quyết các bài toán khó đến từ các lĩnh vực quan trọng như Thị giác Máy tính, Xử lý Ngôn ngữ Tự nhiên, Xử lý Giọng nói,...
* Bài toán nhận diện xe trong quang cảnh giao thông thuộc lớp các bài toán về Giao thông Thông minh, một trong những nhánh nghiên cứu quan trọng trong lĩnh vực Thị giác Máy tính và nhận được sự quan tâm đặc biệt từ giới nghiên cứu, vì nó được kỳ vọng sẽ giảm thiểu tắc nghẽn giao thông và nâng cao hiệu quả của đèn tín hiệu thông qua việc tự động thống kê và cập nhật tình hình giao thông tại các tuyến đường trọng điểm, bên cạnh đó còn cung cấp cái nhìn chi tiết về hành vi lái xe của người dân. Giúp chính quyền có những phương án xử lý hiệu quả hơn trong quá trình giải quyết vấn nạn khó khăn này.

### **2. Giới thiệu bài toán**

* Bài toán mà em hướng đến là ứng dụng xử lí ảnh và thị giác máy tính cho bài toán phát hiện phương tiện giao thông ở Việt Nam. Đây là chủ đề của Hội thi giải pháp ứng dụng Trí tuệ Nhân tạo (AI) trên địa bàn thành phố Hồ Chí Minh năm 2020 - HCM AI CHALLENGE 2020 với chủ đề “Trí tuệ nhân tạo với giao thông thông minh".

|  |  |
| --- | --- |
| **Loại** | **Phương tiện** |
| 1 | Người đi bộ |
| 2 | Xe đạp |
| 3 | Xe máy |
| 4 | Xe hàng rong |
| 5 | Xe ba gác |
| 6 | Xe Taxi |
| 7 | Xe hơi |
| 8 | Xe bán tải |
| 9 | Xe cứu thương |
| 10 | Xe khách |
| 11 | Xe bus |
| 12 | Xe tải |
| 13 | Xe container |
| 14 | Xe cứu hỏa |

Bảng phân loại các đối tượng



Các phương tiện giao thông được nhận dạng và phân loại

* Mỗi video giao thông được ghi nhận tại một camera giao thông cụ thể. Trong mỗi video giao thông, em sẽ nhận diện các vật thể như xe hay người đi bộ và upload lên web demo.
* Bài toán với input đầu vào là một bức ảnh hoặc một video ghi lại hình cảnh các phương tiện giao thông đang di chuyển và output đầu ra sẽ là hình ảnh các phương tiện được mô hình Yolov3 phát hiện và đánh bounding box.



Web Demo

* **3. Thách thức của bài toán**
* Như đã đề cập ở trên, bài toán nhận diện phương tiện giao thông là một bài toán mang tính ứng dụng cao, được giới khoa học trên thế giới quan tâm nghiên cứu. Nhờ sự phát triển vượt bậc của Học sâu, các phương pháp mới ra đời dành cho bài toán Phát hiện Đối tượng ngày càng phá vỡ được nhiều kỷ lục về độ chính xác và tính hiệu quả.
* Tuy nhiên, những kết quả này đều là những kết quả đạt được từ việc xử lý những video giao thông ở Mỹ có chất lượng cao và tình huống giao thông không mấy phức tạp Ngược lại với đó, giao thông ở Việt Nam là một câu chuyện hoàn toàn khác. Và em sau khi thực hiện mô hình thì thấy có những thách thức như sau:

• Các phương tiện che khuất nhau vẫn bị bỏ sót khi giao thông đông đúc

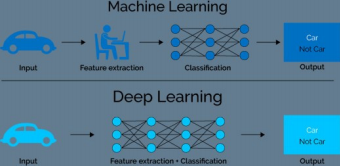
• Chất lượng ảnh thấp khiến quá trình Phát hiện gặp không ít khó khăn

### 

## II. CƠ SỞ LÝ THUYẾT

### **1. Machine learning và Deep learning**

* Machine Learning là một thuật ngữ rộng để chỉ hành động bạn dạy máy tính cải thiện một nhiệm vụ mà nó đang thực hiện. Cụ thể hơn, Machine Learning đề cập tới bất kỳ hệ thống mà hiệu suất của máy tính khi thực hiện một nhiệm vụ sẽ trở nên tốt hơn sau khi hoàn thành nhiệm vụ đó nhiều lần. Hay nói cách khác, khả năng cơ bản nhất của Machine Learning là sử dụng thuật toán để phân tích những thông tin có sẵn, học hỏi từ nó rồi đưa ra quyết định hoặc dự đoán về một thứ gì đó có liên quan. Thay vì tạo ra một phần mềm với những hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ cụ thể, máy tính được “huấn luyện” bằng cách sử dụng lượng dữ liệu và các thuật toán để học cách thực hiện nhiệm vụ. Nói tóm lại, Machine Learning là quá trình dạy máy tính thực hiện một nhiệm vụ, thay vì lập trình nó làm thế nào để thực hiện nhiệm vụ đó từng bước một.
* Deep Learning: là một kỹ thuật của Machine Learning lấy cảm hứng từ cấu trúc bộ não con người, có thể coi Deep Learning là một loại Machine Learning với "mạng thần kinh - neural networks" sâu có thể xử lý dữ liệu theo cách tương tự như một bộ não con người có thể thực hiện. Điểm khác biệt chính là con người không cần phải dạy một chương trình Deep Learning thực hiện một nhiệm vụ, mà chỉ cần cung cấp đủ dữ liệu đầu vào và nó sẽ tự mình hình dung, tự học cách thực hiện nhiệm vụ. Nói tóm lại, Deep Learning là loại Machine Learning mà trong đó máy tự đào tạo chính nó. Những nhân tố góp phần giúp thúc đẩy sự bùng nổ các phương pháp học sâu và giúp chúng trở thành xu hướng mới của thời đại:



Machine Learning và Deep Learning

• Sự ra đời của các bộ dữ liệu lớn được gán nhãn.

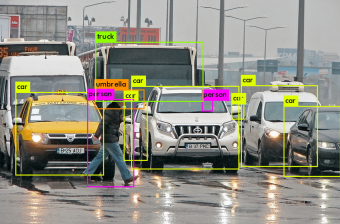
• Sự cải tiến của các kiến trúc: GoogLeNet, VGG, ResNet, . . . và các kỹ thuật transfer learning, fine tuning.

• Nhiều thư viện mới hỗ trợ việc huấn luyện deep network với GPU: theano, caffe, mxnet, tensorflow, pytorch, keras, . . .

• Khả năng tính toán song song tốc độ cao của GPU.

### **2. Object Detection**

* Object detection là một công nghệ máy tính liên quan đến thị giác máy tính và xử lý hình ảnh. Object Detection đề cập đến khả năng của hệ thống máy tính và phần mềm để định vị các đối tượng trong một hình ảnh và xác định từng đối tượng. Object Detection đã được sử dụng rộng rãi để phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật và xe không người lái. Object Detection có lẽ là khía cạnh sâu sắc nhất của thị giác máy do số lần sử dụng trong thực tế.



Object Detection

### **3. Phân loại các bài toán Object Detection**

* Lịch sử hình thành, phát triển và đặc điểm cấu trúc của các thuật toán object detection bao gồm 2 nhóm chính:

• One-stage: bao gồm họ các mô hình YOLO (You Only Look Once), SSD (Sin gle Shot MultiBox Detector). Là một nhóm kỹ thuật để nhận dạng đối tượng được thiết kế để nhận diện vật thể real time.

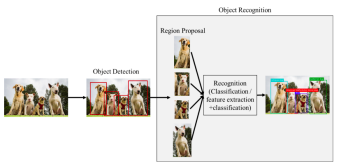
• Two-stage: gồm họ các mô hình R-CNN (Region-Based Convolutional Neu ral Networks) hoặc Mask R-CNN dùng để giải quyết các nhiệm vụ định vị vật thể và nhận diện vật thể.

* One-stage: Coi việc phát hiện đối tượng như một bài toán regression bằng cách lấy hình ảnh đầu vào và hiển thị xác suất lớp và tọa độ bounding box. Các mô hình như vậy đạt tỷ lệ chính xác thấp hơn, nhưng nhanh hơn nhiều so với các phương pháp sử dụng two-stage.



Ví dụ về One-stage trong Object Detection

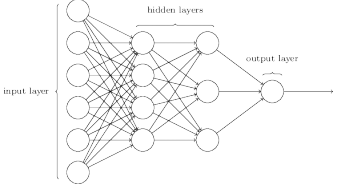
* Two-stage: Sử dụng Region Proposal Network để tạo ra các khu vực đề xuất trong giai đoạn one-stage đầu tiên. Sau đó gửi các đề xuất liên kết xuống để model thực hiện việc phân loại và phát hiện vị trí. Các mô hình như vậy đạt đến độ chính xác cao nhất nhưng thường chậm hơn.



ví dụ về Two-stage trong Object Detection

### **4. Mạng Neural Network**

* Mạng Neural Network bao gồm nhiều lớp (layer) - là tập hợp các Neuron với các kết nối giữa các lớp (layer) khác nhau. Các lớp này biến đổi dữ liệu bằng cách tính toán tổng trọng số của các đầu vào và sau đó chuẩn hóa nó bằng cách sử dụng các hàm kích hoạt được gán cho các Neuron. Layer được chia thành 3 nhóm chính dựa trên hình bên dưới, với lớp ngoài cùng bên trái được gọi là lớp đầu vào (input layer), lớp ngoài cùng bên phải được gọi là lớp đầu ra (output layer), và nhóm các layer ở giữa gọi là các lớp ẩn (hidden layers). Bất kỳ Mạng Neural nào cũng có 1 lớp đầu vào và 1 lớp đầu ra. Tuy nhiên, số lượng lớp ẩn khác nhau giữa các mạng khác nhau tùy thuộc vào mức độ phức tạp của vấn đề. Những Mạng Neural có từ 2 lớp ẩn trở lên được gọi là Deep Neural Network.



Neuron sinh học

### **5. Mạng Neural Network**

* Lớp tích chập (Conv-layer) đảm nhận nhiệm vụ chính trong tiến trình rút trích đặc trưng của dữ liệu đầu vào trong quá trình học. Về căn bản, lớp này thực hiện thao tác tích chập trên dữ liệu, sau đó lan truyền thông tin đầu ra cho lớp tiếp theo mà nó liên kết. Tuy nhiên, không giống với Fully Connected Layer sử dụng toàn bộ thông tin dữ liệu, đối với dữ liệu ảnh thì nó giữ thông tin kết nối cục bộ của các điểm ảnh, vì chúng chứa đặc trưng cục bộ. Nguyên lý hoạt động của Conv-layer dựa trên ý tưởng bộ não con người ghi nhớ một bức tranh, ghi nhớ đặc điểm nổi bật của bức tranh.
* Đối với Conv-layer, để xây dựng được kiến trúc của nó và đảm bảo thực thi chức năng tích chập hiệu quả, chúng ta cần chú trọng ba siêu tham số chính:

• Depth: Chiều sâu của kích thước đầu ra phụ thuộc vào số lượng bộ lọc được áp dụng trên ảnh. Mỗi bộ lọc sẽ có cách đánh giá khác nhau về ảnh đầu vào và rút trích ra các đặc trưng khác nhau.

• Stride: Biểu thị số bước nhảy mà bộ lọc dịch chuyển. Ví dụ: Nếu stride bằng 3, thì mỗi bước nhảy của bộ lọc trên đối tượng là 3 pixel. Vai trò chính của thông số này là giảm kích thước đầu ra.

• Zero-padding: Được biết đến trong tác vụ thêm giá trị 0 vào viền ảnh của dữ liệu đầu vào. Điều này có tác dụng duy trì kích thước dữ liệu tương thích với các bộ lọc được áp dụng trong kiến trúc mạng.

### **6. Mạng Neural Network**

* Đây là lớp chịu trách nhiệm lấy mẫu kích thước dữ liệu đầu vào với độ phi tuyến cho phép quá trình rút trích đặc trưng và học dữ liệu tốt hơn. Lợi ích của việc sử dụng Pooling Layer là có thể cắt giảm kích thức đầu ra về chiều rộng và chiều cao (không cải thiện chiều sâu) từ đó góp phần cắt giảm chi phí tính toán. Ngoài ra lớp này còn có khả năng chuẩn hóa đầu ra, hạn chế tình trạng over-fitting khi tinh chỉnh kích thước đầu ra.
* Có nhiều biến thể của Pooling Layer: Max Pooling, L2-Pooling, Average Pooling... Trong đó, Max Pooling với bộ lọc 2x2 và stride 2 được sử dụng phổ biến trong kiến trúc CNN vì giúp loại bỏ 75% tác vụ xử lý dữ liệu đầu vào.

### **7. YOLO**

* YOLO (You Look Only Once) là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các Convolutional layers và Connected Layers, đây được xem là một trong những thuật toán object detection nhanh nhất hiện nay. Mặc dù không phải là phương pháp có độ chính xác cao nhất, tuy nhiên YOLO vẫn được ứng dụng rất nhiều trong những dự án thực tế khi mà độ chính xác không phải là ưu tiên hàng đầu.
* Dù đều được gọi là YOLO, mô hình này thường được biết đến với tên gọi gồm “YOLO” đi kèm theo thứ tự các phiên bản cải tiến của chính nó (YOLOv1, YOLOv2, YOLOv3, YOLOv4). Ở đây em sử dụng phiên bản YOLOv3 để nhận diện đối tượng trong video.
* YOLOv3 có kiến trúc khá giống YOLOv2 nhưng được thêm các cải tiến mới trong các nghiên cứu vào YOLOv2, để cuối cùng tạo ra YOLOv3. Các cải tiến đó bao gồm:

• Logistic regression cho confidence score: YOLOv3 predict độ tự tin của bounding box (có chứa vật hay không) sử dụng Logistic Regression

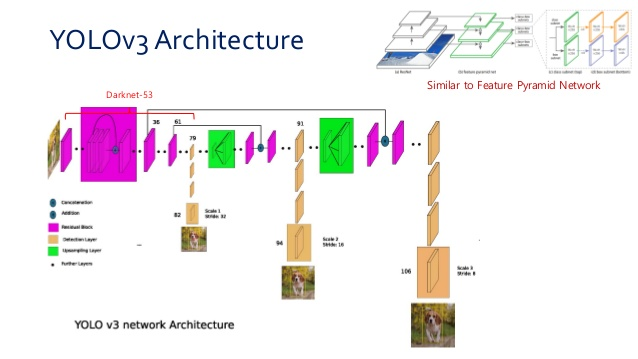
• Thay softmax bằng các logistic classifier rời rạc: YOLOv3 sử dụng các logis tic classifier thay vì softmax cho việc classify đối tượng. Việc này cho hiệu quả tốt hơn nếu các label không "mutually exclusive", tức là có thể có đối tượng cùng thuộc 2 hay nhiều class khác nhau.

• Backbone mới - Darknet-53: Backbone được thiết kế lại với việc thêm các residual blocks (kiến trúc sử dụng trong ResNet).

• Multi-scale prediction: YOLOv3 sử dụng kiến trúc Feature Pyramid Net works (FPN) để đưa ra các dự đoán từ nhiều scale khác nhau của feature map. Việc này giúp YOLOv3 tận dụng các feature map với độ thô - tinh khác nhau cho việc dự đoán.

• Skip-layer concatenation: YOLOv3 cũng thêm các liên kết giữa các lớp dự đoán. Mô hình upsample các lớp dự đoán ở các tầng sau và sau đó concate nate với các lớp dự đoán ở các tầng trước đó. Phương pháp này giúp tăng độ chính xác khi predict các object nhỏ.

• Đối với kích thước của ảnh, YOLOv3 tiến hành dự đoán tại 3 scales (với kích thước của từng scales tương ứng được tính toán dựa trên việc giảm số chiều của dữ liệu đầu vào qua phép chia cho 32, 16, và 8). Việc này giúp giải quyết vấn đề trong việc nhận diện các vật thể có kích thước nhỏ, điều mà YOLOv2 chưa thể làm tốt. Ngoài ra, YOLOv3 cũng thêm các liên kết giữa các lớp dự đoán. Mô hình upsample các lớp dự đoán ở các tầng sau và concatenate với các lớp dự đoán ở các tầng trước đó giúp bảo tồn các tính năng chi tiết và giúp phát hiện các vật thể nhỏ tốt hơn.



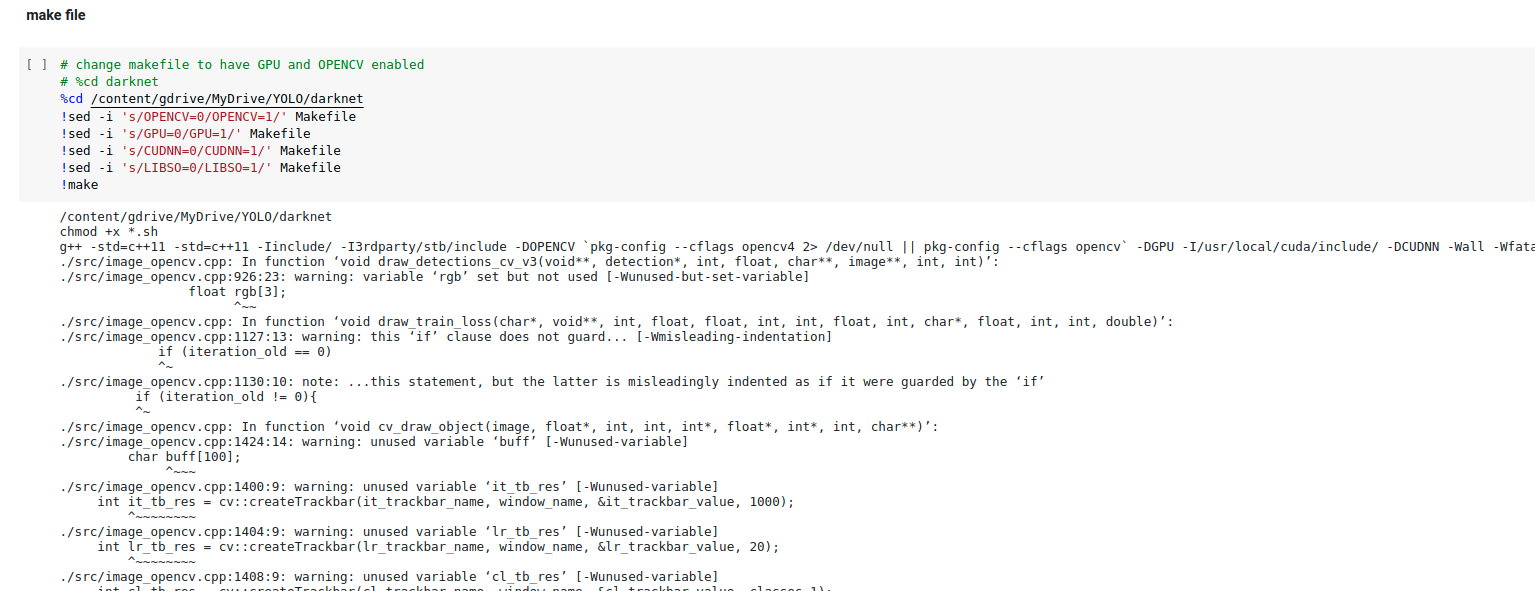
Kiến trúc của YOLOv3

## III. Quá trình thực hiện

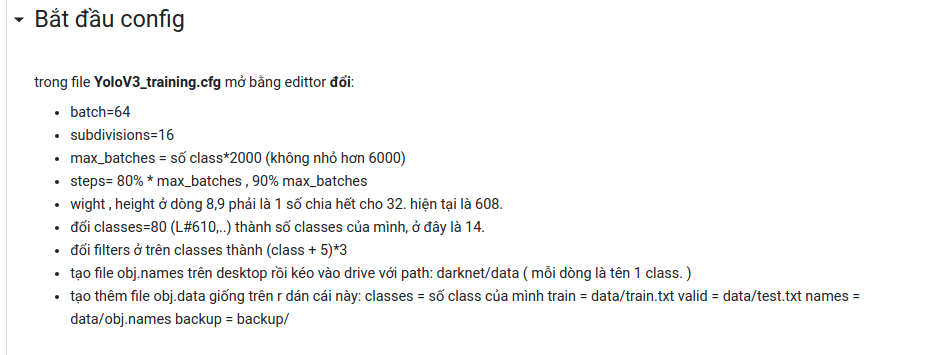
* 1. ***Thu thập và tiền xử lí dữ liệu***
* Hình ảnh được thu thập tại một số camera từ website giao thông thành phố: http://giaothong.hochiminhcity.gov.vn/
* Sử dụng tool labelImg trên github: https://github.com/tzutalin/labelImg Bộ dữ liệu được gãn nhãn gồm 14 đối tượng:
* Người đi bộ (class: 0); Xe đạp (class: 1); Xe máy (class: 2); Xe hàng rong (class: 3); Xe ba gác (4); Taxi (class: 5); Xe hơi (class: 6); Xe bán tải (class: 7); Xe cứu thương (class: 8); Xe khách (class: 9); Xe buýt (class: 10); Xe tải (class: 11); Xe container (class: 12); Xe cứu hỏa (class: 13).
* Bởi vì lúc label các đối tượng thì chúng có định dạng XML mà YOLO đọc file TXT thì em viết 1 tool là hàm chuyển đổi từ định dạng XML sang TXT. Sử dụng thư viện ElementTree để xử lí file XML, sử dụng thêm module OS để xử lí file và thư mục trực tiếp trên google collab .Link : <https://colab.research.google.com/drive/1FRP1ZNbdTdE4CuKBM3131cgHrqB_1X2E>.
  1. ***Train, predict và đánh giá mô hình***
* Sau đây là từng bước thực hiện Training dữ liệu trên Google Collab:
* Bước 1: Liên kết drive và clone Darknet của AlexayAB về.



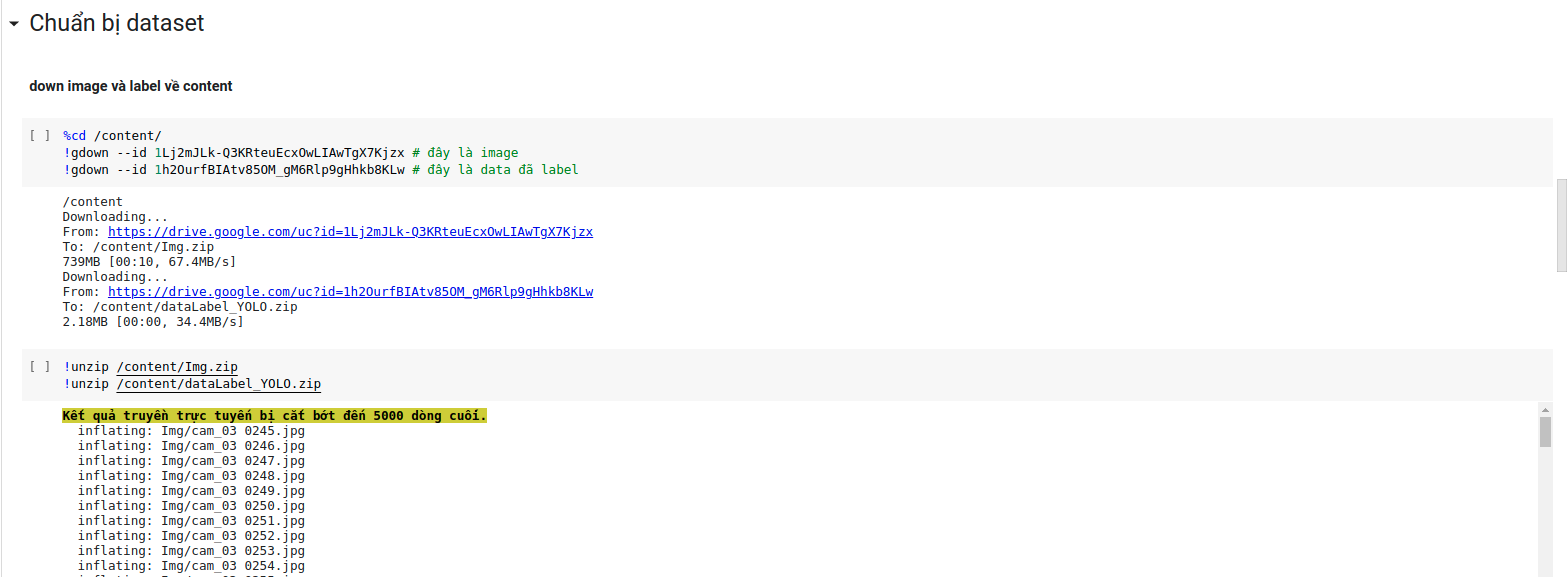
* Bước 2: Make File để sử dụng GPU , CuDNN, OpenCV cho training



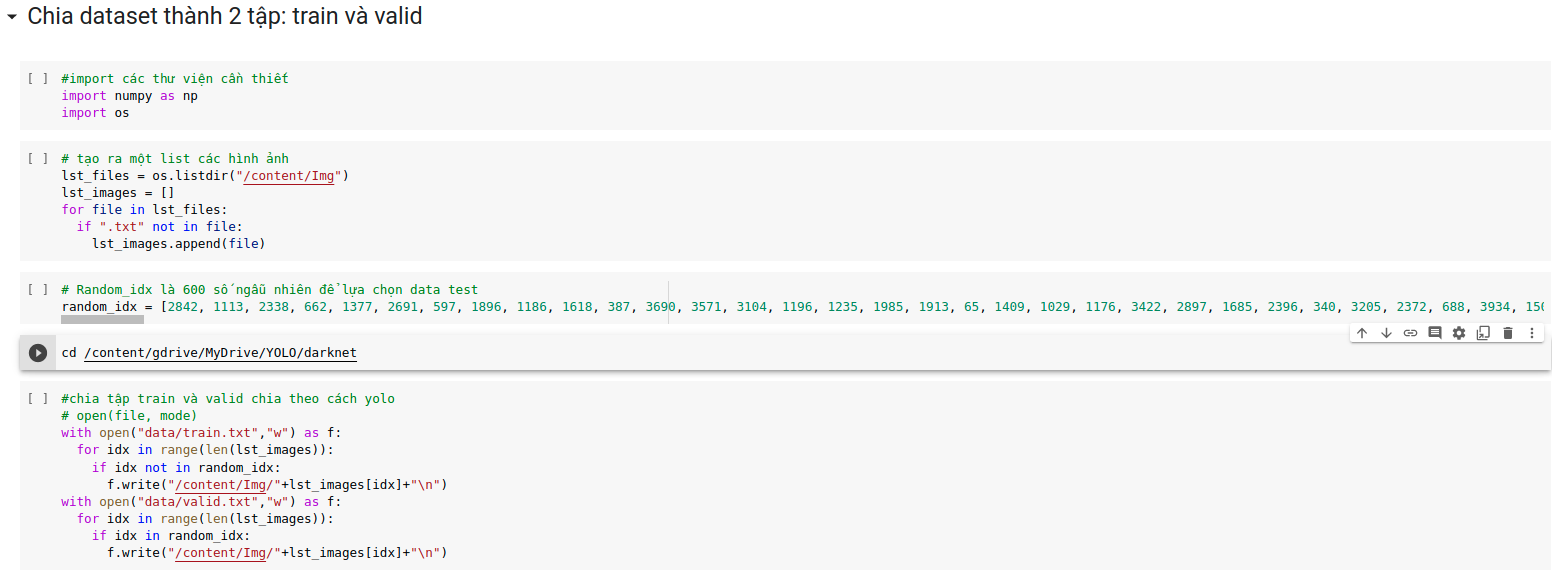
* Bước 3: Bắt đầu config, thay đổi một vài thông số trong file Yolov3\_training.cfg trong Darknet/Config để phù hợp với bộ dữ liệu của em. Ví dụ như số class của bộ dữ liệu MSCOCO là 80 nhưng dữ liệu em đánh nhãn để train là 14 class.



* Bước 4: Chuẩn bị dataset, tải về và giải nén. Bộ dataset thì em để public nên thầy có thể dowload về và chạy thực nghiệm.



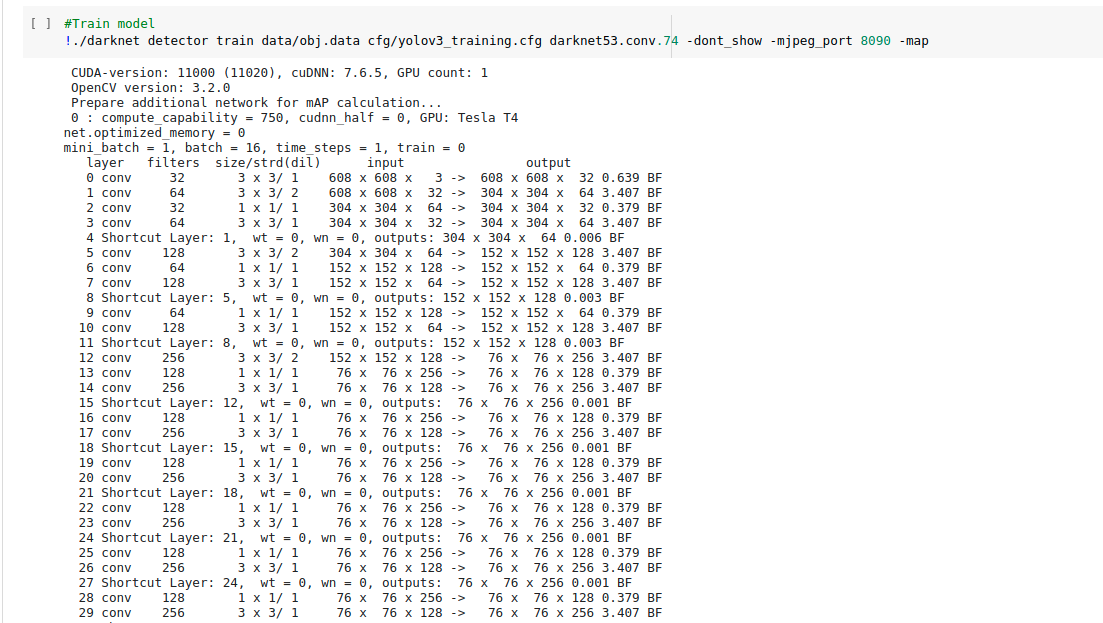
* Bước 5: Chia data set thành 2 tập train và valid để đánh giá. Tạo ra random\_idx là 600 data dùng để test nhưng vì train trên collab giới hạn thời gian và để lần sau tiếp tục vào chạy không bị sai dữ liệu test thì em chia nó thành 600 số cố định.



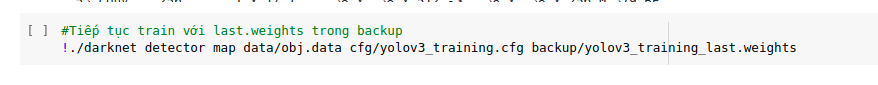
* Bước 6: Tải pretrain model để chuẩn bị cho bước train dữ liệu



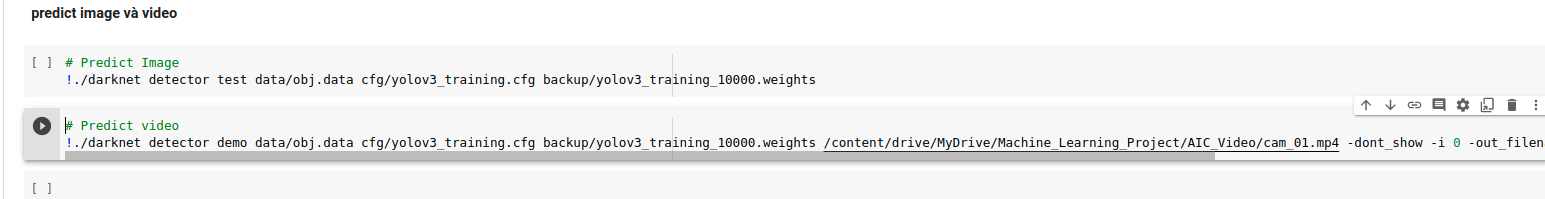
* Bước 7:Train model



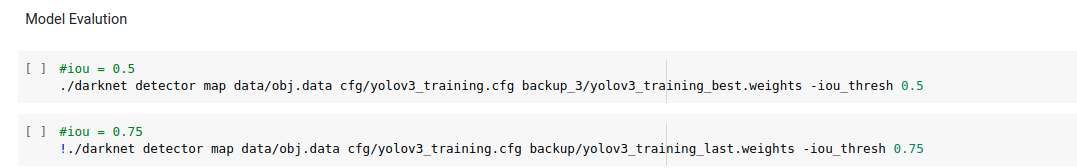
* Bước 8: Để có được file weight 10000 iterations thì phải thực hiện train nhiều lần sau mỗi 12 tiếng ở trên Collab, và để tiếp tục train ta dùng file weight được lưu mỗi 1000 iterations trong file Darknet/Backup.



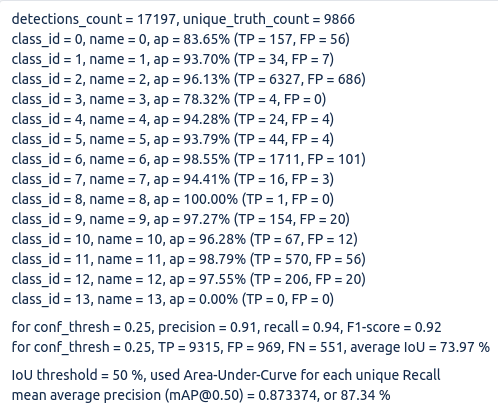
* Bước 9: Predict video and image. Sau khi training được file weight 10000 Ta sử dụng 2 câu lệnh sau cho predict. Thầy có thể predict và dùng file 10000 weight của em đã train được ở đây:
* https://drive.google.com/file/d/1-7i\_O9eK6hxn2rcV-4up3W0G5q8SJRAp



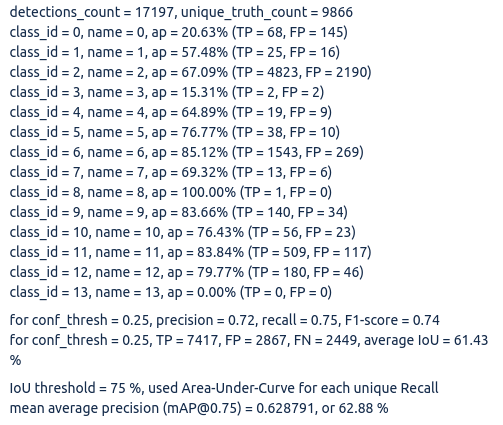
* Bước 10: Đánh giá mô hình ở ngưỡng iou lần lượt là 0.5 và 0.75:



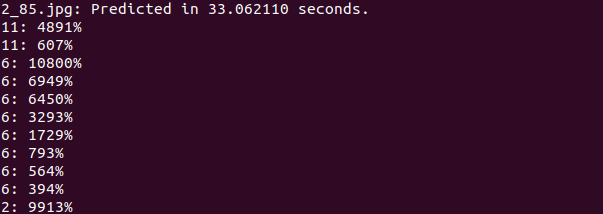
* Ở ngưỡng iou = 0.5 ta thu được mAP là 87.43%



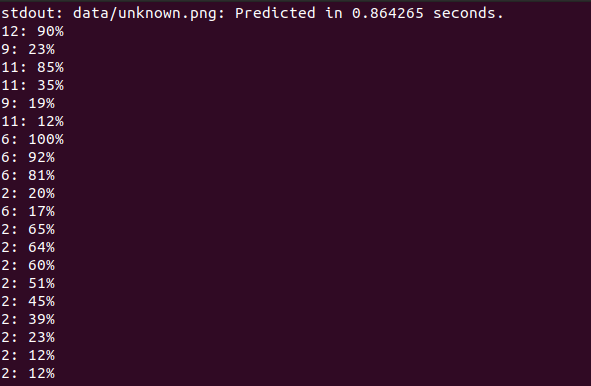
* Ở ngưỡng iou = 0.75 ta thu được mAP là 62.88%



* 1. ***Sử dụng GPU laptop nhận diện và web demo***
* Để có thể chạy được nhận diện ảnh trên laptop thì cũng cần phải clone Darknet của AlexeyAB về.
* Yêu cầu cấu hình: Có card rời Nvidia , cài đặt Opencv.
* Tinh chỉnh một vài thông số trong data/coco.names data/obj.names để phù hợp với laptop và file weight 10000 có được sau khi train trên collab.
* Trong file cfg/obj.data và cfg/yolov3\_training.cfg thì em tải trên collab về để đồng bộ với khi train trên collab. Ngoài ra còn tinh chỉnh trong src,...
* Thực hiện tinh chỉnh một vài thông số trong makefile và thực hiện makefile trên Temiral.
* Nếu sử dụng CPU thì tốc độ detected nhận được là 33s cho 1 tấm ảnh và nó không thể thực hiện được mục đích ban đầu là real time nên em chuyển sang dùng GPU.



* Sau khi thử cùng tấm ảnh với GPU thì tốc độ đạt được chưa đến 1s và thực hiện Website để demo.



* 1. ***Demo***
* Website demo của em có giao diện như sau:



* Với nút chọn file cho phép chọn hình ảnh hoặc video để thực hiện detection.
* Theo mặc định thì Yolo sẽ hiển thị các đối tượng có độ tin cậy lớn hơn 25% và Dòng số 2 Threshould cho phép ta chọn ngưỡng phát hiện đối tượng.
* Thực hiện Upload và xem Model Predict

# TỔNG KẾT

* Trong chương này em sẽ tổng hợp lại những gì em đã làm được đối với bài toán này.

1. **Những kết quả đạt được**

* Hiểu được quá trình thu thập và gán nhãn dữ liệu, thấy được sự thuận lợi và khó khăn khi thực nghiệm bộ dữ liệu
* Tìm hiểu về bài toán Object Detection
* Tiếp cận được với một số thuật toán cơ bản của Deep Learning (CNN, YOLO,...)
* Áp dụng được những gì đã học ở môn Python vào việc tiền xử lí dữ liệu.
* Biết đuợc cách làm việc với file và thư mục.

1. **Khó khăn**

* Tìm hiểu về cách hoạt động của bài toán Object Detection còn gặp nhiều khó khăn do chưa có kiến thức nền tảng
* Quá trình thu thập và gán nhãn dữ liệu gặp nhiều khó khăn, tốn nhiều thời gian cho việc thu thập dữ liệu.
* Làm việc một mình nên chưa đạt được hiệu quả tối đa

1. **Hướng phát triển**

*Một số hướng phát triển cho đồ án chuyên ngành này bao gồm:*

* Tiếp tục tìm hiểu và hoàn thiện bài toán - Tracking và Counting
* Những kiến thức đã học được là nền tảng để em tìm hiểu và tiếp cận thêm những phương pháp tối ưu và hiện đại hơn, từ đó giúp tăng độ chính xác của bài toán lên cao hơn.