**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH**

Thực hiện: Nhóm 13

20C12007 – Trần Đình Lâm

20C11035 – Trương Thế Kiệt

20C11040 – Đặng Nhật Minh

20C11034 – Nguyễn Trung Kiên

BÁO CÁO BÀI BÁO KHOA HỌC

NĂM HỌC 2020-2021

**TRÍ TUỆ NHÂN TẠO nâng cao**

**ĐẠI HỌC QUỐC GIA TPHCM**

**BẢNG THÔNG TIN CHI TIẾT NHÓM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã nhóm:** | **13** | | |
| **Tên nhóm:** | **K2014** | | |
| **Số lượng:** | **4** | | |
| **MSSV** | **Họ tên** | **Email** | **Điện thoại** |
| 20C12007 | Trần Đình Lâm | tdlam123@gmail.com | 0383522356 |
| 20C11035 | Trương Thế Kiệt | truongthekiet709@gmail.com |  |
| 20C11040 | Đặng Nhật Minh | minhdangnhat685@gmail.com |  |
| 20C11034 | Nguyễn Trung Kiên | ngkien1530@gmail.com |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **BẢNG PHÂN CÔNG & ĐÁNH GIÁ HOÀN THÀNH CÔNG VIỆC** | | | |
| **Người thực hiện** | **Công việc thực hiện** | **Mức độ hoàn thành** | **Đánh giá của nhóm** |
| 20C12007  Trần Đình Lâm | Phân công các công việc cần thực hiện | 70% | 7/10 |
| Dịch bài báo sang tiếng Việt và giải thích các thuật ngữ |
| Phối hợp với Minh viết báo cáo |
| Tìm hiểu và giải thích mô hình mạng CNN được đề xuất |
| 20C11035  Trương Thế Kiệt | Tổng hợp tổng quan chủ đề của bài báo | 70% | 7/10 |
| Phân tích kết quả thực nghiệm của nhóm tác giả |
| Soạn slide cho buổi serminar |
| 20C11040  Đặng Nhật Minh | Phân tích các bộ dataset của bài báo và các bài liên quan | 70% | 7/10 |
| Hỗ trợ Kiên cài đặt và chạy code |
| Phối hợp với Lâm viết báo cáo |
| 20C11034  Nguyễn Trung Kiên | Cài đặt và chạy code của bài báo cung cấp | 70% | 7/10 |
| Thực hiện demo kết quả cài đặt |
| Phối hợp viết báo cáo |

MỤC LỤC

[I. TỔNG QUAN 2](#_Toc67013403)

[1. Giới thiệu 2](#_Toc67013404)

[2. Đặt vấn đề 2](#_Toc67013405)

[II. PHÂN TÍCH BỘ DỮ LIỆU SỬ DỤNG 4](#_Toc67013406)

[1. Nhận xét về các bộ dữ liệu về biển báo giao thông trước nghiên cứu này 4](#_Toc67013407)

[2. Cách tạo ra bộ dữ liệu sử dụng trong nghiên cứu 4](#_Toc67013408)

[3. Ưu điểm của bộ dữ liệu mới này 4](#_Toc67013409)

[4. Nhiễu trong bộ dữ liệu 5](#_Toc67013410)

[5. Annotation 5](#_Toc67013411)

[6. Thống kê trong bộ dữ liệu 5](#_Toc67013412)

[III. MÔ HÌNH MẠNG ĐỀ XUẤT 6](#_Toc67013413)

[1. Các mô hình trước đó 6](#_Toc67013414)

[2. Đề xuất mô hình mới 8](#_Toc67013415)

[IV. KẾT QUẢ 10](#_Toc67013416)

[V. CHẠY THỬ NGHIỆM 11](#_Toc67013417)

[1. Yêu cầu môi trường và cài đặt (Đề xuất) 11](#_Toc67013418)

[2. Cấu trúc của mã nguồn 11](#_Toc67013419)

[3. Quan sát bộ dữ liệu TT100K 11](#_Toc67013420)

[4. Chạy thử nghiệm với model của tác giả 13](#_Toc67013421)

[TÀI LIỆU THAM KHẢO 15](#_Toc67013422)

# TỔNG QUAN

## Giới thiệu

Bài báo “*Traffic-Sign Detection and Classification in the Wild*” [1] đến từ đội ngũ Nhóm nghiên cứu của Phòng Nghiên cứu Khoa Học Máy Tính và Công Nghệ, Đại học Thanh Hoa Bắc Kinh (Tsinghua University). Bài báo được công bố ở hội nghị IEEE CVPR 2016 tổ chức tại Mỹ. Nhóm tác giả bao gồm các thành viên Zhe Zhu, Dun Liang, Song-Hai Zhang, Xiaolei Huang, Baoli Li và Shi-Min Hu.

Bài báo thuộc chủ đề Object Detection, là 1 bài toán không mới và cũng đã đạt được rất nhiều các thành tựu trong những năm gần đây, cả phần ứng dụng và mô hình thuật toán. Điển hình là các phương pháp Object Detection sử dụng Deep Learning đã đạt được các bước cải thiện vượt trội so với các phương pháp xử lý ảnh thông thường khác. Ngoài ra, các lĩnh vực liên quan được đề cập còn bao gồm Benchmark, Recogmition, CNN, Image Processing.

Trong phần introduction, tác giả giới thiệu 2 mục tiêu và kết quả cốt lõi chính của bài báo đó là:

* Xây dựng một tập data Tsinghua-Tencent 100K thực tế hơn các tập data về Traffic-Sign trước đó
* Xây dựng được mạng CNN cải tiến để vừa xác định biển báo, vừa phân loại biển báo đồng thời

Nhóm tác giả đã cung cấp bộ dataset 100K tấm ảnh góc rộng (panorama) từ Tencent Street View [2]. Ngoài ra, trên web site của bài báo còn có hướng dẫn chạy model của họ với TT100K dataset, phụ lục kèm theo bao gồm kết quả nhận diện một số trường hợp, và kết quả phân loại một số trường hợp điển hình trong thực nghiệm. Có thêm một file caffemodel và 3 file .prototxt để train và test cho mô hình Fast R-CNN, kèm theo mã nguồn chính của bài báo.

## Đặt vấn đề

Trong chũ đề Object Detection, nhóm tác giả có trích dẫn nhiều bài báo trước đó cùng chủ đề này đã giải quyết được các bài toán “Traffic Sign Recognition” [3] hay cuộc thi GTSRB [4]. Họ đã Ứng dụng và cải tiến được đáng kể các thuật toán phổ biến trong object detection như:

* R-CNN -> SPPNet -> Fast R-CNN -> Faster R-CNN
* OverFeat: là một feature extractor được huấn luyện trên tập dữ liệu ImageNet

Kể từ khi có tập data về biển báo đầu tiên của Đức GTSDB và GTSRB [4] đến nay, có nhiều nhóm nghiên cứu đã làm về 2 pha này (detection & classification), và đạt đến gần 100% độ chính xác (recall & precision for detection) và 99.67% (classification). Các phương pháp đưa ra đã đạt kết quả rất tốt trên tập GTSRB, lên đến 99.6% tốt hơn cả con người.

Ta tìm hiểu sơ qua về tập data GTSRB sử dụng trong cuộc thi:

* Tập data bao gồm hơn 50,000 ảnh được chụp trên đường phố nước Đức
* Chia thành 43 lớp loại khác nhau của Traffic-sign
* Được trích xuất từ hơn 10 giờ video với độ phân giải 1360×1024 pixels, sau đó được convert từ dạng Raw sang RGB format
* Dữ liệu sau đó sử dụng NISYS Advanced Development and Analysis Framework để đánh dấu và gán nhãn thủ công
* Kích thước của traffic signs trong tập này dao động từ 15×15 đến 222×193 pixel

**Vậy theo nhóm tác giả, vấn đề còn tồn đọng và phát sinh từ các bài báo này là gì?**

Tác giả nhận định các dataset trước đó không đại diện cho hầu hết các trường hợp thực tiễn, ví dụ như trong GTSRB, biển báo lại xuất hiện trong hầu hết các bức ảnh, và thuật toán chỉ việc phân loại biển báo trong ảnh mà thôi. Ngoài ra, Bộ dataset chưa có các trường hợp gây nhiễu làm lệch kết quả tính toán. Trong thực tế biển báo chỉ chiếm một phần rất nhỏ trong ảnh, thường nhỏ hơn 1%. Trong khi đó dataset hiện tại thì hình biển báo chiếm tỉ lệ tương đối trong bức ảnh. Như vậy là chưa đủ thực tế.

Trong ImageNet và COCO, hình được lấy từ Internet search engine, mà ở đó khá ít user tải ảnh từ thực tế cuộc sống mà có biển báo giao thông như khi ta đi trên đường, hoặc nếu có thì cũng chỉ là vô tình và ngẫu nhiên. Cách thu thập data này thiếu tính thực tế. Trong GTSRB, các biển báo được trích xuất từ một phân đoạn video, tức hình biển báo sẽ rất giống nhau ở mọi hình. Điều này dẫn đến độ đa dạng trong tập data sẽ không được như mong đợi.

# PHÂN TÍCH BỘ DỮ LIỆU SỬ DỤNG

## Nhận xét về các bộ dữ liệu về biển báo giao thông trước nghiên cứu này

Các bộ dữ liệu đã có gồm: PASCAL VOC, ImageNet ILSVR-C, GTSRB

Khuyết điểm:

* Hình ảnh được thu thập từ các ảnh trên internet do người dùng đăng, trong đó sự xuất hiện của biển báo giao thông chỉ là ngẫu nhiên (khó khăn trong việc có được bộ dữ liệu đủ lớn)
* Cắt các hình liên tiếp từ video dẫn đến các biển báo sẽ xuất hiện rất giống nhau trong các ảnh (GTSRB)

## Cách tạo ra bộ dữ liệu sử dụng trong nghiên cứu

Bộ dữ liệu gồm 100000 tấm ảnh với khoảng 30000 biển báo giao thông được thu thập từ các hình ảnh trên Tencent Street View (từ 300 thành phố ở Trung Quốc và các tuyến đường kết nối chúng), được nhóm tác giả gọi là “Tsinghua-Tencent 100K”

Các ảnh trên Tencent Street View được chụp từ 1 hệ thống gồm camera 6 SLR gắn trên các phương tiện giao thông, hoặc đeo trên vai người ở những nơi mà xe không đi tới được, và sau đó ghép nối lại với nhau tạo thành 1 ảnh panorama 360.

Các tấm ảnh được nhóm tác giả lấy mỗi 10 phút, việc này sẽ giúp:

* Các ảnh được chụp liên tiếp thì liên kết đồng nhất với nhau. Không như trong GTSRB, khi các biển báo được trích xuất từ một phân đoạn video, tức hình biển báo sẽ rất giống nhau ở mọi hình.
* Có được các tấm ảnh liên tục về một biển báo sẽ tăng độ chính xác cho việc xác định loại biển báo đó, ví dụ khi biển báo bị che khuất hoặc bị mờ một phần nào đó, có thể được nhận dạng từ những tấm hình trước đó hoặc sau đó trong chuỗi hình liên kết này.

Sau đó các tấm ảnh panorama được crop thành các ảnh khoảng 2000\*2000 px, và sử dụng các kĩ thuật xử lý (như tăng độ tương phản, xoay ảnh, làm méo ảnh) để cho ra ảnh được sử dụng trong bộ data

## Ưu điểm của bộ dữ liệu mới này

Với bộ data mới này (Tsinghua-Tencent 100K), các tấm ảnh được chụp từ các camera trong những điều kiện ánh sáng và thời tiết khác nhau, các biển báo chỉ chiếm một phần nhỏ và có thể ở bất cứ vị trí nào trong ảnh nên sẽ mô phỏng các tình huống thực tế tốt hơn.

Nhiều loại biển báo (100K ảnh với 30K biển báo), góc chụp đa dạng hơn (chính diện, nghiêng), có các biển nhìn rõ và cả những biển bị che khuất một phần

Ảnh chụp trong nhiều điều kiện thời tiết (trời nắng, có mây)

Chụp ở nhiều điều kiện ngoại cảnh (đô thị, nông thôn, đường cao tốc)

Nhiều ảnh chụp có background phức tạp làm tăng độ khó khi detect biển báo

## Nhiễu trong bộ dữ liệu

Có sử dụng khá nhiều hình ảnh không có biển báo, có add thêm độ nhiễu ngẫu nhiên

Gây nhiễu: xoay hình ngẫu nhiên trong khoảng -20 độ -> +20 độ, scale hình, làm méo hình

## Annotation

Các tấm ảnh được đánh nhãn bằng tay, ghi dấu lại khung bao, các đỉnh của khung và gán nhãn cho biển báo trong từng tấm ảnh

Phân loại thành 3 loại biển báo chính:

* Biển cảnh báo (tam giác vàng viền d9en)
* Biển cấm (tròn trắng viền đỏ)
* Biển hiệu lệnh (tròn xanh hình trắng)
* Các loại biển báo khác ít phổ biến

## Thống kê trong bộ dữ liệu

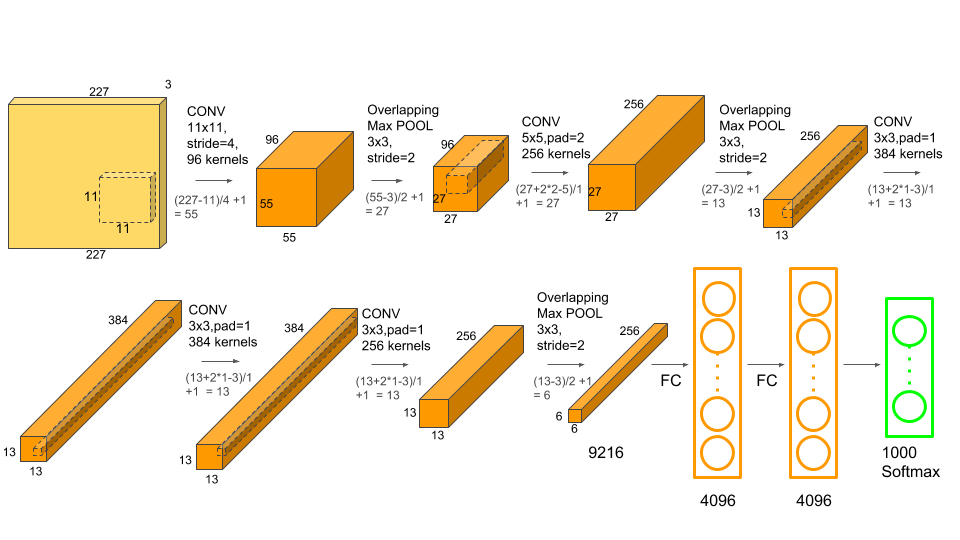
Sự phân bố của các loại biển báo trong bộ dữ liệu không đều nhau (do thực tế trên đường, sự xuất hiện của các loại biển báo sẽ khác nhau, tập trung vào một số biển báo thường gặp)

Sự phân bố của các kích thước của biển báo không đều nhau, trải dài từ 16\*20 px tới 160\*160 px, trong đó tập trung vào khoảng từ 16\*20 px tới 80\*80 px

# MÔ HÌNH MẠNG ĐỀ XUẤT

## Các mô hình trước đó

Đầu tiên, **AlexNet** là một trong những Mô hình khởi điểm giải quyết bài toán phân lớp (classification) một bức ảnh vào 1 lớp trong 1000 lớp khác nhau (vd gà, chó, mèo … ). Đầu ra của mô hình là một vector có 1000 phần tử. Mạng CNN này đã thắng hạng nhất trong cuộc thi ILSVRC năm 2012. Đầu vào của AlexNet là một bức ảnh RGB 3x256x256 ở cả tập train và tập test. Đây là kích thước chuẩn bắt buộc sử dụng trong mạng. Sau khi chuẩn hóa, kích thước đầu vào được sử dụng là 227x227 và cắt ngẫu nhiên trên hình gốc 256x256.

Kiến trúc mạng như hình III.a:

Hình ‑: Kiến trúc mạng AlexNet (Nguồn: phamduytung.com)

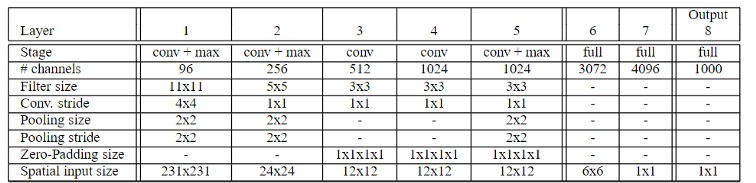
Kiến trúc AlexNet bao gồm 5 convolutional Layer và 3 fully connected layer. Nó có tổng cộng 60 triệu tham số và 650 nghìn neural. Các convolutional Layer (filter) đầu tiên có chức năng trích xuất các đặc trưng cơ bản của tấm ảnh. Filter đầu tiên chứa 96 kernel có kích thước 3x11x11, stride=4.

Các layer sau kết nối với layer trước đó qua một Overlapping Max Pooling ở layer thứ 1,2 và 5. Max Pooling layer thường được sử dụng để giảm chiều rộng và chiều dài của một tensor nhưng vẫn giữ nguyên chiều sâu. ReLU nonlinerity được sử dụng sau tất các các convolution và fully connected layer. Theo tác giả, ReLU giúp cho mạng huấn luyện nhanh hơn và cải thiện độ lỗi gấp nhiều lần so với khi dùng hàm Tanh hay Sigmoid.

Cho đến cuối cùng, layer thứ 7 là fully connected kết nối với layer 8 là một bộ phân lớp softmax với 1000 vector đầu ra, với tổng giá trị bằng 1.

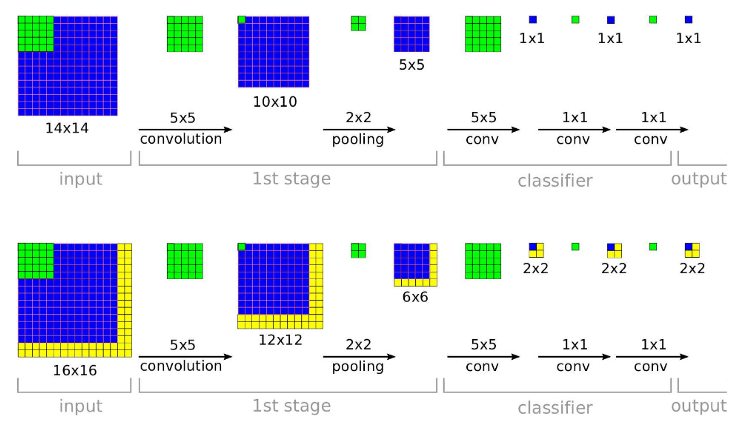
Tiếp đó, **OverFeat** [3] là một mạng CNN dùng để detect (classification, localization and detection) đối tượng cực kỳ hiệu quả, được cải tiến dựa trên mạng CNN AlexNet.

Hình III-b dưới đây mô tả kiến trúc cơ bản của mạng OverFeat



Hình ‑: Kiến trúc mạng OverFeat (Nguồn: IEEE)

Các cải tiến của OverFeat so với AlexNet bao gồm:

* Pooling region không xài Overlapping, tức stride của lớp max pooling đúng bằng kích thước của kernel, các tấm kernel khi trượt sẽ không chồng lắp lên nhau (mạng AlexNet có chồng lên 1 pixel).
* Feature map sẽ to hơn một chút do dùng slide nhỏ hơn so với AlexNet.
* Sử dụng cửa số trượt sliding window hiệu quả hơn nhờ vào việc share kết quả tính toán của một vùng từ lần trước cho lần tính toán sau (thường sẽ overlap nhau rất nhiều). Ta xem hình minh họa *III-c* sau:

Hình ‑: Minh họa kỹ thuật sliding window (Nguồn: IEEE)

* Sử dụng cơ chế multi-view voting: scale tấm ảnh ra thành nhiều tấm ảnh, sau đó dự đoán tất cả chúng, rồi lấy trung bình kết quả.
* Để localization, tìm ra bouding box của đối tượng, mạng này thay classifier layer (layer thứ 6,7,8) bằng một mạng hồi quy, sử dụng kết quả từ layer thứ 5 làm input. Ở đó, output cuối cùng của mạng hồi quy này là các vector 4 chiều mô tả bounding box của input tương ứng. Như vậy bằng việc tách nhánh ở cuối layer 5, tác giả đã xử lý được song song việc classification và tìm ra bounding box luôn.

Sau cùng, Tác giả dựa trên giải pháp “Vehicle Detection” [5] của bài báo trước đó, sử dụng kết hợp OverFeat và một mạng Hồi quy ở cuối. Cụ thể:

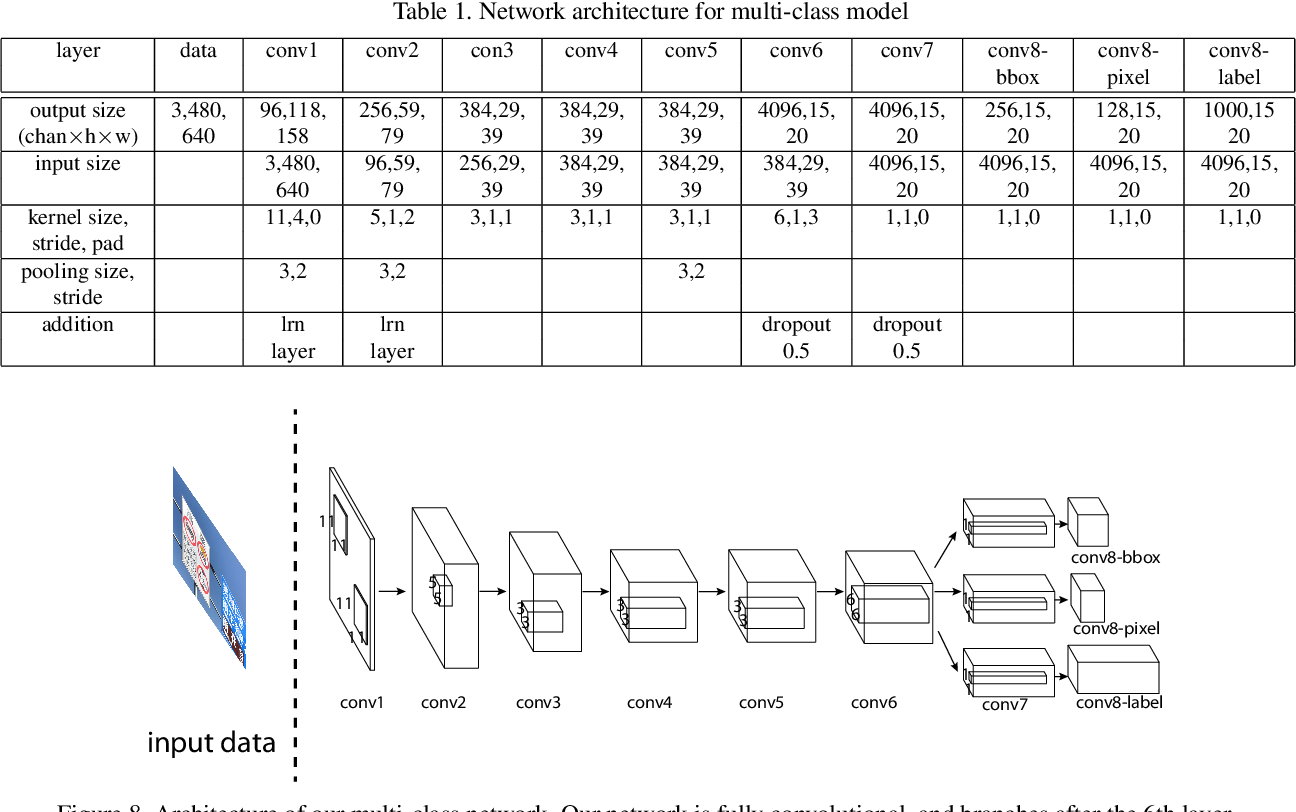
* Hình input của mạng này được nâng lên 640 × 480, để đáp ứng bài toán detect phương tiện và lane đường xa hơn 100m so với camera.
* Kết hợp thêm một mask detector, để xác định được bonding box nhanh hơn so với OverFeat nguyên thủy.
* Các tầng layer fully connected (dense) được chuyển đổi thành các mạng convolution.
* Khi sử dụng hình input lớn hơn, dẫn đến output ở layer thứ 6 cũng nở ra từ 1x1x4096 thành 20x15x4096

## Đề xuất mô hình mới

Mô hình mới lấy ý tưởng từ cả mô hình OverFeat và Vehicle Detection như đã trình bày ở phần trên cho bài toán Traffic Sign Detection. Các thay đổi, điều chỉnh so với mô hình gốc như sau:

* Các tác giả đã thử nghiệm việc chạy song song trước khi xong layer thứ 7, và nhận thấy tách nhánh để chạy song song từ sau layer 6 là giải pháp cân bằng giữa thời gian tính toán và độ chính xác cần thiết (tách càng sớm thì performance càng nhanh nhưng tốn nhiều tài nguyên, tách quá trễ sẽ tính toán lâu hơn)
* So với bài toán Vehicle Detection, thì layer cuối cùng có thêm 1 nhánh chạy song song giúp phân loại luôn loại biển báo.

Kiến trúc mạng được mô tả trong hình *III-d*



Hình ‑: Kiến trúc mạng được nhóm tác giả đề xuất

# KẾT QUẢ

# CHẠY THỬ NGHIỆM

## Yêu cầu môi trường và cài đặt (Đề xuất)

* Ubuntu 14.04.5 LTS
* Nvidia driver phù hợp
* CUDA SDK 8.0
* cuDNN 6.0
* Python 2.7.x
* Jupyter Notebook (Python 2.7.x)

## Cấu trúc của mã nguồn

Thư mục **code**:

* caffe: caffe code tác giả sử dụng
* model: tập tin caffe model
* script: những script được sử dụng để build và train model
* python: một số tiện ích sử dụng python

Thư mục **data:**

* train: tất cả hình ảnh sử dụng để huấn luyện
* test: tất cả hình ảnh sử dụng để thử nghiệm
* other: một số hình ảnh ngoài tập train và test
* marks: tất cả hình ảnh về các loại biển báo
* annotations.json: tập tin json chứa tất cả annotation của tập dữ liệu
* results: kết quả của mô hình tác giả và fast-rcnn

## Quan sát bộ dữ liệu TT100K

Chúng ta cần nạp vào tập tin json annotations và danh sách id của bộ dữ liệu test

datadir = "../../data/"

filedir = datadir + "/annotations.json"

ids = open(datadir + "/test/ids.txt").read().splitlines()

annos = json.loads(open(filedir).read())

Sau đó, hiện thị một hình ảnh bất kì sử dụng tiện ích đã cung cấp

datadir = "../../data/"

filedir = datadir + "/annotations.json"

ids = open(datadir + "/test/ids.txt").read().splitlines()

annos = json.loads(open(filedir).read())

****

*Hình V-a: Kết quả quan sát từ tập dữ liệu TT100K*

## Chạy thử nghiệm với model của tác giả

Sau khi đã huấn luyện thành công với bộ dữ liệu, ta nhận được một tập tin json annotation sử dụng tập tin .caffemodel.

Tiến hành nạp vào tập tin kết quả, hàm eval\_annos cũng sẽ giúp thể hiện được bao nhiêu biển báo được đánh giá đúng, sai hoặc bị thiếu sót.

result\_anno\_file = "./../results/ours\_result\_annos.json"

results\_annos = json.loads(open(result\_anno\_file).read())

sm = anno\_func.eval\_annos(annos, results\_annos, iou=0.5, types=anno\_func.type45, check\_type=True)

Sau đó, hiển thị kết quả với hình ảnh bất kì. Hiển thị những biển báo được đánh giá đúng với bounding-box màu xanh lá, sai với màu đỏ và bị thiếu với màu xanh dương.

while True:

imgid = random.sample(results\_annos['imgs'].keys(), 1)[0]

if len(sm['wrong']["imgs"][imgid]["objects"]) and len(sm['miss']["imgs"][imgid]["objects"]):

break

print imgid

pl.figure(figsize=(20,20))

imgdata = anno\_func.load\_img(annos, datadir, imgid)

imgdata = anno\_func.draw\_all(sm['right'], datadir, imgid, imgdata, (0,1,0), False)

imgdata = anno\_func.draw\_all(sm['wrong'], datadir, imgid, imgdata, (1,0,0), False)

imgdata = anno\_func.draw\_all(sm['miss'], datadir, imgid, imgdata, (0,0,1), False)

pl.imshow(imgdata)

****

*Hình V-b: Kết quả sử dụng model của tác giá với tập dữ liệu TT100K*

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | D. L. S. Z. X. H. B. L. S. H. Zhe Zhu, "Traffic-Sign Detection and Classification in the Wild," in *CVPR*, Las Vegas, Nevada, 2016. |
| [2] | D. L. S. Z. X. H. B. L. S. H. Zhe Zhu, "TSinghua University-Tencent Joint Laboratory," [Online]. Available: https://cg.cs.tsinghua.edu.cn/traffic-sign/. |
| [3] | T. s. r. w. m.-s. C. Networks, "Traffic sign recognition with multi-scale Convolutional Networks," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, 2011. |
| [4] | M. S. Johannes Stallkamp, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA. |
| [5] | T. W. S. T. J. K. W. S. J. P. M. A. P. R. T. M. R. C.-Y. F. M. A. C. A. Y. N. Brody Huval, "An Empirical Evaluation of Deep Learning on Highway Driving," 2015. |