

ỨNG DỤNG THỊ GIÁC MÁY TÍNH

NỘP ĐỒ ÁN LẦN 2 - BÁO CÁO TIẾN ĐỘ

ĐỀ TÀI: Nhận dạng logo các thương hiệu thức uống tại Việt Nam

1. Thông tin nhóm

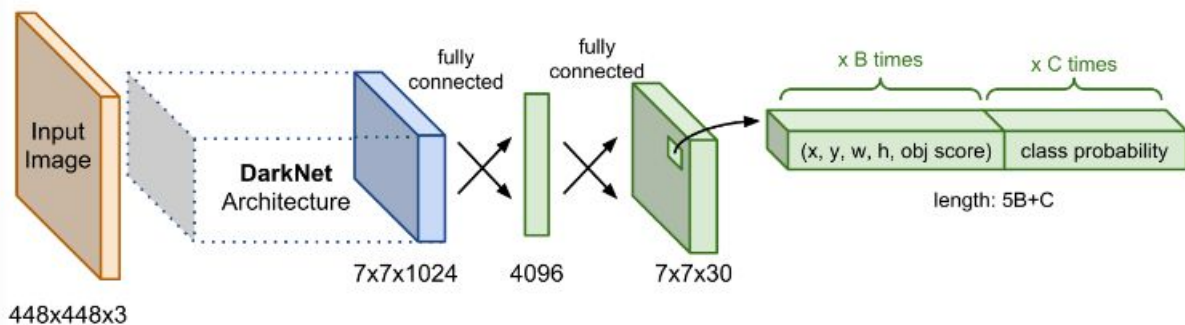
MSSV	HỌ TÊN	EMAIL
1612374	Trần Quang Minh	minhtranquangcr@gmail.com
1612899	Hoàng Xuân Trường	hoangxuantruong98@gmail.com

2. Nội dung báo cáo

2.1. Tìm hiểu thuật toán YOLO

Mô hình YOLO (“You only look one”), là 1 trong những bài báo đầu tiên về nhận diện vật thể trong ảnh với ưu điểm là tốc độ thực thi vượt trội so với các thuật toán khác.

Kiến trúc của mô hình:



- Input đầu vào là tấm ảnh với kích thước (448x448x3), ảnh này sẽ được chia nhỏ thành các ô lưới, bao gồm SxS ô lưới (S được chọn bằng 7 trong paper).
- Ảnh đầu vào được đưa vào một **pre-trained darknet** (khá tương tự GoogLeNet).
- Sau khi đưa vào pre-trained, feature được encode thành 1 vector 4096 chiều.
- Cuối cùng, vector encode này được upsample thành 1 tensor 7x7x30 thể hiện các giá trị dự đoán. Mỗi một vector 1x30 trong tensor trên thể hiện 1 dự đoán đối với 1 ô lưới trong ảnh đầu vào.

Các giá trị mà mô hình dự đoán:

Vector 30 chiều này được tách ra như sau: $30 = 5B + C$ ($B = 2, C = 20$), trong đó:

- B là số lượng bounding box có trong ô lưới, 5 ở đây bao gồm 4 giá trị tọa độ của bounding box (x, y, w, h - x, y ở đây là tọa độ tâm bounding box cùng với 1 độ tự tin là bounding box này thuộc trong ô lưới).
- C ở đây là số lượng loại object (số class) mà mô hình có thể dự đoán, ứng với 1 vector phân phối C chiều trên C class.

Các giá trị ground truth:

Ứng với mỗi bounding box được gán nhãn trong ảnh sẽ là 1 bộ (x, y, w, h - x, y là tọa độ tâm bounding box).

Hàm lỗi:

Hàm lỗi dùng để tính toán độ sai biệt giữa các giá trị dự đoán và các giá trị ground truth được nêu ra ở trên, tối ưu hàm này với tham số là các tham số (parameter) của mô hình. Gồm các hàm lỗi sau đây:

a. Hàm lỗi về độ lệch các bounding box đôi 1 trên từng B trong mỗi ô lưới với B ground truth (tính trên 4 giá trị x, y, w, h).

$$\mathcal{L}_{\text{loc}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

b. Hàm lỗi về confidence score và class distribution.

$$\mathcal{L}_{\text{cls}} = \sum_{i=0}^{S^2} \sum_{j=0}^B (1_{ij}^{\text{obj}} + \lambda_{\text{noobj}}(1 - 1_{ij}^{\text{obj}}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \in \mathcal{C}} 1_i^{\text{obj}} (p_i(c) - \hat{p}_i(c))^2$$

Hàm lỗi cuối cùng là tổng của 2 hàm lỗi trên.

YOLO cho thấy điểm mạnh của mình là 1 mô hình nhất quán từ đầu đến đuôi, dễ dàng thực thi, cũng như với các mô hình phiên bản nhẹ hơn phù hợp cho triển khai trên ứng dụng. Với ý tưởng chính như trên, YOLOv2 và YOLOv3 ra đời với các cải tiến dựa trên 1 vài ý tưởng từ thuật toán khác như (generate anchor box) cùng với các mô hình backbone mạnh mẽ khác ra đời trong những năm sau để làm pre-trained cho darknet.

2.2. Thu thập và xử lý dữ liệu bước đầu.

2.2.1. Thu thập dữ liệu

Để thuận tiện cho việc lấy hình ảnh từ trên mạng về, chủ yếu từ google search photo, script được dùng để lấy các link ảnh sau đó dùng tool để tải ảnh từ các link đó. Cụ thể là sử dụng tool uGet trên hệ điều hành Ubuntu. Một vài nhãn hiệu đã được thu thập:

- Aquafina
- Lavie

Nội dung script như sau:

```
1. var cont=document.getElementsByTagName("body")[0];
2. var imgs=document.getElementsByTagName("a");
3. var i=0;var divv= document.createElement("div");
4. var array=new Array();var j=-1;
5. while(++i<imgs.length){
6.   if(imgs[i].href.indexOf("/imgres?imgurl=http")>0){
7.     divv.appendChild(document.createElement("br"));
8.
9.     array[++j]=decodeURIComponent(imgs[i].href).split("/=|%|&/")[1]
10.    .split("?imgref")[0];
11.   }
12.   cont.insertBefore(divv,cont.childNodes[0]);
```

Ý tưởng chung của đoạn script là tìm hết các element là img trên website đang cần lấy, sau đó lấy nguồn/link ảnh là src của tag img.

Hiện tại, do mỗi nhãn hiệu đã tải về và sau khi lọc các ảnh chất lượng, đủ rõ nét, sau đó các phép augmentation được áp dụng cho dữ liệu để tăng tính tổng quát cũng như gia tăng lượng ảnh trong dataset:

- Flip ảnh theo chiều ngang
- Random chỉnh sửa kích thước của ảnh dựa trên ảnh mẫu, resize kích thước từ 70%-90% so với ảnh gốc.

Sau cùng, nhóm hiện đang có 546 ảnh cho 2 class, và vẫn đang tiếp tục thu thập thêm.

2.2.2. Tổ chức dữ liệu

Để thuận tiện cho việc gán nhãn, nhóm sử dụng “**Yolo annotation tool new**” để làm tool cho việc gán nhãn thủ công và có thể tự động chuyển đổi sang mà yolov3 dùng để training.

Khi tạo tập dữ liệu là mỗi tấm ảnh dạng .jpg sẽ có một file .txt có cùng tên và để cùng thư mục:

- File .jpg: là file ảnh. Trong trường hợp này là logo nhãn hiệu nước uống.
- File .txt: được tạo ra nhờ annotation tool được đề cập ở trên, là file chứa các thông tin annotation của logo trong tấm ảnh. Thông tin của một tấm ảnh thể hiện trong file .txt như sau: <object-class> <x> <y> <width> <height>. Trong đó:
 - + <object-class> là một số nguyên của object bắt đầu từ 0.
 - + <x><y> <width> <height> là các số thập phân có giá trị trong khoảng [0.0, 1.0] thể hiện vị trí của logo trong tấm hình.
 - + Với $\text{<x> = <absolute_x> / <image_width>}$ và $\text{<height> = <absolute_height> / <image_height>}$.

Ví dụ:



Với tấm hình trên, ta có file chú thích tương ứng đối với vị trí logo trong tấm ảnh, cụ thể nội dung của file này như sau:

```
0 0.45125 0.5902255639097744 0.5525 0.39849624060150374
```

Đường dẫn đến dữ liệu đã thu thập và xử lý của nhóm:

<https://drive.google.com/open?id=17RQIblUug77IlepbeN9vFdL94Al89Ewc>

3. Các công việc tiếp theo

- Khảo sát thêm các thuật toán detect khác.
- Tiếp tục tìm kiếm thêm dữ liệu về số lượng mỗi class, đồng thời tìm thêm các class mới.
- Tiến hành training dữ liệu sau khi kết thúc khảo sát.

4. Tài liệu tham khảo

- <https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html>
- <https://github.com/ManivannanMurugavel/Yolo-Annotation-Tool-New->
- https://medium.com/@manivannan_data/yolo-annotation-tool-new-18c7847a2186
- https://medium.com/@manivannan_data/how-to-train-yolov3-to-detect-custom-objects-ccbcafeb13d2
- https://colab.research.google.com/drive/1ITGZsfMaGUpBG4inDIQwIJVW476ibXk_#scrollTo=wkzMqLZV-rF5%5C