**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**



**BÁO CÁO NIÊN LUẬN CƠ SỞ - CNTT**

**ĐỀ TÀI**

**NHẬN DẠNG VẬT THỂ VỚI YOLO VÀ OPENCV**

**BẰNG NGÔN NGỮ PYTHON**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:**  **TS. Phạm Thế Phi** | **Sinh viên thực hiện:**  **Nguyễn Hồ Quốc Huy**  **MSSV: B1809128**  **Lớp: DI18V7A1** |

**CẦN THƠ 11/2021**

# LỜI CẢM ƠN

Để hoàn thành bài báo cáo này, em xin gửi lời cảm ơn chân thành đến:

Ban giám hiệu trường Đại Học Cần Thơ vì đã tạo điều kiện về cơ sở vật chất với hệ thống thư viện hiện đại, đa dạng các loại sách, tài liệu thuận lợi cho việc tìm kiếm, nghiên cứu thông tin.

Xin cảm ơn giảng viên hướng dẫn – TS. Thầy Phạm Thế Phi đã hướng dẫn tận tình, chi tiết và chia sẻ nhiều kiến thức để chúng em có thể vận dụng chúng vào bài tiểu niên luận cơ sở này.

Do chưa có nhiều kinh nghiệm làm để tài cũng như những hạn chế về kiến thức, trong bài niên luận cơ sở chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự nhận xét, ý kiến đóng góp, phê bình từ phía Thầy để bài niên luận cơ sở được hoàn thiện hơn.

Lời cuối cùng, em xin kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc.

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc87190830)

[MỤC LỤC 3](#_Toc87190831)

[DANH MỤC HÌNH ẢNH 5](#_Toc87190832)

[TÓM TẮT 6](#_Toc87190833)

[PHẦN GIỚI THIỆU 7](#_Toc87190834)

[**1. Lý do chọn đề tài** 7](#_Toc87190835)

[**2. Mục đích chọn đề tài** 7](#_Toc87190836)

[**3. Đối tượng và phạm vi nghiên cứu** 7](#_Toc87190837)

[**4. Phương pháp nghiên cứu** 7](#_Toc87190838)

[**5. Nội dung nghiên cứu** 7](#_Toc87190839)

[PHẦN NỘI DUNG 8](#_Toc87190840)

[CHƯƠNG 1 – ĐẶC TẢ YÊU CẦU 8](#_Toc87190841)

[**1. Các chức năng:** 8](#_Toc87190842)

[**2. Mô tả các chức năng:** 8](#_Toc87190843)

[**CHƯƠNG 2 – GIỚI THIỆU NỀN TẢNG CÔNG NGHỆ** 9](#_Toc87190844)

[**1. Computer Vision** 9](#_Toc87190845)

[**2. OpenCV** 9](#_Toc87190846)

[**3. YOLOV3** 9](#_Toc87190847)

[4. Ngôn ngữ Python 10](#_Toc87190848)

[CHƯƠNG 3 – CÀI ĐẶT GIẢI PHÁP 11](#_Toc87190849)

[**1. Cài đặt ngôn ngữ Python** 11](#_Toc87190850)

[**2. Cài đặt cấu trúc thư mục** 11](#_Toc87190851)

[**3. Cài đặt thư viện** 12](#_Toc87190852)

[CHƯƠNG 4 – XỬ LÝ VÀ NHẬN DẠNG HÌNH ẢNH 13](#_Toc87190853)

[**1. YOLOv3** 13](#_Toc87190854)

[**2. Xử lý hình ảnh** 13](#_Toc87190855)

[**3. Kết quả nhận dạng** 15](#_Toc87190856)

[**PHẦN KẾT LUẬN** 17](#_Toc87190857)

[TÀI LIỆU THAM KHẢO 18](#_Toc87190858)

# DANH MỤC HÌNH ẢNH

[Hình 1: Biểu đồ thể hiện độ chính xác trung bình (mAP) 10](#_Toc87190631)

[Hình 2: Cài đặt Python 11](#_Toc87190632)

[Hình 3: Phiên bản Python trên máy sau khi cài đặt thành công 11](#_Toc87190633)

[Hình 4: Cấu trúc thư mục 11](#_Toc87190634)

[Hình 5: Hình ảnh huấn luyện với vật thể là ô tô 13](#_Toc87190635)

[Hình 6:YOLO đưa ra dự đoán cách xác định hộp giới hạn 13](#_Toc87190636)

[Hình 7: Hình ảnh sau khi được chia tỷ lệ khung hình 14](#_Toc87190637)

[Hình 8: Vật thể sau khi phân tích các tọa độ (x, y , w, h) 14](#_Toc87190638)

[Hình 9: Hình ảnh vật thể sau khi phân tích thành các hộp giới hạn tin cậy 14](#_Toc87190639)

[Hình 10: Kết quả sau khi triệt tiêu các khung hình kép bằng non-maxima suppression 15](#_Toc87190640)

[Hình 11: Kết quả sau khi nhận dạng vật thể ô tô 15](#_Toc87190641)

[Hình 12: Kết quả nhận dạng các vật thể trong ảnh 15](#_Toc87190642)

[Hình 13: Nhận dạng vật thể qua video.mp4 16](#_Toc87190643)

[Hình 14: Nhận diện qua Camera IP 16](#_Toc87190644)

# TÓM TẮT

Qua các hướng dẫn của thầy và thu thập thông tin từ các nguồn internet, em đã học được cách xây dựng chương trình nhận diện vật thể từ thư viện OpenCV và YOLOv3 bằng ngôn ngữ Python.

Chương trình sử dụng phương pháp học sâu trong lĩnh vực thị giác máy tính huấn luyện chính xác dữ liệu nhằm thu được kết quả tốt nhất.

Chương trình áp dụng phương thức nhận dạng như sau:

1. Nhận dạng dữ liệu hình ảnh hoặc webcam.
2. Kiểm tra độ chính xác huấn luyện bằng cách xuất kết quả ra màn hình và lưu lại trong thư mục.

Chương trình có thể huấn luyện vật thể cơ bản với 80 lớp được huấn luyện từ thư viện YOLOv3 cho tốc độ nhanh hơn và kết quả chính xác hơn 90%.

# PHẦN GIỚI THIỆU

**1. Lý do chọn đề tài**

- Trí tuệ nhân tạo đang ngày càng phát triển trong đời sống hiện nay, góp phần không nhỏ đến sự phát triển trong rất nhiều lĩnh vực nhất được đẩy mạnh trong thời kỳ công nghệ 4.0. Trong đó nhận dạng trở thành 1 phần không thể thiếu khi nhắc đến các hệ thống bảo mật ứng dụng trí tuệ nhân tạo có thể phổ biến như: điểm danh vân tay, khuôn mặt, móng mắt, …

- Vì là lĩnh vực được đầu tư phát triển mạnh mà nhận dạng vật thể không còn xa lạ với các công trình nghiên cứu khoa học về ứng dụng trong việc xác thực vật thể như nhận dạng biển số xe, nhận dạng vật thể trong công nghệ ô tô tự lái nhằm đưa người sử dụng hướng đến những tiến tiến trong công nghệ về sản phẩm thông minh, đáp ứng được nhu cầu người dùng.

**2. Mục đích chọn đề tài**

- Nghiên cứu đề tài nhằm mục đích tìm hiểu cách nhận dạng vật thể xác định trong cuộc sống xung quanh, xây dựng chương trình nhận dạng áp dụng đưa vào thực tiễn.

**3. Đối tượng và phạm vi nghiên cứu**

- Đối tượng: là vật thể được ghi lại bằng hình ảnh qua file \*.jpg hoặc dữ liệu trực tiếp từ camera hoặc webcam được nhận dạng phát hiện từ tập dữ liệu.

- Phạm vi nghiên cứu: do kiến thức và thời gian hạn chế do dịch Covid-19, hệ thống tập dữ liệu thực tế còn ít, nên em sử dụng các dữ liệu có sẵn từ thư viện và các dữ liệu thu thập từ internet.

**4. Phương pháp nghiên cứu**

- Hình ảnh được chụp hoặc hình ảnh ghi hình từ camera.

- Xử lý dữ liệu bằng chương trình nhận dạng vật thể.

- Hiển thị dữ liệu trực tiếp trên webcam, camera an ninh hoặc dữ liệu được chụp lại và lưu trữ tại thư mục sau khi được phân tích thành công.

**5. Nội dung nghiên cứu**

- Xây dựng cơ sở dữ liệu là các tệp hình ảnh.

- Xây dựng thuật toán dựa trên thư viện có sẵn.

# PHẦN NỘI DUNG

# CHƯƠNG 1 – ĐẶC TẢ YÊU CẦU

**1. Các chức năng:**

- Hiển thị nhận dạng các vật thể thông qua (tên-độ chính xác tương đối).

**2. Mô tả các chức năng:**

- Truyền đường dẫn đến thư mục chứa tập tin ảnh hoặc nhập vào phương pháp nhận dạng với webcam (webcam máy tính hoặc truy cập camera an ninh thực tế).

* **Trình tự thực hiện:**

- Truyền hình ảnh để phần mềm xử lý.

- Phần mềm xử lý nhận dạng đối tượng

- Xuất hình ảnh với các thông số bao gồm: tên, độ chính xác nhận dạng, khung nhận dạng vật thể

* **Kết quả:**

**-** Hiển thị nhận dạng các vật thể thông qua (tên-độ chính xác tương đối).

**-** Đối với file input là ảnh thì đầu ra sẽ được lưu tại thư mục lưu trữ giúp chúng ta xem lại và phân tích kết quả thu được

**CHƯƠNG 2 – GIỚI THIỆU NỀN TẢNG CÔNG NGHỆ**

**1. Computer Vision**

**-** Computer Vision được dịch là Thị giác máy tính, là một thuật ngữ được dùng nhiều trong ngành Khoa học máy tính liên quan đến lĩnh vực khoa học áp dụng tạo ra máy tính có được sự hiểu biết cao cấp từ hình ảnh hoặc video kỹ thuật số mà qua đó máy tính học được cách tự động hóa, áp dụng để thay thế được hệ thống thị giác của con người nhằm đáp ứng nhu cầu phục vụ đời sống hằng ngày qua các ứng dụng công nghệ thông tin trong thời kì công nghệ 4.0.

**-** Thị giác máy tính là một công nghệ được đào tạo thông qua cơ sở lý thuyết các dữ liệu thô sau đó được xử lý thông qua các mô hình và đưa vào các chuỗi dữ liệu để các thuật toán giải quyết và đưa ra các kết quả giúp máy tính học được thông qua các thuộc tính được áp dụng.

**2. OpenCV**

- OpenCV viết tắt cho Open Source Computer Vision Library, là một thư viện mã nguồn mở hàng đầu cho Computer Vision và Machine Learning. Hiện nay thư viện được phần lớn lập trình viên sử dụng vì tính đáp ứng của thư viện hỗ trợ rất tốt cho việc nâng cao hiệu năng đối với GPU và cũng bởi được phát hành dựa trên giấy phép BSD nên nó miễn phí ngay cả học tập lẫn thương mại hóa.

- OpenCV hỗ trợ rất nhiều tiến trình trên các giao diện như C/C++, Python, Java và trên các hệ điều hành như Window, Linux, MacOS.

**-Tính ứng dụng của OpenCV:**

+ Nhận dạng hình ảnh street view

+ Kiểm tra và giám sát tự động

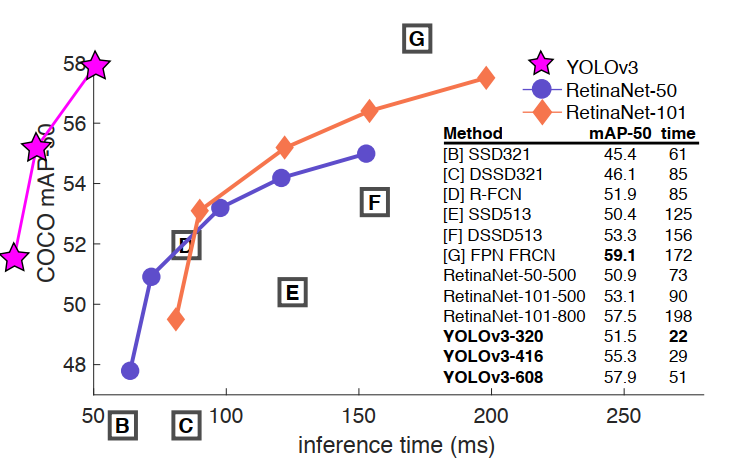
+ Xử lý hình ảnh trong phân tích y học

+ Phục hồi hình ảnh, video

+ Robot và ô tô tự lái

**3. YOLOV3**

- YOLO viết tắt là You only look once, là một trong những thuật toán phát hiện đối tượng thời gian thực nhanh nhất (45 fps/giây) so với họ R-CNN hay các thuật toán học sâu như SSD MobileNet xử lý chụp đơn ảnh. Trong một nghiên cứu về độ chính xác trung bình mean Average Precision (mAP) YOLOv3 cho ra kết quả rất tốt và được chọn là thuật toán có hiệu năng ổn định.



Hình : Biểu đồ thể hiện độ chính xác trung bình (mAP)

- YOLO áp dụng mạng nơ ron của toàn bộ hình ảnh để dự đoán các hộp giới hạn và xác suất của chúng. Và sau rất nhiều phiên bản cải thiện về độ chính xác cũng như hiệu suất YOLOV3 được cho ra mắt với tập huấn luyện coco với hơn 80 lớp với các vật thể được gắn nhãn không giới hạn như: Con người, xe đạp, ô tô, máy bay, biển báo, động vật, …

## 4. Ngôn ngữ Python

- Ngôn ngữ Python được thiết kế phát triển bởi Guido Van Rossum, hiện nay được sử dụng rất phổ biến trong phát triển hướng đối tượng được rất nhiều lập trình viên ưa thích. Có cấu trúc dữ liệu mạnh mẽ cách tiếp cận dễ dàng cho người mới bắt đầu và những phần mềm hướng đối tượng.

- Tính năng chính của Python: là một thư viện mã nguồn mở, miễn phí, ngôn ngữ lập trình dễ học dựa trên C/C++, thư viện chuẩn giúp cho nhiều tác vụ sử dụng chung, hướng đối tượng mạnh mẽ, … là những tính năng mà Python đem lại và ngày càng phát triển bằng ngôn ngữ này.

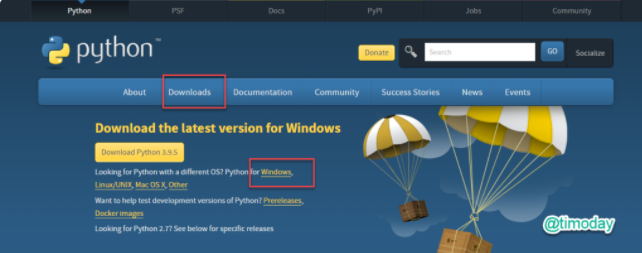
- Tính ứng dụng của Python: được dùng trong lập trình website, khoa học dữ liệu và tính toán, xử lý hình ảnh trong Thị giác máy tính, nguyên lý máy học, trí tuệ nhân tạo, khuôn mẫu phần mềm, …

# CHƯƠNG 3 – CÀI ĐẶT GIẢI PHÁP

**1. Cài đặt ngôn ngữ Python**

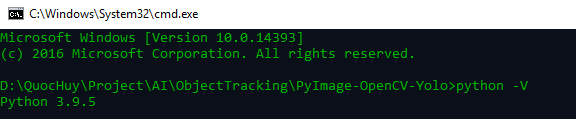
Chương trình chạy trên hệ điều hành Window sử dụng ngôn ngữ Python.

- Tiến hành cài đặt Python tại trang chủ: <https://www.python.org/downloads/>



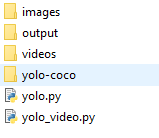
Hình : Cài đặt Python

- Thực hiện cài đặt với file setup.exe sau đi cài đặt thành công kiểm tra bằng lệnh sau:



Hình : Phiên bản Python trên máy sau khi cài đặt thành công

**2. Cài đặt cấu trúc thư mục**



Hình : Cấu trúc thư mục

- images/videos: để lưu trữ dữ liệu cần phát hiện đối tượng.

- yolo-coco: lưu trữ file weights của thuật toán YOLO và file gán nhãn.

- yolo.py và yolo\_video.py: file python thực thi.

**3. Cài đặt thư viện**

**- YOLOv3:** tải file .zip tại link vào thư mục yolo-coco và giải nén

<https://drive.google.com/drive/folders/1vTk3Xfv7uWRIRwGgZRLTKwEHiOMw4cZB>

<https://pjreddie.com/darknet/yolo/>

**- OpenCV:** pip install opencv-python

**- Numpy:** pip install numpy

**- Imutils:** pip install imutils

**- Argparse:** pip install argparse

# CHƯƠNG 4 – XỬ LÝ VÀ NHẬN DẠNG HÌNH ẢNH

**1. YOLOv3**

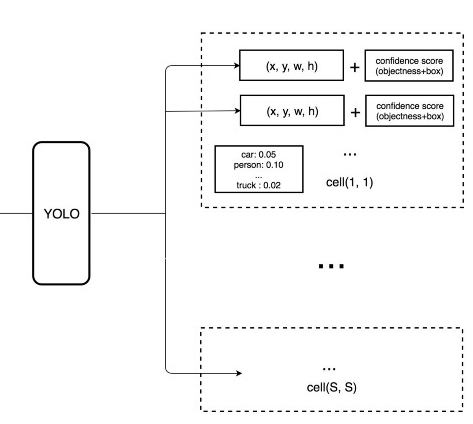
- Các vật thể được đào tạo huấn luyện là những vật dụng xung quanh chúng ta, được đưa vào và huấn luyện nhận dạng bằng thuật toán YOLOv3 với hơn 80 nhãn không giới hạn số vật thể xuất hiện trong một khung hình.



Hình : Hình ảnh huấn luyện với vật thể là ô tô

**2. Xử lý hình ảnh**

- Để thực hiện các bươc nhận dạng em sử dụng đọc dữ liệu YOLOv3 bằng sự hỗ trợ của thư viện OpenCV với **cv2.dnn.readNet** đọc các tệp cấu hình và trọng số của YOLO. Sau đó lấy dữ liệu nhãn từ tập **coco.names** lưu trữ dưới dạng các lớp và thông qua **cv2.dnn** chuyển sang dữ liệu trả về các lớp để phân tích nhận dạng.



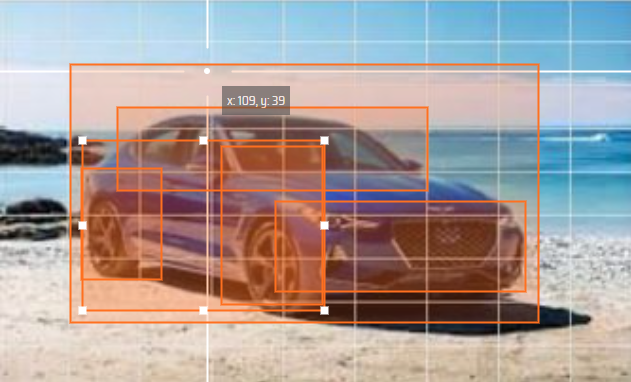
Hình 6:YOLO đưa ra dự đoán cách xác định hộp giới hạn

- Hình ảnh được đưa qua xử lý nhận dạng chia tỷ lệ khung hình thành các ô lưới thành mỗi đối tượng S x S thông qua **cv2.dnn.blobImage** để chấp nhận khung hình, các lớp mô hình và đầu ra làm tham số.



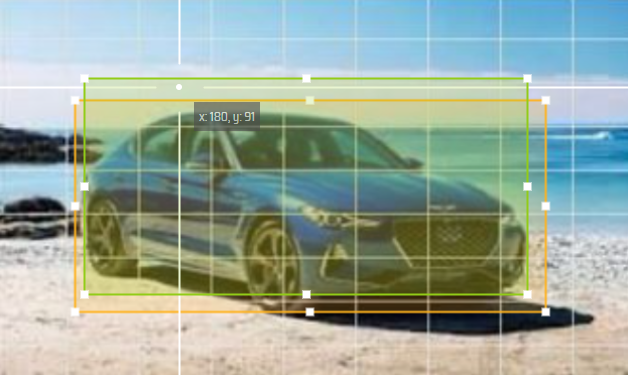
Hình : Hình ảnh sau khi được chia tỷ lệ khung hình

- Lọc và chia các khung hình cho vật thể, chia nhỏ tiếp theo tỷ lệ pixel hình ảnh trong phạm vi từ 0 đến 1, sau đó trả về một danh sách lồng nhau :



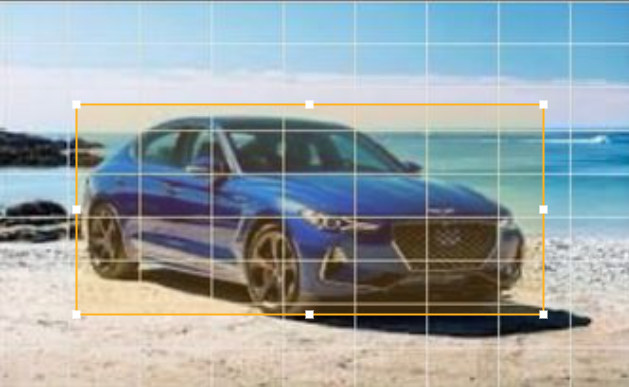
Hình : Vật thể sau khi phân tích các tọa độ (x, y , w, h)

- Vật thể sau khi phân tích được bố cục chính bao gồm tọa độ (x, y, w, h) của tâm đối tượng được phát hiện, chiều cao và chiều rộng của hộp giới hạn, độ tin cậy và điểm cho tất cả các lớp đối tượng được đem đi phân tích trực quan trong tập dữ liệu **coco.names**:



Hình : Hình ảnh vật thể sau khi phân tích thành các hộp giới hạn tin cậy

**-** Các hộp giới hạn vật thể có thể bị trùng lặp nhiều lần do giới hạn trong phạm vi từ 0 đến 1 có nhiều giá trị xấp xỉ, để đảm bảo cho việc không chồng chéo nhận dạng 1 đối tượng có nhiều hộp giới hạn nhận dạng chúng ta sử dụng, vì vậy mà chúng ta sử dụng chế độ triệt tiêu không cực đại (**non-maxima suppression**) để lấy kết quả tối ưu nhất của tập huấn luyện.



Hình : Kết quả sau khi triệt tiêu các khung hình kép bằng non-maxima suppression

**3. Kết quả nhận dạng**

- Mô hình được huấn luyện cho các kích thước hình ảnh khác nhau cho ra các tốc độ huấn luyện và độ chính xác khác nhau: 320x320(tốc độ cao, độ chính xác kém), 416x416(tốc độ trung bình, độ chính trung bình), 618x618(tốc độ thấp, độ chính cao). Ở chương trình này em sử dụng kích thước hình ảnh 416x416 cho kết quả tương đối chính xác.

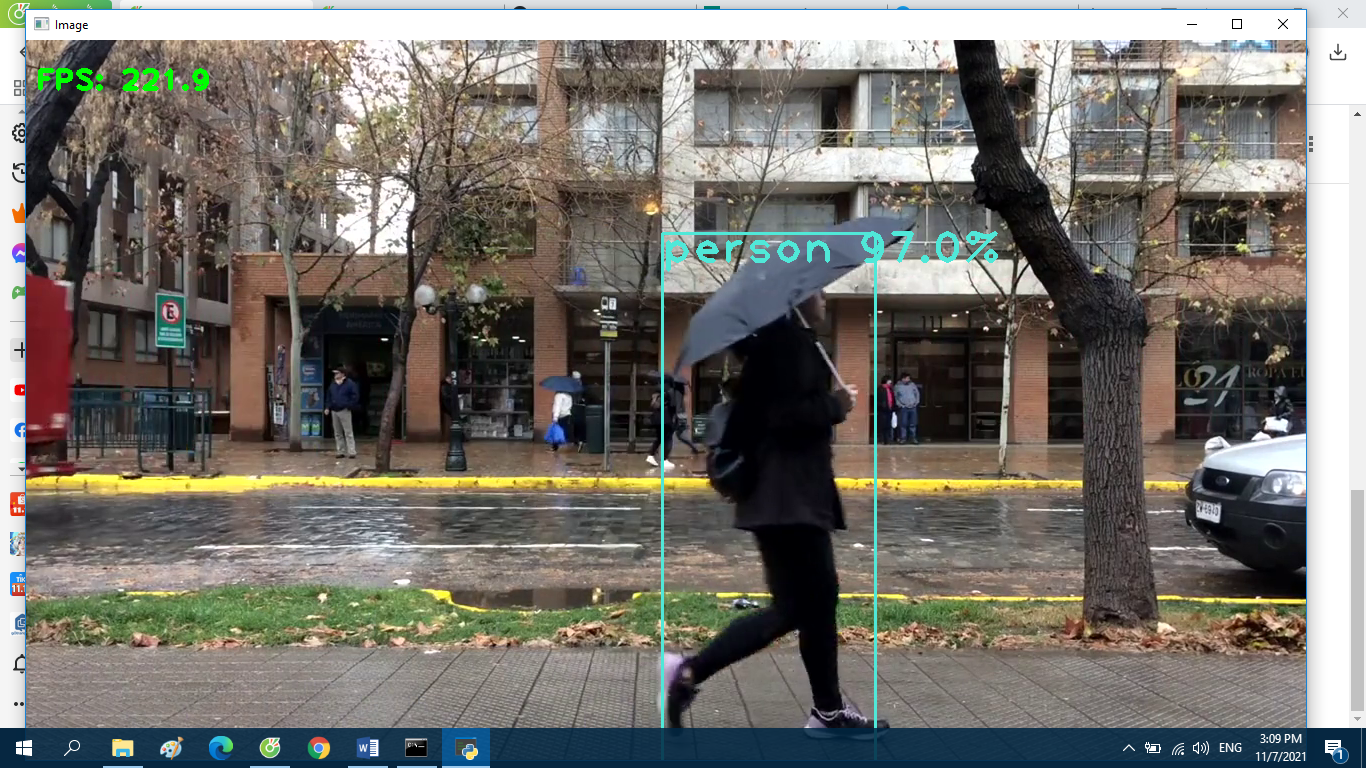


Hình : Kết quả sau khi nhận dạng vật thể ô tô

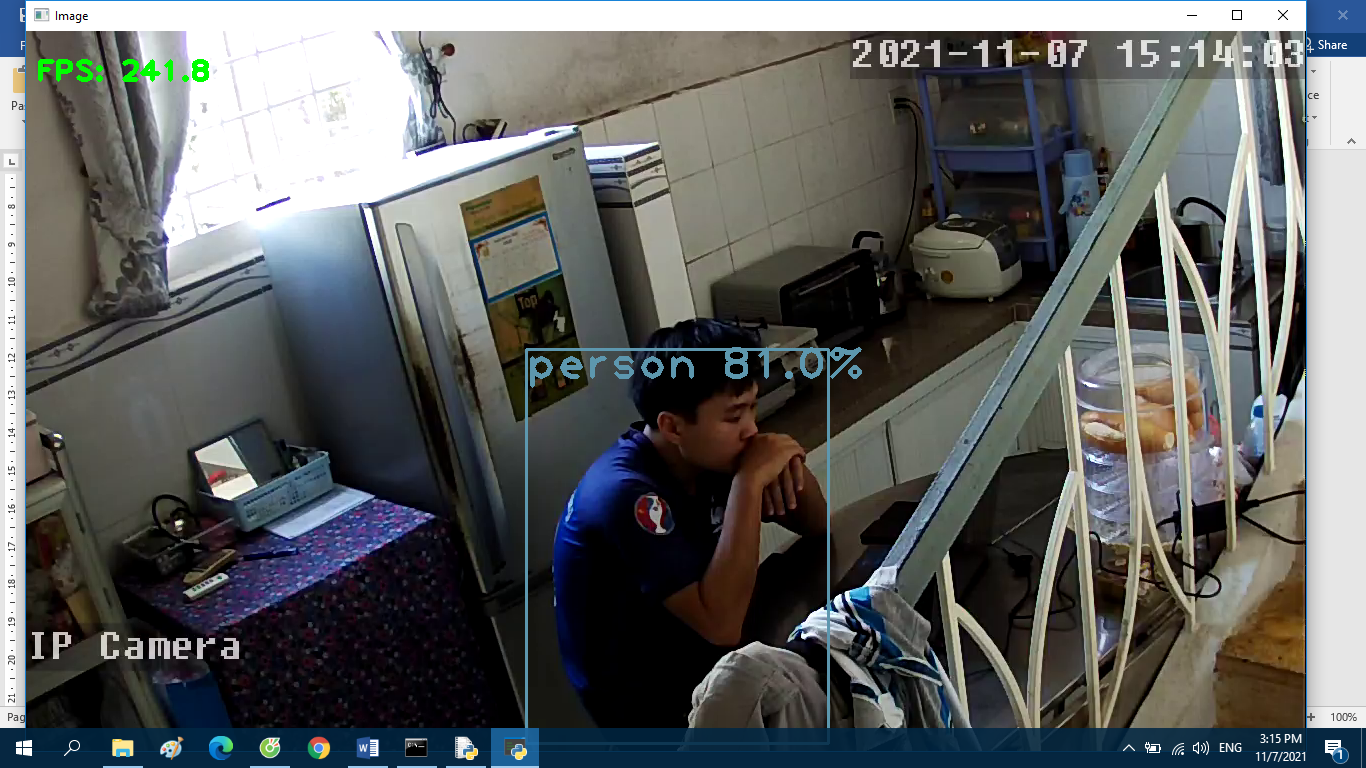


Hình 12: Kết quả nhận dạng các vật thể trong ảnh

-Huấn luyện webcam/camera thời gian thực hoặc với file video.mp4 bằng yolo-tiny cho tốc độ xử lý cao hơn so với YOLOv3, thuật toán được cải thiện đáng kể nhằm tăng tốc độ xử lý trên GPU cũng như độ chính xác cho từng khung hình theo thời gian. Chúng ta thu được kết quả như sau:



Hình 13: Nhận dạng vật thể qua video.mp4



Hình 14: Nhận diện qua Camera IP

**PHẦN KẾT LUẬN**

Ứng dụng nhận dạng vật thể với thư viện OpenCV và thuật toán YOLOv3 giúp chúng ta nhận dạng được vật thể đặc thù với tốc độ nhận dạng nhanh, độ chính xác tương đối ổn định và tốc độ khung hình trên GPU nhanh hơn so với thuật toán R-CNN tuy nhiên cần được cải thiện hơn để đảm bảo chất lượng đầu ra. Để đạt được kết quả chính xác, chương trình được vận dụng vào xử lý hình ảnh/video hoặc qua các ứng dụng video stream như webcam hoặc camera an ninh, … giúp ích rất nhiều cho việc phát triển các công trình nghiên cứu nhận dạng để cải thiện quy trình xử lý công việc. Do thời gian hạn chế và kiến thức chưa đủ để nghiên cứu nên em chưa thể xử lý đào tạo huấn luyện tập dữ liệu tự do nên em sẽ cố gắng cải thiện trong thời gian sắp tới.

**Kết quả đạt được:**

- Cài đặt và sử dụng có hiệu quả thư viện OpenCV và thuật toán YOLOv3.

- Tìm hiểu và cài đặt chương trình ứng dụng nhận dạng vật thể.

- Tìm hiểu và học được quá trình làm việc trong môi trường lập trình với các nguyên lý học sâu.

**Hướng phát triển tương lai:**

- Thu thập thêm tập dữ liệu hình ảnh để huấn luyện cho mô hình kế tiếp với các thuật toán khác.

- Cải thiện tốc độ và độ chính xác cho chương trình nhận dạng.

- Huấn luyện tập dữ liệu tự chọn, đưa hình ảnh vào và xác định các nhãn theo thực tế.

# TÀI LIỆU THAM KHẢO

1. [https://www.pyimagesearch.com/2018/11/12/YOLO-object-detection-with-opencv/](https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/)

2. <https://arxiv.org/abs/1506.02640v3>

3. <https://timviec365.vn/blog/computer-vision-la-gi-new8985.html>

4. [https://ichi.pro/vi/phat-hien-doi-tuong-bang-YOLOv3-va-opencv-28510119961632](https://ichi.pro/vi/phat-hien-doi-tuong-bang-yolov3-va-opencv-28510119961632)

5. <https://ichi.pro/vi/phat-hien-doi-tuong-trong-thoi-gian-thuc-voi-yolo-yolov2-va-bay-gio-la-yolov3-44743782178952>

Source Code: <https://github.com/b1809128/yolo-object-detection-machine-learning>