**TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN**

**Thị Giác Máy Tính**

**Đề Tài** : **NHẬN DIỆN CÁC PHƯƠNG TIỆN GIAO THÔNG**

**Giảng Viên** : Nguyễn Đình Quý

**Nhóm Thực Hiện** : 8

**Lớp 67CS2**

**Thành Viên**:

- Nguyễn Sơn Nam – 0055267

-Đỗ Văn Duy - 0190167

-Phạm Ngọc Minh - 0117867

-Nguyễn Xuân Hoàng - 0034267

**I. Giới thiệu đề tài**

Trong bối cảnh sự phát triển mạnh mẽ của trí tuệ nhân tạo và thị giác máy tính (Computer Vision), các kỹ thuật học sâu (Deep Learning) đã mở ra nhiều cơ hội để giải quyết các bài toán phức tạp trong nhiều lĩnh vực, đặc biệt là trong quản lý và tối ưu hóa giao thông đô thị. Một trong những bài toán quan trọng và mang tính ứng dụng cao là **nhận diện các phương tiện giao thông**, nhằm tự động phát hiện, phân loại và theo dõi các loại phương tiện (như xe hơi, xe máy, xe buýt, xe tải, v.v.) từ hình ảnh hoặc video thu thập được từ camera.

Với sự gia tăng nhanh chóng của mật độ giao thông và nhu cầu xây dựng các hệ thống giao thông thông minh, xe tự hành, cũng như giám sát an ninh giao thông, việc nhận diện chính xác các phương tiện giao thông đóng vai trò cốt lõi. Nhận diện phương tiện không chỉ giúp xác định loại phương tiện mà còn hỗ trợ phân tích hành vi giao thông, phát hiện vi phạm, tối ưu hóa lưu lượng giao thông và đảm bảo an toàn cho các phương tiện tự hành.

Đề tài **"Nhận diện các phương tiện giao thông bằng Deep Learning"** tập trung nghiên cứu và ứng dụng các mô hình học sâu tiên tiến (như YOLO, Faster R-CNN, hoặc SSD) để thực hiện nhiệm vụ phát hiện và phân loại phương tiện giao thông trong các hình ảnh hoặc video giao thông thực tế. Dữ liệu đầu vào thường được lấy từ các bộ dữ liệu chuẩn như KITTI, COCO, hoặc các tập dữ liệu giao thông tùy chỉnh, với đầu ra là các hộp bao (bounding boxes) kèm nhãn phân loại tương ứng cho từng phương tiện.

* **Mục tiêu của đề tài**
* Phát triển và huấn luyện các mô hình học sâu đạt độ chính xác cao trong việc nhận diện và phân loại phương tiện giao thông.
* Tối ưu hóa hiệu năng mô hình để đáp ứng yêu cầu xử lý thời gian thực, phù hợp với các ứng dụng thực tế như xe tự hành hoặc hệ thống giám sát giao thông.
* So sánh và đánh giá hiệu quả của các kiến trúc mạng khác nhau (như YOLOv5, YOLOv8, Faster R-CNN) trong bài toán nhận diện phương tiện giao thông.
* **Ý nghĩa và ứng dụng**

Đề tài không chỉ mang ý nghĩa khoa học trong việc thúc đẩy nghiên cứu thị giác máy tính mà còn có giá trị thực tiễn cao, góp phần:

* Hỗ trợ các hệ thống giao thông thông minh, như quản lý đèn giao thông, phát hiện ùn tắc, hoặc tối ưu hóa luồng giao thông.
* Cung cấp dữ liệu quan trọng cho các phương tiện tự hành để nhận biết và phản ứng với các phương tiện xung quanh.
* Tăng cường an ninh giao thông thông qua giám sát và phát hiện các phương tiện bất thường hoặc vi phạm.

Đề tài hứa hẹn mang lại những đóng góp thiết thực trong việc xây dựng các hệ thống giao thông hiện đại, an toàn và hiệu quả hơn, đồng thời mở ra hướng nghiên cứu sâu hơn về ứng dụng học sâu trong các bài toán thị giác máy tính liên quan đến giao thông.

**II.Cơ sở lý thuyết**

**2.1 Tìm hiểu thuật toán**

**2.1.1.YOLOv5**

YOLOv5 (You Only Look Once) là một thuật toán phát hiện đối tượng thời gian thực, sử dụng kiến trúc CNN để dự đoán bounding box và lớp đối tượng trong một lần quét. YOLOv5 tối ưu hóa tốc độ và độ chính xác với các kích thước mô hình khác nhau (n, s, m, l, x).

**2.1.2.YOLOv8**

YOLOv8 là phiên bản cải tiến của YOLO, với các cải tiến về kiến trúc backbone (CSPDarknet). Đây là mô hình mới nhất với độ chính xác cao hơn, hỗ trợ phân đoạn đối tượng và phát hiện keypoint.

**2.1.3.Faster R-CNN**

Faster R-CNN sử dụng Region Proposal Network (RPN) để tạo vùng đề xuất, sau đó phân loại và tinh chỉnh bounding box. Đây là mô hình hai giai đoạn, phù hợp với các bài toán yêu cầu độ chính xác cao.

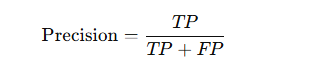
**2.1.4.SSD**

SSD (Single Shot MultiBox Detector) là mô hình một giai đoạn, sử dụng các feature map đa tỷ lệ để phát hiện đối tượng. SSD nhanh nhưng độ chính xác thấp hơn trong các đối tượng nhỏ.

|  |  |  |
| --- | --- | --- |
| **Mô Hình** | **Ưu Điểm** | **Nhươc Điểm** |
| **YOLOv5** | - Tốc độ cao, dễ triển khai.  - Cân bằng tốt giữa tốc độ và độ chính xác.  - Hỗ trợ nhiều phiên bản cho thiết bị khác nhau. | - Độ chính xác thấp với đối tượng nhỏ.  - Hiệu suất giảm trong điều kiện ánh sáng yếu. |
| **YOLOv8** | - Độ chính xác cao hơn YOLOv5. - Hỗ trợ phân đoạn và keypoint.  - Tối ưu hóa tốt cho bài toán phức tạp. | - Yêu cầu tài nguyên cao hơn YOLOv5.  - Tốc độ chậm hơn YOLOv5 ở cùng cấu hình. |
| **Faster R-CNN** | - Độ chính xác cao, hiệu quả với đối tượng nhỏ.  - Linh hoạt với tập dữ liệu đa dạng | - Tốc độ chậm, không phù hợp thời gian thực.  - Triển khai phức tạp, yêu cầu cấu hình mạnh. |
| **SSD** | - Tốc độ nhanh, triển khai đơn giản.  - Phù hợp với thiết bị tài nguyên hạn chế. | - Độ chính xác thấp với đối tượng nhỏ.  - Khó xử lý tỷ lệ khung hình đa dạng. |

**2.2. Các chỉ số đánh giá mô hình**

**- Precision ( Độ chính xác):**

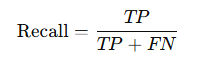
****

Trong đó:

TP: True Positive (nhận diện đúng đối tượng)

FP: False Positive (nhận diện sai đối tượng)

* **Recall ( Độ bao phủ):**

****

Trong đó:

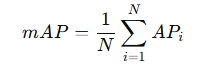
FN: False Negative (bỏ sót đối tượng)

* **F1-score (Trung bình điều hòa giữa precision và recall):**



F1-score đặc biệt hữu ích khi bạn cần cân bằng giữa precision và recall.

* **mAP – mean Average Precision**

****

Trong đó:

N: số lớp

APi: Average Precision cho lớp thứ i

**2.3 .Công cụ gán nhãn dữ liệu**

**• LabelImg**: Công cụ mã nguồn mở, hỗ trợ định dạng YOLO, Pascal VOC. Dễ sử dụng, phù hợp cho dữ liệu nhỏ.

**• Roboflow**: Nền tảng trực tuyến, hỗ trợ quản lý tập dữ liệu, tăng cường dữ liệu, và xuất nhiều định dạng. Phù hợp với dự án lớn nhưng có giới hạn miễn phí.

**III.Phương pháp thực hiện**

**3.1. Lập kế hoạch chọn mô hình**

Dựa trên yêu cầu về tốc độ và độ chính xác, YOLOv8 được chọn vì:

• Độ chính xác cao, phù hợp với bài toán nhận diện đa lớp.

• Hỗ trợ thời gian thực trên video giao thông.

• Cộng đồng hỗ trợ mạnh, dễ tích hợp.

**3.2 Xác định danh sách phương tiện**

Các phương tiện cần nhận diện:

• Xe máy

• Ô tô

• Xe tải

• Xe buýt

**3.3 Thu thập và tiền xử lý dữ liệu**

**• Nguồn dữ liệu**: Sử dụng tập COCO (chứa các lớp phương tiện) và Open Images (chứa ảnh giao thông đa dạng).

**• Lọc dữ liệu**: Loại bỏ ảnh mờ, ánh sáng kém, hoặc không chứa phương tiện bằng kiểm tra thủ công và thuật toán đo độ nét (Laplacian variance).

**• Chuẩn hóa:** Chuyển tất cả ảnh về kích thước 640x640 pixel để đồng nhất đầu vào mô hình.

**3.4 Gán nhãn dữ liệu**

Sử dụng LabelImg để gán nhãn thủ công, định dạng YOLO (txt file chứa tọa độ bounding box và lớp).Mỗi ảnh có file nhãn tương ứng với định dạng:

<class\_id> <x\_center> <y\_center> <width> <height>

**3.5 Chia tập dữ liệu**

• Train: 80% (8,000 ảnh).

• Validation: 20% (2,000 ảnh).

**3.6 Thiết lập tham số huấn luyện**

• Batch size: 16

• Learning rate: 0.001

• Epochs: 20

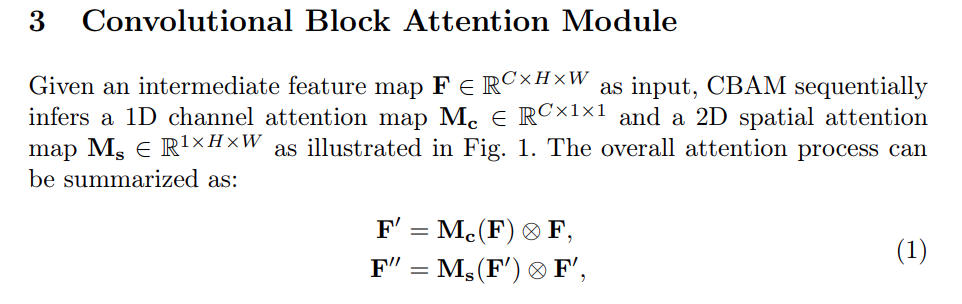
**3.7 Tối ưu mô hình**

🞄 Sử dụng Data Augmentation để cải thiện tập dữ liệu, tăng độ chính xác

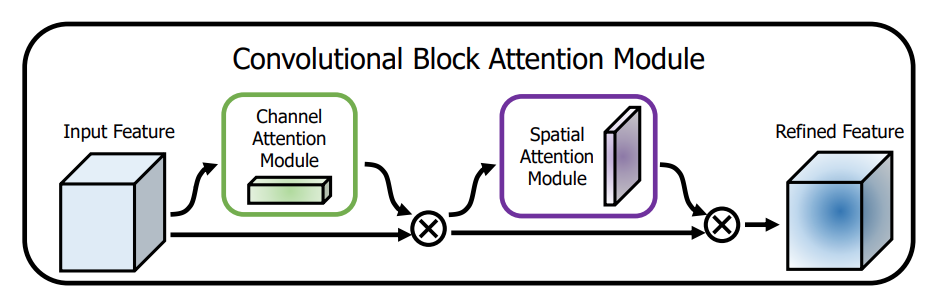
A computer screen with text and images

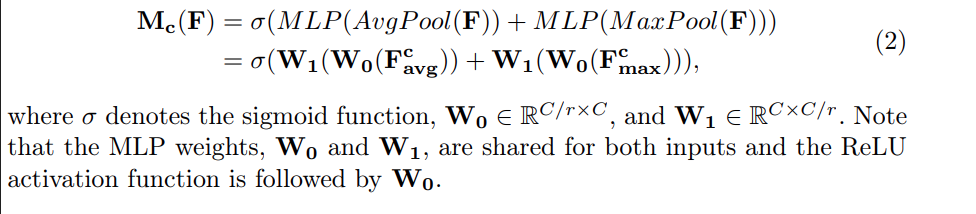
AI-generated content may be incorrect.

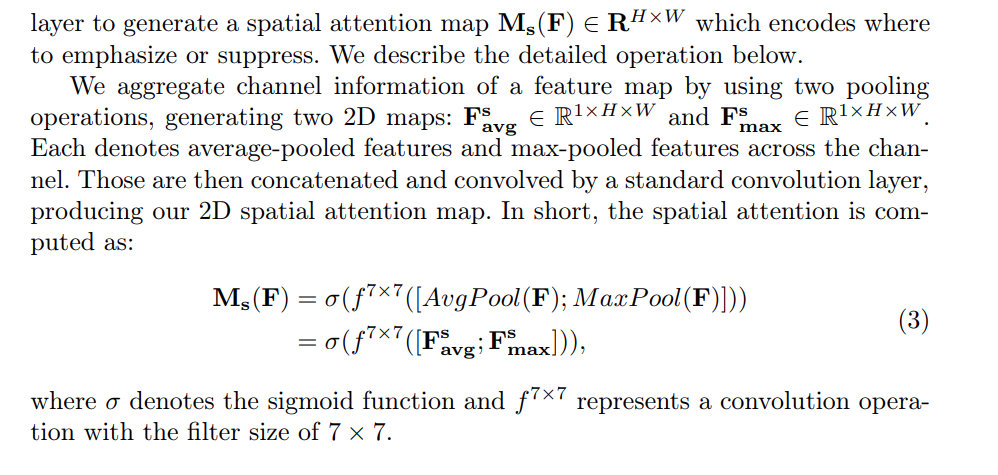
**🞄 Điều chỉnh để mô hình tăng độ chính xác**

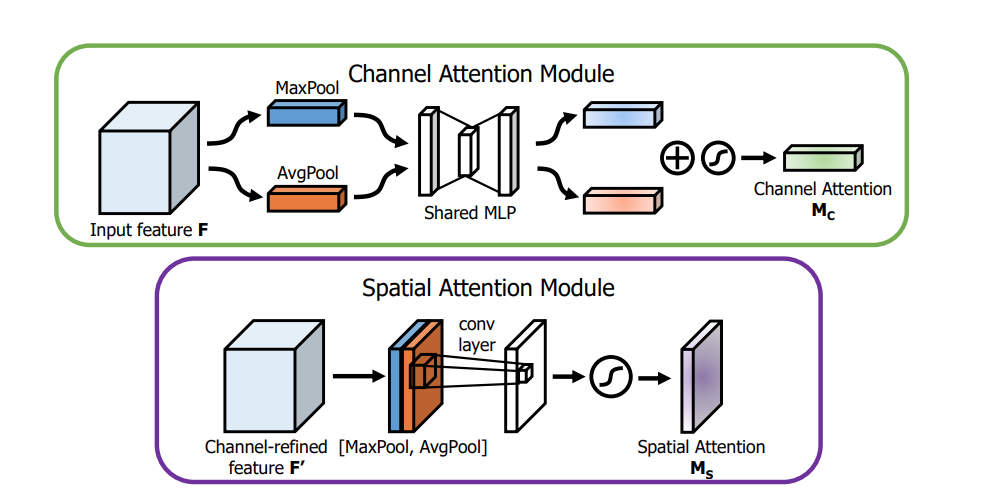


* Việc tích hợp CBAM vào kiến trúc YOLOv8 giúp mô hình tập trung hơn vào các vùng và kênh đặc trưng quan trọng trong ảnh. Thay vì đối xử đồng đều với toàn bộ thông tin đầu vào, mô hình sẽ ưu tiên xử lý các đặc trưng nổi bật, nhờ đó cải thiện khả năng phân biệt giữa các loại phương tiện có hình dáng, màu sắc hoặc vị trí tương đối gần nhau.  
    
  **Cụ thể:**  
  - Channel Attention giúp mô hình tập trung vào các kênh quan trọng nhất, loại bỏ thông tin dư thừa hoặc nhiễu.  
  - Spatial Attention làm nổi bật các vùng có khả năng chứa đối tượng, giúp mô hình dễ dàng xác định vị trí phương tiện.

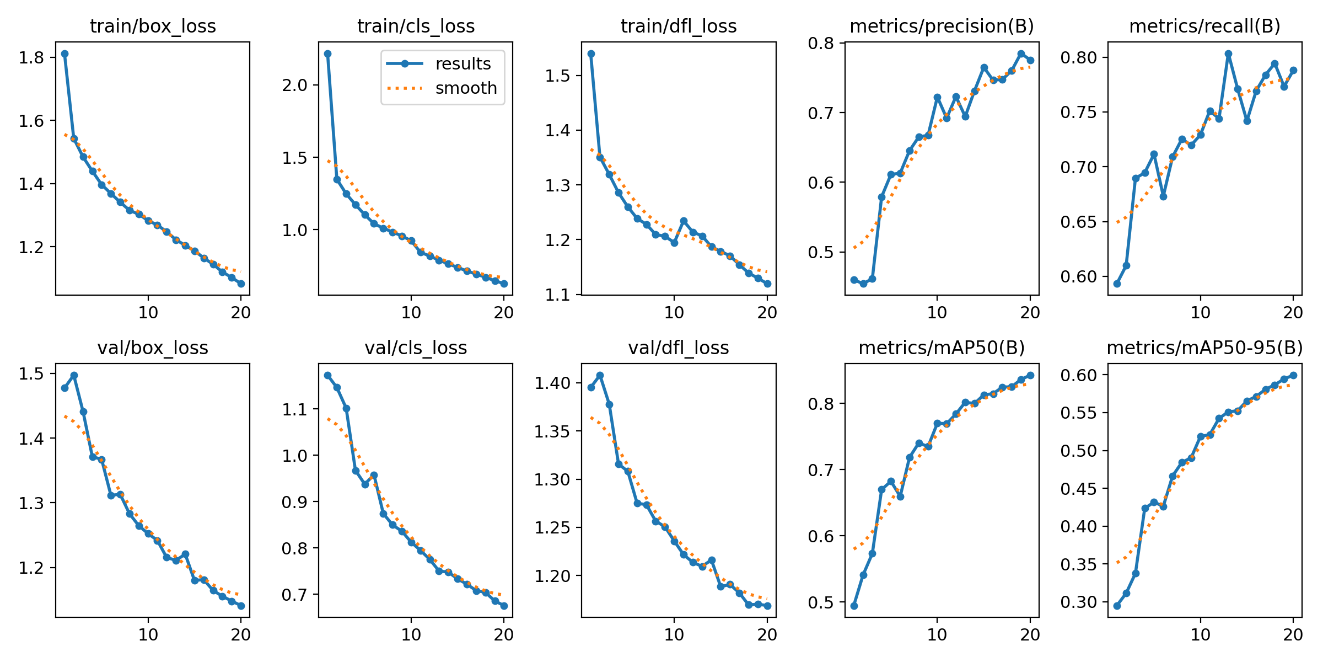








**3.8 Kết quả train mô hình**





**Phân tích kết quả huấn luyện**

Biểu đồ thể hiện tiến trình huấn luyện trong 20 epoch của mô hình YOLOv8 (có thể đã tích hợp Attention Module), với các chỉ số chính:

1. Train Loss (hàng đầu):

- train/box\_loss, train/cls\_loss, train/dfl\_loss đều giảm dần và ổn định → cho thấy mô hình học tốt theo thời gian, không có dấu hiệu overfitting ở giai đoạn huấn luyện.

2. Validation Loss (hàng dưới):

- val/box\_loss, val/cls\_loss, val/dfl\_loss cũng giảm ổn định, bám sát train loss → chứng tỏ mô hình tổng quát hóa tốt trên dữ liệu kiểm thử.

- Đây là dấu hiệu tích cực cho việc mô hình học đúng cách và không bị lệch nhãn hoặc sai phân phối.

3. Chỉ số chính xác và độ bao phủ:

- precision(B) và recall(B) đều tăng ổn định qua từng epoch, đạt gần 0.8 → hiệu suất phát hiện đối tượng cao.

- recall có dao động nhẹ nhưng xu hướng vẫn tăng → cần theo dõi thêm nếu tập dữ liệu có nhiều đối tượng nhỏ hoặc trùng nhau.

4. Chỉ số mAP (mean Average Precision):

- metrics/mAP50(B) tiến gần tới 0.85 → rất tốt cho các bài toán nhận diện giao thông.

- metrics/mAP50-95(B) (khắt khe hơn) cũng vượt 0.6, chứng tỏ mô hình hoạt động tốt ở nhiều mức độ IoU.

**Kết luận ngắn gọn:**

* Mô hình huấn luyện tốt, không overfit.
* Độ chính xác và khả năng bao phủ đều cao.
* mAP đạt mức rất tốt cho bài toán nhận diện phương tiện giao thông, đặc biệt nếu đã tích hợp CBAM thì hiệu quả càng được nâng cao.

**IV. Tích hợp mô hình vào ứng dụng**

**1. TỔNG QUAN**

YOLO Detector là ứng dụng web phát hiện đối tượng sử dụng mô hình YOLO, cho phép người dùng tải lên hình ảnh và nhận kết quả phát hiện real-time.

**2. KIẾN TRÚC HỆ THỐNG**

Frontend (React) ←→ Backend (Flask) ←→ MongoDB

Port: 3000 Port: 5000 Port: 27017

**3. CÔNG NGHỆ SỬ DỤNG**

**Frontend**

- React: Giao diện người dùng

- React Router: Điều hướng

- Axios: HTTP client

- CSS3: Styling và animation

**Backend**

- Flask: Web framework

- MongoDB: Cơ sở dữ liệu

- JWT: Xác thực

- YOLO: Phát hiện đối tượng

- OpenCV: Xử lý hình ảnh

**4. CHỨC NĂNG CHÍNH**

**4.1 Xác thực người dùng**

**4.2 Phát hiện đối tượng**

- Upload hình ảnh, video

- YOLO detection với bounding boxes

- Hiển thị kết quả real-time

**4.3 Quản lý lịch sử**

- Lưu trữ kết quả phát hiện

- Hiển thị lịch sử upload

- Phân quyền theo user

**KẾT LUẬN**

Dự án hoàn thành tốt các yêu cầu cơ bản về phát hiện đối tượng và quản lý người dùng. Kiến trúc ổn định, có khả năng mở rộng, phù hợp cho môi trường phát triển và có thể triển khai production sau khi cải thiện bảo mật.