ĐỒ ÁN CUỐI KHÓA

NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

Giảng viên hướng dẫn:	TS. Lê Đình Duy
	TS. Nguyễn Tấn Trần Minh Khang
Học viên thực hiện:	Bùi Mạnh Toàn
	MSHV: CH1701007

Dạng đồ án:	Dạng 3 - Phát hiện đối tượng (object detection).
Phương pháp sử dụng:	YOLO (https://pjreddie.com/darknet/yolo/).
GITHUB đồ án:	https://github.com/buimanhtoanit/VRA
GITHUB bài tập:	https://github.com/buimanhtoanit/MATLAB
YOUTUBE:	

Mục tiêu

Mục tiêu đồ án cuối khóa theo dạng 3 được mô tả như sau:

- Cài đặt hệ thống YOLO.
- Chọn một đối tượng mới, train lại model.
- Đánh giá chất lượng model đã train trên file ảnh nằm ngoài tập train.

Nội dung

- Đồ án chọn loại đối tượng mới là trái thanh long chín.
- Dữ liệu train là hình ảnh trái thanh long tìm trên Google Images (50 hình). Dùng 10 hình test và 40 hình train.

Thực hiện chương trình

Cài đặt YOLO trên hệ điều hành Windows

Yêu cầu cài đặt các chương trình:

- Visual Studio 2015 (https://go.microsoft.com/fwlink/?LinkId=615448&clcid=0x409)
- CUDA 8.0 (https://developer.nvidia.com/cuda-downloads)
- OpenCV 3.2.0 vc14 (https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.2.0/opencv-3.2.0-vc14.exe/download)
- GPU CC >= 3.0

Cấu hình máy tính sử dụng để train:

- CPU: Intel Core i5-4460 3.2GHz
- RAM: 8GB
- GPU: GTX 1050 6033MB

Cài đặt darknet:

B1. Clone project darknet từ github:

git clone https://github.com/AlexeyAB/darknet.git

B2. Chỉnh sửa một số thông số môi trường cho phù hợp các chương trình đã cài đặt ở trên (OpenCV, CUDA) trong file \darknet\build\darknet\darknet.vcxproj

B3. Chạy file solution \darknet\build\darknet\darknet.sln

B4. Build Project với tùy chọn Release và x64 để có được chương trình darknet (\darknet\build\darknet\x64\darknet.exe).

Chuẩn bị dữ liệu

Cài đặt Yolo_mark (https://github.com/AlexeyAB/Yolo_mark)

Đây là phần mềm sử dụng để xác định bounding box cho các ảnh train. Phần mềm này phù hợp cho Yolo v2.

B1. Clone project Yolo_mark từ github:

git clone https://github.com/AlexeyAB/Yolo_mark.git

B2. Chỉnh sửa một số thông số môi trường cho phù hợp các chương trình đã cài đặt ở trên (OpenCV, CUDA) trong file \Yolo_mark\ yolo_mark.vcxproj

B3. Chạy file solution \Yolo_mark\yolo_mark.sln

B4. Build Project với tùy chọn Release và x64 để có được chương trình Yolo_mark (\Yolo_mark\x64\Release\yolo_mark.exe).

Xác định bounding box cho dữ liệu train

B1. Xóa tất cả file trong folder x64/Release/data/img.

B2. Copy file ảnh cần xác định bounding box vào folder này (x64/Release/data/img).

B3. Chỉnh số lượng đối tượng trong file x64/Release/data/obj.data

Do đồ án chỉ detect 1 đối tượng nên file obj.data có cấu trúc như sau:

classes= 1

train = data/train.txt

valid = data/test.txt

names = data/obj.names

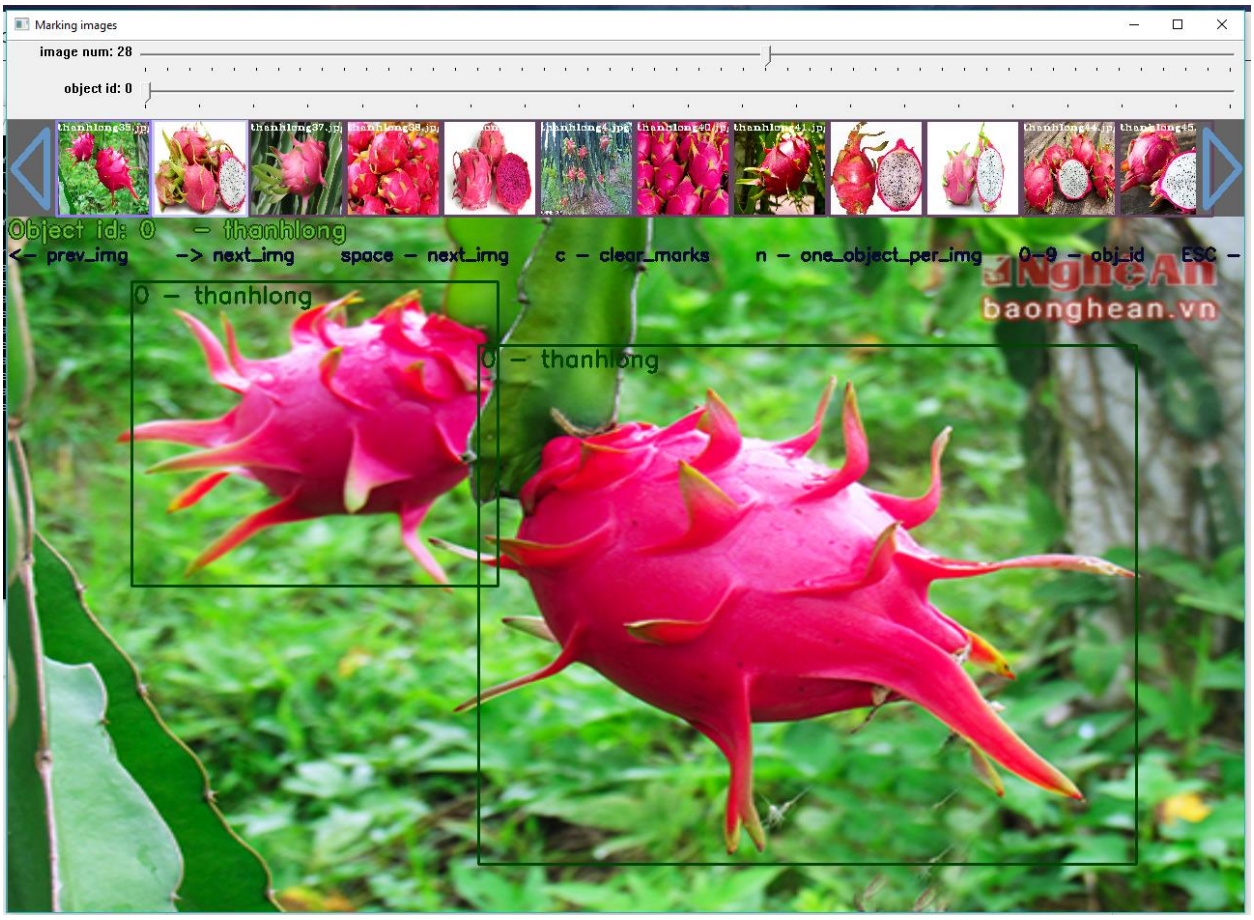
backup = backup/

B4. Đặt tên cho đối tượng vào file x64/Release/data/obj.names

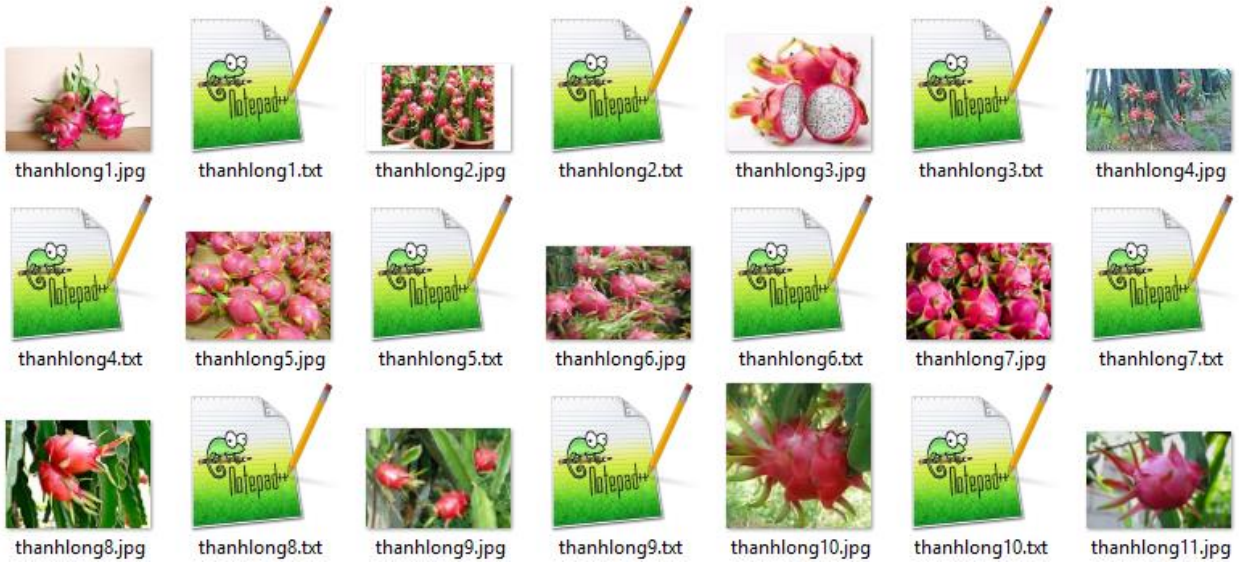
thanhlong

B5. Chạy ứng dụng bằng cách chạy file: x64\Release\yolo_mark.cmd

B6. Dữ liệu train gồm 50 hình. Tiến hành chọn đối tượng, ưu tiên chọn những trái nguyên vẹn, ít bị che khuất:



Chương trình sẽ tạo file txt dùng để lưu vị trí của bounding box, tên file trùng với tên ảnh:



Nội dung file:

0 0.373437 0.620139 0.384375 0.759722

0 0.733984 0.557639 0.332031 0.695833

[số thứ tự của object] [vị trí giữa của object theo chiều X] [vị trí giữa của object theo chiều Y] [kích thước của object theo chiều X] [kích thước của object theo chiều Y]

Thông tin dữ liệu train sẽ được lưu vào file \x64\Release\data\train.txt.

Tách dữ liệu trên thành 40 hình train và 10 hình test. 10 hình test được lưu đường dẫn vào file \x64\Release\data\test.txt (tên file này đã được cung cấp ở file obj.data).

Chuẩn bị các file config cho Yolo v2

obj.data
obj.names
yolo-thanhlong.cfg

Config cho YoloV2 bao gồm 3 file, 2 file obj.data và obj.names đã được tạo ở trên. File yolo-thanhlong.cfg được chỉnh sửa từ một config có sẵn là yolo-voc.cfg.

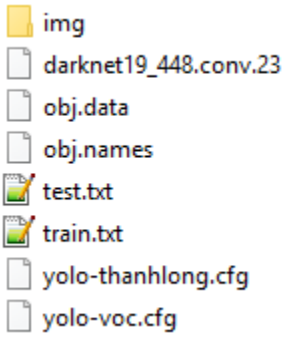
Nội dung chỉnh sửa gồm:

- batch=12 là sử dụng 12 trong mỗi bước trainning.
- subdivisions=12 là chia 12 của batch cho mỗi bước trainning. Các tùy chọn này được thiết lập tùy thuộc phần cứng máy tính (card đồ họa) để tránh tình trạng *CUDA out of memory error*.
- classes=1 là số lượng object muốn train.
- filters=(classes + 5)*5 là số lượng filter, ở đây là 30 do classes = 1.

Để bắt đầu train cần file convolutional layers có sẵn: 

.

Tổng hợp các thành phần cần thiết để train:

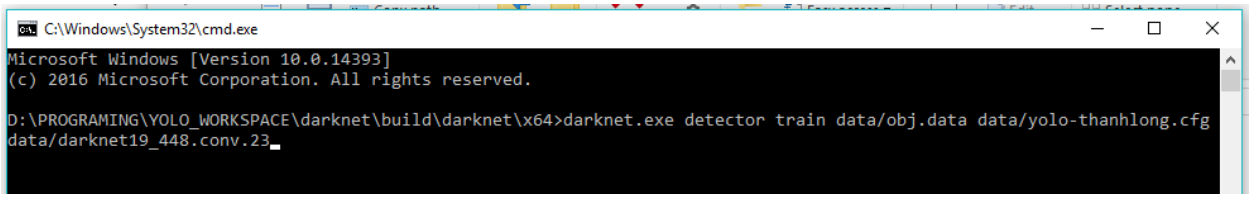


Trainning

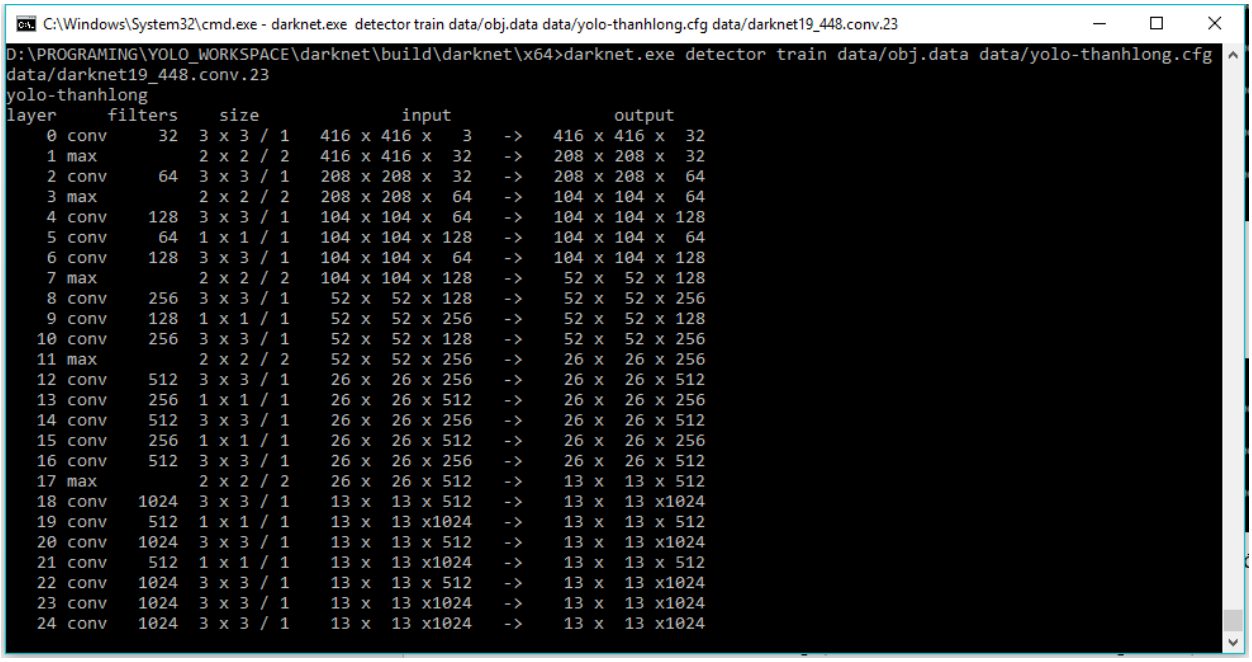
B1. Copy các file trong Yolo_mark\data vào \darknet\build\darknet\x64\data

B2. Chạy lệnh sau trên cmd để dùng darknet train dữ liệu (không xuống dòng):

darknet.exe detector train data/obj.data data/yolo-thanhlong.cfg data/darknet19_448.conv.23



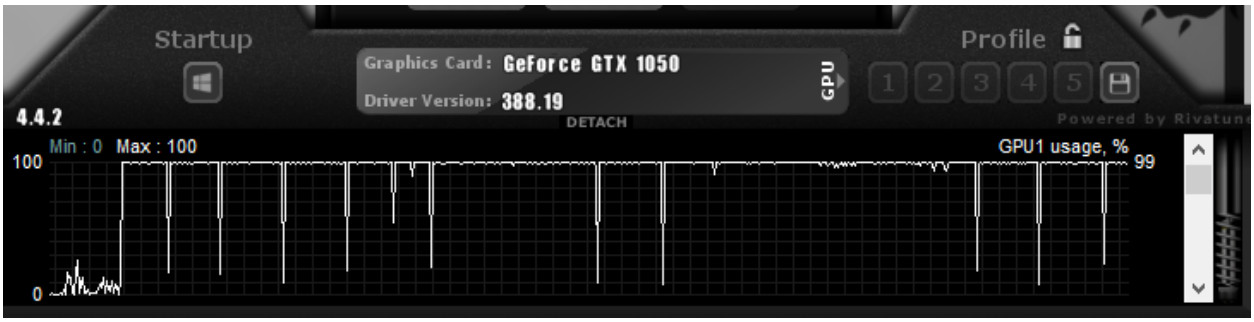
Màn hình load config ban đầu:



Chương trình sẽ resize ảnh và train theo file config đã dùng:

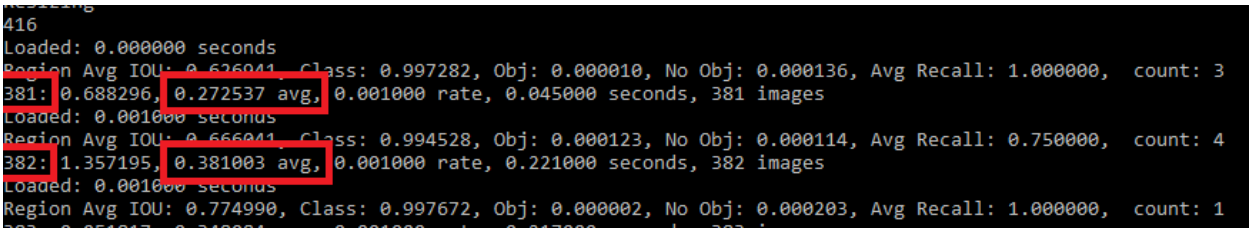
```
Region Avg IOU: 0.743007, Class: 0.993867, Obj: 0.000758, No Obj: 0.100028, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.487474, Class: 0.966773, Obj: 0.000188, No Obj: 0.100207, Avg Recall: 0.428571,  count: 7
Region Avg IOU: 0.601649, Class: 0.033333, Obj: 0.500000, No Obj: 0.100046, Avg Recall: 1.000000,  count: 1
60: 4.301883, 5.910082 avg, 0.001000 rate, 2.736000 seconds, 720 images
Resizing
576
Loaded: 0.000000 seconds
Region Avg IOU: 0.417845, Class: 0.516667, Obj: 0.250000, No Obj: 0.100133, Avg Recall: 0.000000,  count: 2
Region Avg IOU: 0.641800, Class: 0.033333, Obj: 0.500000, No Obj: 0.100167, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.665823, Class: 0.993856, Obj: 0.000167, No Obj: 0.100037, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.579868, Class: 0.930184, Obj: 0.000071, No Obj: 0.100292, Avg Recall: 0.833333,  count: 12
Region Avg IOU: -nan(ind), Class: -nan(ind), Obj: -nan(ind), No Obj: 0.100158, Avg Recall: -nan(ind),  count:
Region Avg IOU: 0.765035, Class: 0.033333, Obj: 0.500000, No Obj: 0.100058, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.548129, Class: 0.977438, Obj: 0.000014, No Obj: 0.100098, Avg Recall: 0.750000,  count: 12
Region Avg IOU: 0.853763, Class: 0.033333, Obj: 0.500000, No Obj: 0.100066, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.546104, Class: 0.964900, Obj: 0.000003, No Obj: 0.100027, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.517233, Class: 0.033333, Obj: 0.500000, No Obj: 0.100151, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.536640, Class: 0.740167, Obj: 0.125135, No Obj: 0.100101, Avg Recall: 0.500000,  count: 4
Region Avg IOU: 0.671563, Class: 0.962143, Obj: 0.000405, No Obj: 0.100041, Avg Recall: 1.000000,  count: 3
61: 6.595539, 5.978628 avg, 0.001000 rate, 3.342000 seconds, 732 images
Loaded: 0.001000 seconds
Region Avg IOU: 0.661787, Class: 0.999633, Obj: 0.000152, No Obj: 0.100104, Avg Recall: 1.000000,  count: 4
Region Avg IOU: 0.620130, Class: 0.033333, Obj: 0.500000, No Obj: 0.100049, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.579048, Class: 0.033333, Obj: 0.500000, No Obj: 0.100120, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.849632, Class: 0.033333, Obj: 0.500000, No Obj: 0.100056, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.880552, Class: 0.033333, Obj: 0.500000, No Obj: 0.100084, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.589091, Class: 0.033333, Obj: 0.500000, No Obj: 0.100097, Avg Recall: 0.500000,  count: 2
Region Avg IOU: 0.689837, Class: 0.033333, Obj: 0.500000, No Obj: 0.100035, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.958432, Class: 0.033333, Obj: 0.500000, No Obj: 0.100041, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.702773, Class: 0.990740, Obj: 0.000008, No Obj: 0.100103, Avg Recall: 1.000000,  count: 4
Region Avg IOU: 0.619680, Class: 0.937383, Obj: 0.003309, No Obj: 0.100213, Avg Recall: 1.000000,  count: 3
Region Avg IOU: 0.577953, Class: 0.033333, Obj: 0.500000, No Obj: 0.100053, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.617881, Class: 0.033333, Obj: 0.500000, No Obj: 0.100096, Avg Recall: 0.500000,  count: 2
62: 6.251188, 6.005884 avg, 0.001000 rate, 3.508000 seconds, 744 images
Loaded: 0.002000 seconds
Region Avg IOU: 0.375178, Class: 0.033333, Obj: 0.500000, No Obj: 0.100044, Avg Recall: 0.000000,  count: 1
Region Avg IOU: 0.407643, Class: 0.033333, Obj: 0.500000, No Obj: 0.100204, Avg Recall: 0.000000,  count: 1
Region Avg IOU: 0.651684, Class: 0.990364, Obj: 0.000439, No Obj: 0.100073, Avg Recall: 0.666667,  count: 3
Region Avg IOU: 0.853403, Class: 0.033333, Obj: 0.500000, No Obj: 0.100270, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.642598, Class: 0.516560, Obj: 0.250047, No Obj: 0.100045, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.683293, Class: 0.033333, Obj: 0.500000, No Obj: 0.100045, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.655580, Class: 0.910011, Obj: 0.000135, No Obj: 0.100170, Avg Recall: 0.666667,  count: 3
Region Avg IOU: 0.678205, Class: 0.976754, Obj: 0.000007, No Obj: 0.100086, Avg Recall: 1.000000,  count: 4
Region Avg IOU: 0.670601, Class: 0.994466, Obj: 0.000001, No Obj: 0.100037, Avg Recall: 1.000000,  count: 2
Region Avg IOU: 0.422418, Class: 0.033333, Obj: 0.500000, No Obj: 0.100126, Avg Recall: 0.000000,  count: 1
Region Avg IOU: 0.728253, Class: 0.033333, Obj: 0.500000, No Obj: 0.100103, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.661140, Class: 0.033333, Obj: 0.500000, No Obj: 0.100103, Avg Recall: 1.000000,  count: 1
63: 5.993156, 6.004611 avg, 0.001000 rate, 3.523000 seconds, 756 images
Loaded: 0.001000 seconds
Region Avg IOU: 0.532092, Class: 0.033333, Obj: 0.500000, No Obj: 0.100122, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.582864, Class: 0.992734, Obj: 0.000023, No Obj: 0.100034, Avg Recall: 0.500000,  count: 4
Region Avg IOU: 0.768292, Class: 0.033333, Obj: 0.500000, No Obj: 0.100043, Avg Recall: 1.000000,  count: 1
Region Avg IOU: 0.539180, Class: 0.999452, Obj: 0.000006, No Obj: 0.100050, Avg Recall: 0.750000,  count: 4
Region Avg IOU: 0.671797, Class: 0.997370, Obj: 0.000165, No Obj: 0.100095, Avg Recall: 1.000000,  count: 4
Region Avg IOU: 0.600714, Class: 0.999903, Obj: 0.000099, No Obj: 0.100067, Avg Recall: 0.500000,  count: 2
Region Avg IOU: 0.494478, Class: 0.033333, Obj: 0.500000, No Obj: 0.100302, Avg Recall: 0.000000,  count: 1
```

GPU Usage khi train



Khi nào thì dừng train?

Darknet đưa ra rất nhiều thông số quan trọng của mỗi bước train:



Tuy nhiên để xác định khi nào cần dừng train ta lưu ý đến 2 thông số được bôi ở trên:

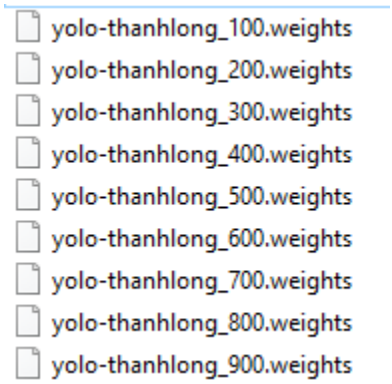
- **381** – Số lần lặp (dựa trên số batch mà ta thiết lập ở trên)
- **0.272537 avg** – Giá trị lỗi trung bình – **Càng nhỏ càng tốt**

Theo đề xuất của tác giả thì khoảng 0.0x avg thì có thể dừng train, hoặc nếu giá trị này không còn giảm nữa qua các vòng lặp.

Hoặc chương trình sẽ dừng ở 45000 lần lặp.

Để dừng train ta chỉ cần tắt chương trình.

Dữ liệu trọng số .weights sẽ được lưu vào \darknet\build\darknet\x64\backup



Test model

Để test hình data/test/test_thanhlong.jpg (là hình không nằm trong 50 hình train), với model đã train backup/yolo-thanhlong_1000.weights Chạy lệnh:

darknet.exe detector test data/obj.data data/yolo-thanhlong.cfg backup/yolo-thanhlong_1000.weights data/test/test_thanhlong.jpg