



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

**Khoa Khoa Học Máy Tính**  
**MÔN MÁY HỌC**

**Đồ án cuối kỳ**

**PHÁT HIỆN MỘT SỐ PHƯƠNG TIỆN GIAO THÔNG TRÊN ĐƯỜNG**

**Thành Viên nhóm :**

Nguyễn Tiến Công - 17520299

Huỳnh Trọng Khoa – 18520918

Nguyễn Tấn Phúc - 18521259

**Giảng viên hướng dẫn :**

ThS. Phạm Nguyễn Trường An

PGS TS. Lê Đình Duy

**Mã lớp:** CS114.L11.KHCL

**Tháng 1/2021**

## Mục lục

<b>I. GIỚI THIỆU</b>	3
1.1. Giới thiệu bài toán	3
1.2. Nền tảng chương trình	4
<b>II. MÔ TẢ DỮ LIỆU</b>	4
2.1. Thu thập dữ liệu	4
2.2. Tiền xử lý dữ liệu	4
<b>III. PHƯƠNG PHÁP</b>	5
<b>IV. HUẤN LUYỆN MÔ HÌNH</b>	8
<b>V. ĐÁNH GIÁ, KẾT LUẬN</b>	9
<b>VI. HƯỚNG PHÁT TRIỂN</b>	11

# I. GIỚI THIỆU

## 1.1. Giới thiệu bài toán

Nhận diện vật thể (Object Detection) là một bài toán thuộc lĩnh vực thị giác máy tính có nhiệm vụ xác định các đối tượng có trong ảnh. Input của bài toán này là ảnh có chứa 1 hay nhiều đối tượng quan tâm. Output là các vị trí của các đối tượng và nhãn của đối tượng đó.

Đối với đề tài “**Phát hiện một số phương tiện giao thông trên đường**” thì:

- **INPUT:** Một tấm ảnh có chứa xe cộ trên đường.



- **OUTPUT:** Bao gồm 2 danh sách là “Nhãn” và “Vị trí (toạ độ)”. Sau đó sử dụng thư viện để đưa ra ảnh đã được dự đoán gán nhãn.



*Ảnh 2. Minh họa Output*

## 1.2. Nền tảng chương trình

Chương trình sử dụng một số thư viện hỗ trợ cho việc hiển thị cũng như giúp huấn luyện mô hình như: Torch, TorchVision, numpy, Detectron2,... Các thư viện sử dụng được hỗ trợ trên ngôn ngữ Python.

# II. MÔ TẢ DỮ LIỆU

## 2.1. Thu thập dữ liệu

Dữ liệu được thu thập thông qua việc quay video có bối cảnh là trên đường (được chụp từ vị trí là trên cầu bộ hành) và có xe cộ lưu thông qua lại.

Tổng cộng đã quay **12** video ngắn và đưa ra được **1,465** ảnh cho bộ dữ liệu.

Bộ dữ liệu được chia theo tỉ lệ **7:3** cho Training (**1,025** ảnh) và Test (**440**).

Với số lượng ảnh trên thì số lượng mẫu cho mỗi lớp là:

ID [Class]	Số lượng mẫu train	Số lượng mẫu test
0 Pedestrian	551	244
1 Car	1700	705

2 Van	252	115
3 Bus	323	133
4 Truck	1428	644
5 Motor	6611	2890
6 Bicycle	65	19
7 Tricycle	10	1
8 Others	0	0

## 2.2. Tiền xử lý dữ liệu

Sử dụng công cụ **Dark Label** cho việc gán nhãn.

- **B1.** Lưu lại video dưới dạng hình ảnh theo từng frame.
- **B2.** Vẽ boundingBox và gán nhãn cụ thể cho các phương tiện giao thông có trong ảnh.
- **B3.** Lưu lại kết quả vừa làm mỗi ảnh dưới dạng **\*.xml**. File này sẽ chứa các tọa độ boundingBox và class tương ứng của các vật thể đã được gán nhãn trong ảnh.
- **B4.** Tiếp theo là chuyển thông tin tọa độ BBox và class thành **file \*.txt**.
- **B5.** Chuyển tiếp file **\*.txt** thành **\*.json** cho quá trình huấn luyện dữ liệu.

## III. PHƯƠNG PHÁP

Sử dụng thuật toán Faster R-CNN trên Detectron2.

Faster R-CNN nằm trong nhóm thuật toán two-stage object detection. Trong stage-1, ảnh sẽ được đưa ra 1 sub-network gọi là RPN (Region Proposal Network) với nhiệm vụ trích xuất các vùng trên ảnh có khả năng chứa đối tượng dựa vào các anchor. Sau khi đã thu được các vùng đặc trưng từ RPN, model Faster-RCNN sẽ thực hiện tiếp việc phân loại đối tượng và xác định vị trí nhờ vào việc chia làm 2 nhánh tại phần cuối của mô hình (Object classification & Bounding box regression)

Kiến trúc mô hình đã được cải thiện hơn nữa về cả tốc độ huấn luyện và phát hiện được đề xuất bởi Shaoqing Ren và các cộng sự tại Microsoft Research trong bài báo năm 2016

có tiêu đề Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Dịch nghĩa là “Faster R-CNN: Hướng tới phát hiện đối tượng theo thời gian thực với các mạng đề xuất khu vực”.

Kiến trúc này mang lại độ chính xác cao nhất đạt được trên cả hai nhiệm vụ phát hiện và nhận dạng đối tượng tại các cuộc thi ILSVRC-2015 và MS COCO-2015.

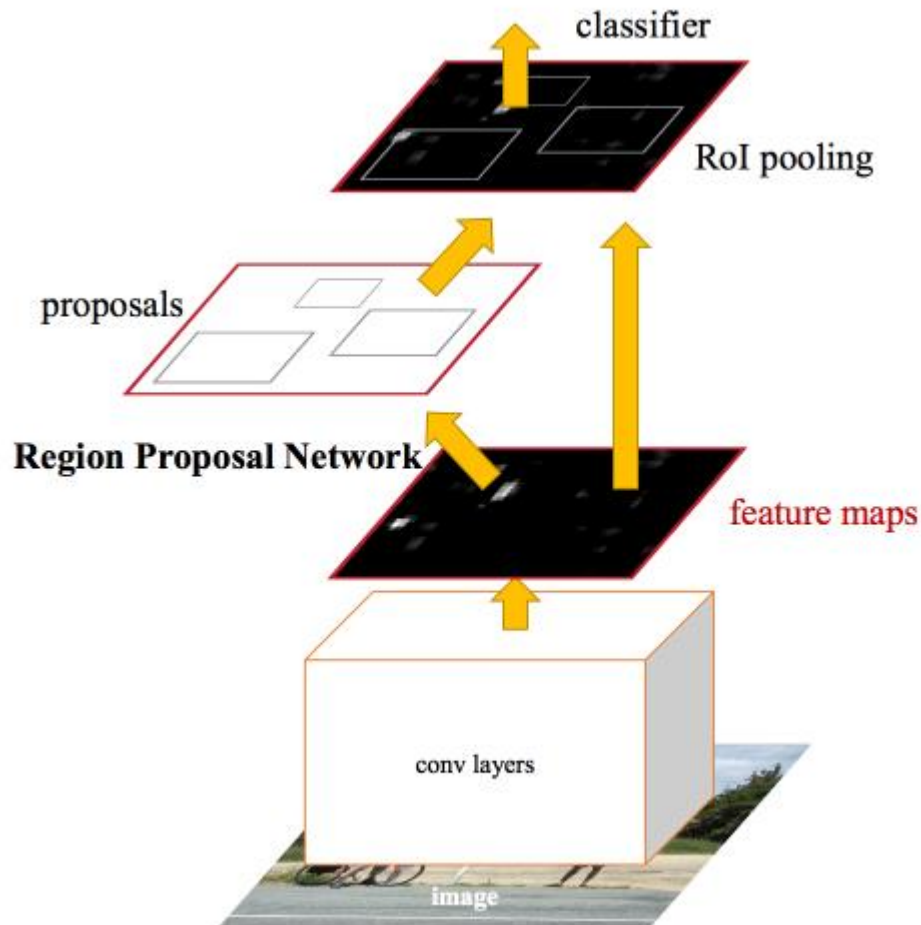
Kiến trúc được thiết kế để đề xuất và tinh chỉnh các region proposals như là một phần của quá trình huấn luyện, được gọi là mạng đề xuất khu vực (Region Proposal Network), hoặc RPN. Các vùng này sau đó được sử dụng cùng với mô hình Fast R-CNN trong một thiết kế mô hình duy nhất. Những cải tiến này vừa làm giảm số lượng region proposal vừa tăng tốc hoạt động trong thời gian thử nghiệm mô hình lên gần thời gian thực với hiệu suất tốt nhất. Tốc độ là 5fps trên một GPU.

Mặc dù là một mô hình đơn lẻ duy nhất, kiến trúc này là kết hợp của hai modules:

- Mạng đề xuất khu vực (Region Proposal Network, viết tắt là RPN). Mạng CNN để đề xuất các vùng và loại đối tượng cần xem xét trong vùng.
- Fast R-CNN: Mạng CNN để trích xuất các features từ các region proposal và trả ra các bounding box và nhãn.

Cả hai modules hoạt động trên cùng một output của một mạng deep CNN. Mạng RPN hoạt động như một cơ chế attention cho mạng Fast R-CNN, thông báo cho mạng thứ hai về nơi cần xem hoặc chú ý.

Kiến trúc của mô hình được tổng kết thông qua sơ đồ bên dưới:



**Hình 4:** Kiến trúc mô hình Faster R-CNN (được trích xuất từ bài báo gốc). Ở giai đoạn sớm sử dụng một mạng deep CNN để tạo ra một feature map. Khác với Fast R-CNN, kiến trúc này không tạo RoI ngay trên feature map mà sử dụng feature map làm đầu vào để xác định các region proposal thông qua một RPN network. Đồng thời feature maps cũng là đầu vào cho classifier nhằm phân loại các vật thể của region proposal xác định được từ RPN network.

RPN hoạt động bằng cách lấy đầu ra của một mạng pretrained deep CNN, chẳng hạn như VGG-16, và truyền feature map vào một mạng nhỏ và đưa ra nhiều region proposals và nhãn dự đoán cho chúng. Region proposals là các bounding boxes, dựa trên các anchor boxes hoặc hình dạng được xác định trước được thiết kế để tăng tốc và cải thiện khả năng đề xuất vùng. Dự đoán của nhãn được thể hiện dưới dạng nhị phân cho biết region proposal có xuất hiện vật thể hoặc không.

Một quy trình huấn luyện xen kẽ được sử dụng trong đó cả hai mạng con được đào tạo cùng một lúc. Điều này cho phép các tham số trong feature detector của deep CNN được tinh chỉnh cho cả hai tác vụ cùng một lúc.



Tại thời điểm viết, kiến trúc Faster R-CNN này là đỉnh cao của họ model R-CNN và tiếp tục đạt được kết quả gần như tốt nhất trong các nhiệm vụ nhận diện đối tượng. Một mô hình mở rộng hỗ trợ cho phân đoạn hình ảnh, được mô tả trong bài báo năm 2017 có tựa đề Mask R-CNN.

Detectron2 là một framework để xây dựng bài toán Object Detection and Segmentation. Được phát triển bởi nhóm Facebook Research. Detectron2 sử dụng Pytorch.

Các bước thực hiện sau khi tiền xử lý dữ liệu để lưu thành định dạng JSON:

B1. Đăng ký bộ dữ liệu vừa tạo cho detectron2.

```
import json
from detectron2.structures import BoxMode

def get_traffic_dicts(imgdir):
    json_file = imgdir+'/data.json'
    with open(json_file) as f:
        dataset_dicts = json.load(f)
    for i in dataset_dicts:
        filename=i['file_name']
        i['file_name'] = imgdir+"/"+filename
        for j in i['annotations']:
            j['bbox_mode'] = BoxMode.XYWH_ABS
            j['category_id'] = int(j['category_id'])
    return dataset_dicts
from detectron2.data import DatasetCatalog, MetadataCatalog
for d in ['Train','Test']:
    DatasetCatalog.register('traffic_'+d, lambda d=d: get_traffic_dicts('/content/Images/'+d))
    MetadataCatalog.get('traffic_'+d).set(thing_classes=['pedestrian', 'car', 'van', 'bus', 'truck', 'motor', 'bicycle', 'tricycle', 'other'])
traffic_metadata = MetadataCatalog.get('traffic_Train')
```

B2. Thực hiện bước huấn luyện mô hình [Phần IV]

B3. Đánh giá

```
from detectron2.evaluation import COCOEvaluator, inference_on_dataset
from detectron2.data import build_detection_test_loader
evaluator = COCOEvaluator("traffic_Test", cfg, False, output_dir="/output/")
test_loader = build_detection_test_loader(cfg, "traffic_Test")
inference_on_dataset(predictor.model, test_loader, evaluator)
```

## IV. HUẤN LUYỆN MÔ HÌNH

Mô hình được sử dụng ở đây là **Faster R-CNN R50-C4**

Thiết lập bộ config:

```
cfg.DATALOADER.NUM_WORKERS = 8
cfg.SOLVER.IMS_PER_BATCH = 4
```



```
cfg.SOLVER.BASE_LR = 0.0125
cfg.SOLVER.MAX_ITER = 1500
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 9
cfg.TEST.EVAL_PERIOD = 500
```

Load weight và đặt ngưỡng (threshold) cho confident score của các output đầu ra của mô hình, ở đây chúng em đặt ra 2 ngưỡng để so sánh :

**Trường hợp:** `cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.9`

**Trường hợp:** `cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.1`

## V. ĐÁNH GIÁ, KẾT LUẬN

- Trong quá trình training thì thường xảy ra một số lỗi như 1 vật thể nhưng xác định thành 2 vật thể (có 2 bounding box và nhãn lồng nhau) làm cho quá trình đánh giá bị sai, vì vậy mức ngưỡng khi đặt mở mức 0.9 sẽ cho kết quả tốt hơn.



*“Một số lỗi phát hiện sai, nhầm nhãn ”*

- Ở trường hợp đặt ngưỡng là 0.9: Độ AP@IoU = 50 ở mức cao tương đối là **84.8%** trong quá trình nhận dạng có thể có sai sót, thường là xe bus và van.

AP	AP50	AP75	APs	APm	APl
63.332	84.800	75.602	21.228	51.794	75.336

[01/22 01:49:35 d2.evaluation.coco\_evaluation]: Per-category bbox AP:

category	AP	category	AP	category	AP
pedestrian	49.141	car	65.729	van	59.060
bus	67.654	truck	75.312	motor	59.342
bicycle	40.422	tricycle	90.000	other	nan

- Ở trường hợp đạt ngưỡng là 0.1: Tuy độ  $AP@IoU=50$  có cao nhưng vì ở trường hợp này mô hình sẽ tính các boundingbox mà nó nhận diện được (bao gồm nhãn đúng và sai).

[01/22 01:53:03 d2.evaluation.coco\_evaluation]: Evaluation results for bbox:

AP	AP50	AP75	APs	APm	APl
69.018	94.199	81.123	28.126	59.796	79.097

[01/22 01:53:03 d2.evaluation.coco\_evaluation]: Per-category bbox AP:

category	AP	category	AP	category	AP
pedestrian	51.196	car	67.349	van	70.426
bus	82.069	truck	77.097	motor	60.748
bicycle	53.257	tricycle	90.000	other	nan

Mô hình hiện có thể nhận diện ảnh bất kì .

```
im = cv2.imread('/content/00000092.png')
outputs = predictor(im) # format is documented at https://detectron2.readthedocs.io/tutorials/models.html#model-output-format
v = Visualizer(im[:, :, :-1],
               metadata=traffic_metadata,
               scale=0.5,
               instance_mode=ColorMode.IMAGE # remove the colors of unsegmented pixels. This option is only available for segmentation models
)
out = v.draw_instance_predictions(outputs["instances"].to("cpu"))
cv2.imshow(out.get_image()[:, :, :-1])
```



## VI. HƯỚNG PHÁT TRIỂN

Với tính chất đề tài là phát hiện vật thể với chủ đề giao thông thì có thể phát triển đồ án hơn là về đếm số lượng xe lưu thông của 1 hoặc nhiều chiều dựa trên chuyển động của đối tượng.

## VII. THAM KHẢO

[1] Phạm Đình Khanh, “Các thuật toán object detection”. [Trực tuyến] Available: [Khoa học dữ liệu \(phamdingkhanh.github.io\)](https://phamdingkhanh.github.io) [Truy cập ngày 28 tháng 1 năm 2021].

[2] Nguyen Viet Hoai, “Face Detection on Custom Database using Detectron2 in Google Colab”. [Trực tuyến] Available: [Face Detection on Custom Dataset using Detectron2 in Google Colab \(viblo.asia\)](https://viblo.asia) [Truy cập ngày 26 tháng 1 năm 2021].

