



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN: MÁY HỌC THỐNG KÊ
LỚP: 16CNTN

ĐỒ ÁN CUỐI KÌ

YOLO – VERSION 3

Người thực hiện:
Nguyễn Thanh Tuấn – 1612774

Giáo viên:
Ngô Minh Nhựt

Mục lục

I. Thông tin chung.....	3
II. Đánh giá hoàn thành đồ án.	3
III. Chi tiết các yêu cầu.....	3
1. Train Yolo nhận dạng đối tượng mới.....	3
1.1. Mô tả dữ liệu.....	3
1.2. Tranning đối tượng mới.	4
2. Xây dựng web app để detect đối tượng.....	5
2.1. Tổng quan về web và các chức năng chính.....	5
2.2. Hình ảnh minh họa web.....	6
III. Tài liệu tham khảo.....	7

I. Thông tin chung.

STT	Họ và tên	MSSV	Mail
1	Nguyễn Thanh Tuấn	1612774	1612774@student.hcmus.edu.vn

II. Đánh giá hoàn thành đồ án.

STT	Yêu cầu	Mức độ hoàn thành (tỉ lệ: %)
1	Chạy YOLOv3 với bộ weights sẵn có	100%
2	Train đối tượng nhận diện mới cho YOLOv3	100%
3	Xây dựng app (ứng dụng web) chạy YOLO để nhận diện đối tượng	100%

=> **Tổng quan:** Hoàn thành tất cả yêu cầu đồ án.

III. Chi tiết các yêu cầu.

1. Train YOLO nhận dạng đối tượng mới.

- Đối tượng lựa chọn: Con khỉ - monkey.

1.1. Mô tả dữ liệu.

- Quy mô dữ liệu: Dữ liệu gồm tập 1711 hình ảnh theo định dạng *.jpg. Các hình ảnh gồm nhiều tư thế và hoàn cảnh sống của các loài khỉ khác nhau để có tính tổng quát cao - có tầm khoảng gần 10 loài khỉ.

- Nguồn dữ liệu: Dữ liệu được thu thập từ nhiều nguồn khác nhau trên Internet - chủ yếu Google Images. Ngoài ra một số ảnh lấy từ các nguồn chia sẻ - github.

- Thư mục dữ liệu: Dữ liệu được đặt trong thư mục monkey_data_new (Thư mục con của thư mục Sources).

- Tiền xử lý dữ liệu:

+ Dữ liệu được chuẩn hóa tên file theo format sau 'monkey_SoThuTu.jpg' (SoThuTu tính từ 0), và một số ảnh theo format 'monkey_camera_SoThuTu.jpg' (SoThuTu tính từ 0).

Code - mã nguồn chuẩn hóa tên dữ liệu ảnh được đính kèm trong thư mục Sources (file python - rename_monkey_data.py).

+ Resize kích thước ảnh: Để thuận tiện cho việc gán nhãn, các dữ liệu ảnh được chuẩn hóa sao cho kích thước tối đa là 800x800. Code - mã nguồn resize ảnh được đính kèm trong thư mục Sources (file python - resize_all_image.py).

+ Gán nhãn: Các hình được gán nhãn - boundingBox đối tượng bằng tay sử dụng Yolo-Annotation-Tool. Nhãn - các file *.txt (tên tương ứng với tên file ảnh) được đặt cùng thư mục với thư mục chứa file ảnh. Cấu trúc file nhãn gồm:

- n dòng: tương ứng với n đối tượng trong ảnh.
- Mỗi đối tượng - mỗi dòng có cấu trúc sau: class_id x y width height. class_id là số thự tự của đối tượng trong file class (file *.names trong lúc train và detect) - số này tính từ 0; x, y lần lượt là hai tọa độ tâm của boundingBox với width, height là chiều rộng và chiều cao tương ứng của boundingBox.

Code - mã nguồn gán nhãn được đính kèm trong thư mục Sources (file python - monkey_main_new.py).

1.2. Tranning đối tượng mới.

- Công cụ: Đồ án sử dụng Google colab (<https://colab.research.google.com>) để train đối tượng mới.

- Phân chia dữ liệu train/test: Dữ liệu được chia thành 2 phần.

+ Phần 1: 9/10 bộ dữ liệu dùng làm tập train.

+ Phần 2: 1/10 dữ liệu dùng làm tập validation.

Tạo 2 file monkey_train.txt và monkey_test.txt là tập các đường dẫn tuyệt đối đến thư mục chứa bộ dữ liệu train. Code - mã nguồn phân chia dữ liệu được đính kèm trong thư mục Sources (file python - monkey_process_new.py).

- Bộ weights cuối cùng được sử dụng là weights 21000. Với các thông số loss tương ứng là, train loss 0.04-0.11, test loss 0.05-0.13.

- Chi tiết các bước thực hiện:

+ Chuẩn bị dữ liệu: Ảnh, nhãn, phân chia dữ liệu - đã trình bày ở trên.

+ Chuẩn bị các tập tin cấu hình:

- Thư mục: cfg/
 - File monkey.data: Là file chứa số lượng đối tượng - ở đây là 1, đường dẫn đến tập tin chứa đường dẫn của dữ liệu train và test và đường dẫn đến thư mục backup.
 - File monkey_yolov3.cfg: Tương tự file yolov3.cfg, có tùy chỉnh các thông số sau:
 - batch = 24: Số lượng ảnh load lên cho mỗi lần đọc train.

- filter = 18: Thay đổi ở 3 dòng 603, 689, 776, Được tính dựa trên công thức $(soClass + 5) * 3$, tức $18 = (1 + 5) * 3$.
- classes = 1: Số đối tượng train.
- Thư mục: data/
 - Thêm tập tin monkey.names: Chứa tên các đối tượng train (thứ tự này cần đúng với thứ tự của các classes lúc gán nhãn) - ở đây chứa 1 tên duy nhất là monkey.
- Thư mục: examples/
 - Tùy chỉnh thông số backup trong khi train:
 - Mặc định: Backup sau mỗi 100 lần đọc dữ liệu nếu nhỏ hơn 1000, sau đó backup sau mỗi 10000 lần.
 - Tùy chỉnh: Backup sau mỗi 100 lần đọc dữ liệu nếu nhỏ hơn 1000, sau đó backup sau mỗi 500 lần đọc dữ liệu.

+ Thực hiện biên dịch lại mã nguồn theo với các thông số đã chọn: bằng lệnh make.

+ Training: Training đối tượng mới cho yolov3 bằng thực thi lệnh:

***./darknet detector train cfg/monkey.data cfg/monkey_yolov3.cfg
darknet53.conv.74***

Với tập tin darknet53.conv.74 là tập weights ban đầu.

2. Xây dựng web app để detect đối tượng.

2.1. Tổng quan về web và các chức năng chính.

- Ngôn ngữ server: Python - python2 sử dụng Flask.
- Thư mục mã nguồn: Mã nguồn web được đặt trong thư mục web - thư mục con của thư mục Sources.
- Chạy server bằng cách thực thi tập tin python darknet.py trong thư mục web.
- Hoạt động chính: Dữ liệu - weights và cấu trúc mạng sẽ được load lên 1 lần lúc khởi động server - ở đây dùng localhost (port mặc định 5000). Muốn detect đối tượng trong ảnh ta upload ảnh lên server - ảnh upload được lưu ở thư mục static, chọn Detect Image để thực hiện detect các đối tượng có trong ảnh. Kết quả nhận được (hiển thị trên web) gồm có 1 box chứa hình ảnh có boundingBox có đánh kèm tên đối tượng và độ chính xác
- tin cậy, 1 box chứa danh sách các đối tượng có trong ảnh kèm theo độ chính xác (tỉ lệ %).

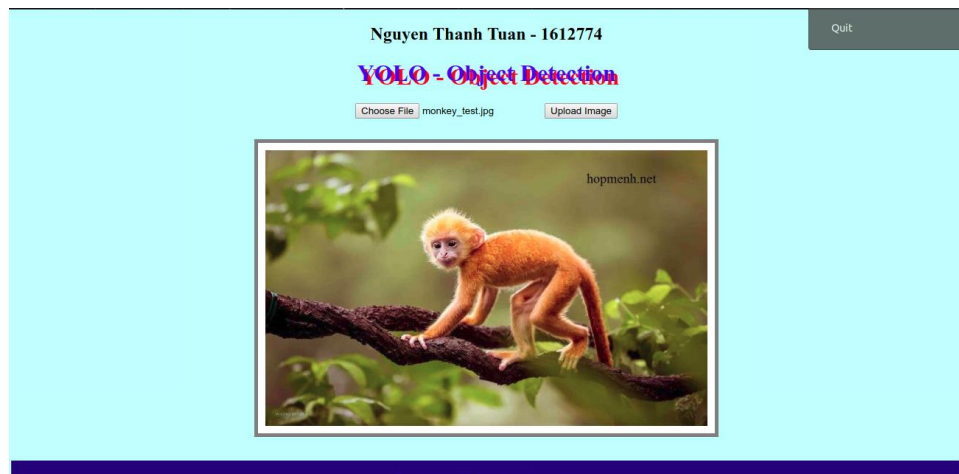
- Do sử dụng file python để detect nên lệnh detect chỉ trả về list các đối tượng được nhận dạng có trong ảnh - cấu trúc mỗi phần tử gồm tên, độ chính xác, 1 list gồm 4 thông số của boundingBox (x, y, width, height). Ta thực hiện boundingBox các đối tượng trong ảnh dựa theo các thông số nhận được - hàm thực hiện chính **def boundingBox(fileName, r)** trong tập tin darknet.py - sử dụng opencv để thực hiện.

- Cấu trúc web: Web gồm 3 trang chính:

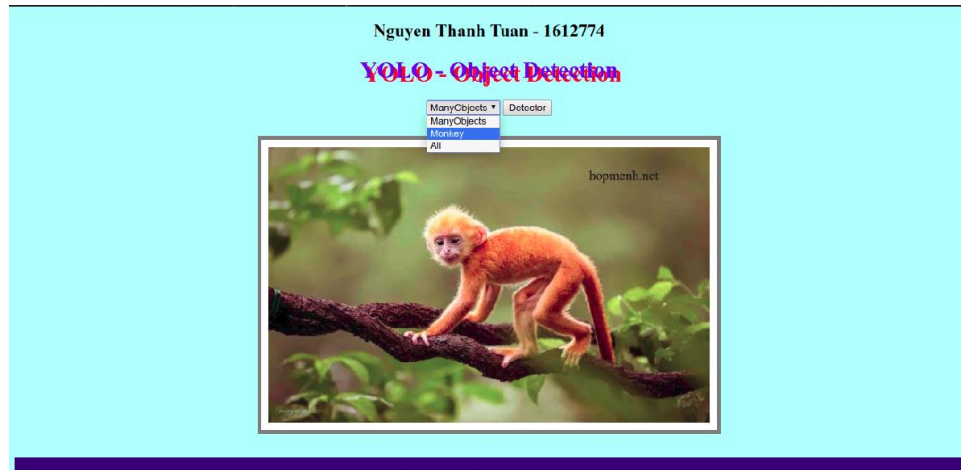
- Trang upload - trang chủ: Gồm 2 button chọn Choose File và Upload Image thực hiện 2 chức năng chính là lựa chọn file - ảnh cần detect và upload lên server. Nếu nhấn button Upload Image ảnh được chọn sẽ upload lên server và chuyển sang trang detect nếu upload thành công, ngược lại vẫn ở trang ban đầu.
- Trang detect: Chọn loại detect (gồm 3 loại manyObjects thực hiện nhận diện với yolov3 có sẵn, monkey thực hiện nhận diện đối tượng mới - con khỉ, all thực detect cả các đối tượng sẵn có của yolov3 và cả đối tượng mới là con khỉ). Nhấn button detect để thực hiện việc detect đối tượng trong ảnh và chuyển sang trang cuối cùng nếu thành công.
- Trang completed: Hiển thị kết quả detect lên màn hình: Gồm ảnh có nhãn đối tượng và danh sách các đối tượng.

2.2. Hình ảnh minh họa web.

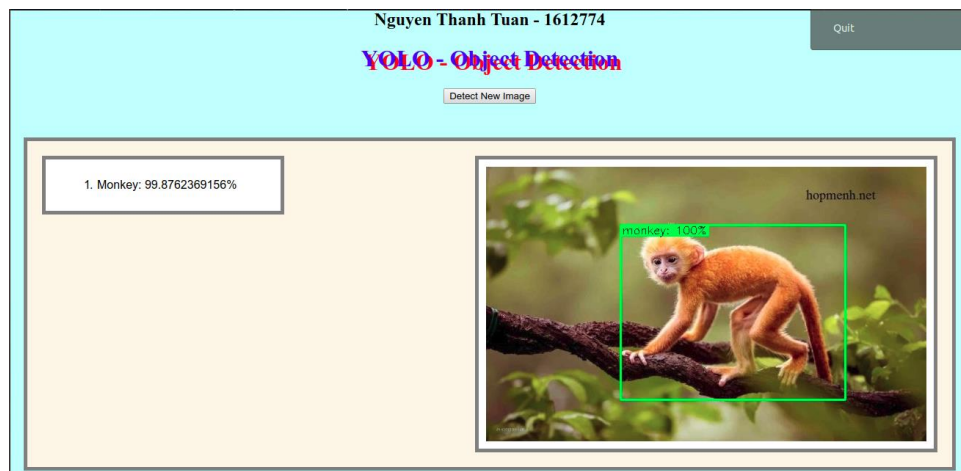
- Trang chủ:



- Trang detect:



- Trang hiển thị kết quả:



III. Tài liệu tham khảo.

[1]. YOLO: Real-Time Object Detection. <https://pjreddie.com/darknet/yolo/>