



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



**BÁO CÁO THỰC HÀNH
LAB 2**

**TÊN ĐỀ TÀI: NGHIÊN CỨU MỘT SỐ THƯ VIỆN TRONG
PHÁT HIỆN ĐỐI TƯỢNG**

**TÊN ĐỀ TÀI TIẾNG ANH: RESEARCH SOME LIBRARIES IN
OBJECT DETECTION**

**GIẢNG VIÊN HƯỚNG DẪN
TS. DƯƠNG VIỆT HẰNG**

**HỌC VIÊN THỰC HIỆN
NGUYỄN HỒNG SƠN**

1 Tóm tắt đề tài

Trong báo cáo này, học viên trình bày một số nội dung cơ bản, phương pháp sử dụng và một số ví dụ của các thư viện được áp dụng trong bài toán phát hiện đối tượng. Các nội dung được trình bày sẽ bao gồm nhưng không giới hạn bởi khái niệm, phương pháp huấn luyện, phương pháp dự đoán, các cài đặt và các nội dung khác. Do giới hạn về tài nguyên và thời gian, các nội dung training trong các công cụ không được đề cập trong báo cáo này. Khi so sánh về hiệu suất huấn luyện của các công cụ, các kết quả từ các nghiên cứu đã được công bố sẽ được sử dụng thay vì kết quả thực nghiệm của học viên.

Phần 2 giới thiệu tổng quan về bài toán phát hiện đối tượng và các thư viện TensorFlow Object Detection API, Detectron2, MXNet GluonCV, MMDetectio cùng các ví dụ của chúng. Phần 3 so sánh các thư viện với nhau và đưa ra kết luận. Cuối cùng là tài liệu tham khảo

2 Giới thiệu

2.1 Tổng quan về bài toán phát hiện đối tượng

Phát hiện đối tượng (Object Detection) được coi là một trong những lĩnh vực quan trọng nhất trong nghiên cứu máy học nói chung và xử lý ánh nói riêng. Các mô hình phát hiện đối tượng được training nhằm nhận diện các đối tượng, phát hiện đối tượng là gì trong các hình ảnh hoặc video.

Convolutional Neural Network(CNN) là một kiến trúc học sâu lấy cảm hứng từ nhận thức trực quan của sinh vật. Năm 1980 Kunihiko Fukushima[3] đề xuất neocognitron , có thể coi là tiền thân của CNN. Kể từ đó, nhiều nghiên cứu mới về CNN đã được công bố và đưa ra nhiều thuật toán phát hiện đối tượng mạnh mẽ.

Một số ứng dụng phổ biến trong phát hiện đối tượng bao gồm:

- Nhận dạng biển số. Tính năng phát hiện đối tượng được sử dụng thông qua việc lưu giữ lại hình ảnh và phát hiện các đối tượng cụ thể như xe cộ, phương tiện đi lại trên bức ảnh đó.
- Phát hiện và nhận dạng khuôn mặt: phát hiện khuôn mặt trong hình ảnh hoặc ai trong hình ảnh đó.
- Theo dõi đối tượng: công nghệ này có thể ứng dụng để theo dõi chuyển động của một đối tượng hay đồ vật cụ thể.
- Ô tô tự lái: sử dụng nhằm phát hiện từng đối tượng xung quanh xe để tránh va chạm, tuân thủ luật giao thông.

Các phần dưới, học viên tiến hành sử dụng các API mà không sử dụng cuda.



Hình 2.1: Kết quả của Tensorflow API

2.2 TensorFlow Object Detection API

TensorFlow Object Detection API là một open-source framework để xây dựng các mô hình object detection và image segmentation có thể localize nhiều đối tượng trong cùng một hình ảnh. Framework hỗ trợ cả 2 phiên bản tensorflow nhưng được khuyến khích dùng version 2.

Nhằm minh họa phần sử dụng API này, học viên sử dụng document của Tensorflow[5]. Code sử dụng được trình bày trong [9].

Trong ví dụ minh họa này, hình ảnh được download từ internet thông qua url, đưa vào local API của Tensorflow để xử lý, kết quả được hiển thị như hình 2.1. Trong đó bên trái là hình ảnh mẫu từ tài hướng dẫn, hình bên phải là hình lấy từ internet, thời gian xử lý lần lượt là 28.4 và 30.3 giây.

2.3 Detectron2

Detectron2[10] là một thư viện mã nguồn mở do Facebook AI research phát triển, hỗ trợ cả hai thuật toán detection và segmentation.

Sau quá trình cài đặt thử nghiệm, nhận thấy việc cài đặt tại máy khó khăn và cần nhiều thư viện tương thích, vì vậy sử dụng notebook mẫu của Meta và thay đổi đường dẫn hình ảnh. Code được thể hiện như trong [7].

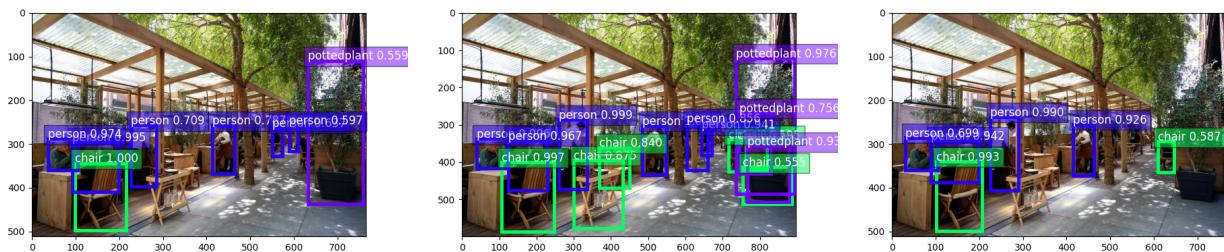
Sử dụng hình ảnh tùy chọn tương tự ví dụ với *TensorFlow*, ta được kết quả như hình 2.2

2.4 MXNet GluonCV

MXNet GluonCV là một framework đang được phát triển bởi Apache Software Foundation [2, 4]. Nó hỗ trợ nhiều bài toán trong xử lý ảnh như Image Classification, Object Detection, Semantic Segmentation, Instance Segmentation... Vì vậy, ngoài các ứng dụng của nó, lượng tài liệu tham khảo và các ví dụ cũng được trình bày rất chi tiết và dễ hiểu.



Hình 2.2: Kết quả của Detectron2



Hình 2.3: Kết quả Object Detection của GluonCV

Vì vậy, trong [8], học viên sẽ trình bày các ứng dụng cùng các mô hình có hỗ trợ.

Đối với bài toán Object Detection, có 3 mô hình là SSD, Faster RCNN và YOLO có ví dụ, kết quả predict theo thứ tự từ trái sang được thể hiện trong hình 2.3

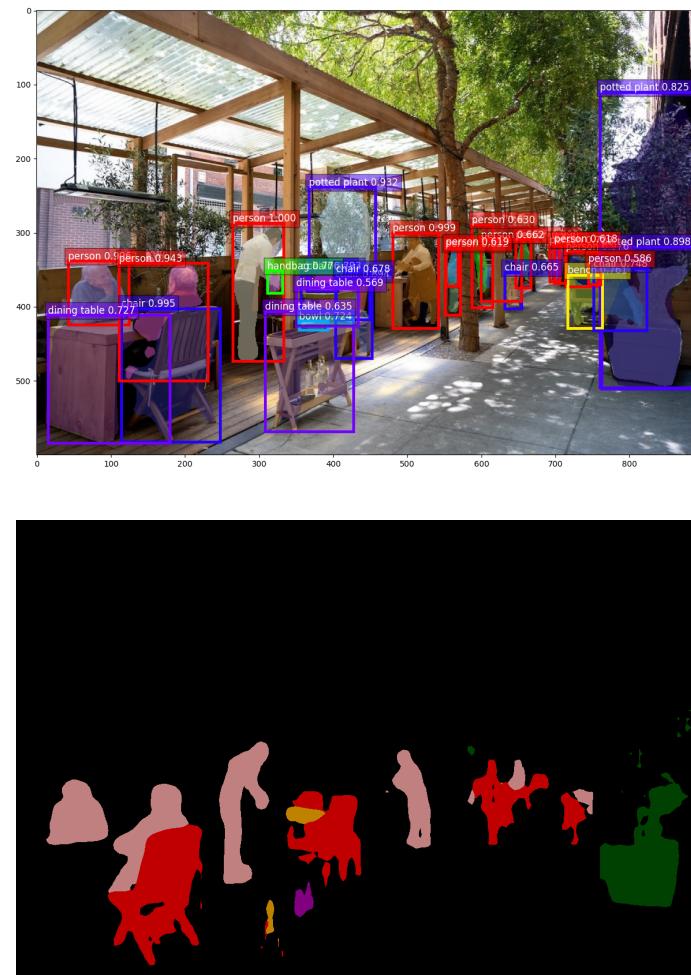
Đối với bài toán Segmentation GluonCV hỗ trợ cả instance và semantic, kết quả pre-trained predict thể hiện trong hình 2.4.

Do giới hạn về thời gian, các ứng dụng khác của GluonCV không được hiện thực hết, có thể tham khảo thêm tại [1].

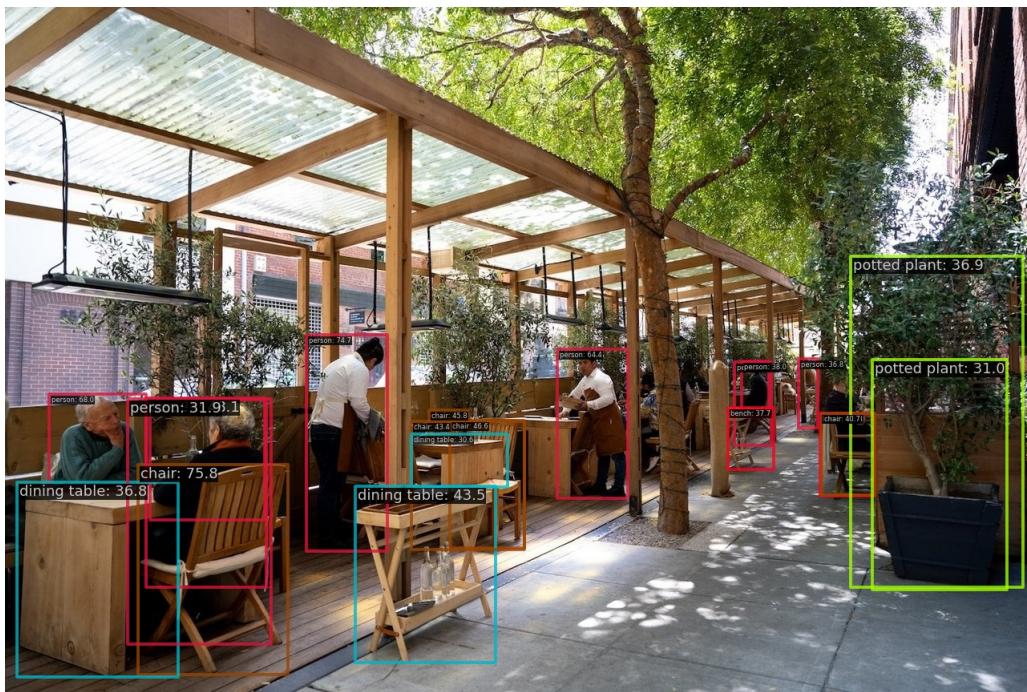
2.5 MMDetection

MMDetection là một opensource framework hỗ trợ bài toán object detection được phát triển bởi OpenMMLab[6].

So với các thư viện trước, MMDetection khá dễ cài đặt và sử dụng, chỉ cần theo hướng dẫn tại document và chạy thử nghiệm, kết quả trả về cho ta hình ảnh predict như hình 2.5



Hình 2.4: Kết quả Segmentation của GluonCV



Hình 2.5: Kết quả detection của MMDetection

3 Kết luận

Sau khi tìm hiểu và tiến hành thử nghiệm các thư viện, học viên nhận thấy

- Về mặt chức năng và tài liệu, GluonCV nổi trội hơn cả. Nó có nhiều chức năng và tài liệu hỗ trợ, theo khuôn khổ của Apache nên cấu trúc khoa học và dễ sử dụng. Các thư viện còn lại tương tự nhau.
- Về mặt cài đặt, cấu hình, MMDetection là dễ cài đặt nhất. Tensor và GluonCV khó hơn do xung đột phiên bản nhưng vẫn có thể cài đặt. Còn detectron2 thì khó dùng, chỉ sử dụng trên colab.
- Về tốc độ inference, nhìn chung MMDetection và GluonCV khá nhanh. Detectron2 thì không thể so sánh do dùng tài nguyên của colab không giống các thư viện còn lại. Tensor thì rất lâu cho mỗi hình

Tài liệu tham khảo

- [1] Apache Software Foundation. *Apache MXNet Tutorials*. 2020. URL: <https://cv.gluon.ai/tutorials/index.html> (visited on 02/05/2024).
- [2] Apache Software Foundation. *Gluon - Neural network building blocks*. 2020. URL: <https://mxnet.apache.org/versions/1.5.0/tutorials/gluon/gluon.html> (visited on 02/04/2024).
- [3] Kunihiko Fukushima. “A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biol, Cybern* 36 (1980), pp. 193–202.
- [4] Jian Guo et al. “GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing”. In: *Journal of Machine Learning Research* 21.23 (2020), pp. 1–7. URL: <http://jmlr.org/papers/v21/19-429.html>.
- [5] TensorFlow Hub. *Object Detection*. 2023. URL: https://www.tensorflow.org/hub/tutorials/object_detection (visited on 02/04/2024).
- [6] OpenMMLab. *OpenMMLab*. 2023. URL: <https://openmmlab.com/> (visited on 02/05/2024).
- [7] Facebook AI Research. *Detectron2 Beginner’s Tutorial*. 2024. URL: <https://colab.research.google.com/drive/1AAPNFDNcr3-ZwSrH8cs4KueViREVL4Q?usp=sharing> (visited on 02/04/2024).
- [8] sonnh-uit. *Apache MXNet Gluon CV Implementation*. 2024. URL: https://github.com/sonnh-uit/it2038_task2/blob/master/code/GluonCV/gluonCV.py (visited on 02/05/2024).
- [9] sonnh-uit. *TensorFlow Object Detection API Implementation*. 2024. URL: https://github.com/sonnh-uit/it2038_task2/blob/master/code/tensor/obj_detect_tensorflow.py (visited on 02/04/2024).
- [10] Yuxin Wu et al. *Detectron2*. <https://github.com/facebookresearch/detectron2>. 2019.