

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI  
KHOA: CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: KHOA HỌC MÁY TÍNH**  
**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG THEO DÕI PHƯƠNG TIỆN**  
**GIAO THÔNG TỪ NHIỀU NGUỒN CAMERA**

**GVHD: TS. Nguyễn Mạnh Cường**

**Sinh viên thực hiện: Đinh Minh Đại**

**Mã sinh viên: 2020604890**

**Lớp: KHMT02 – K15**

Hà Nội – Năm 2024

## MỤC LỤC

LỜI CẢM ƠN .....	i
LỜI NÓI ĐẦU .....	ii
DANH MỤC HÌNH VẼ .....	iv
DANH MỤC BẢNG BIÊU .....	vii
DANH MỤC TỪ NGỮ VIẾT TẮT .....	viii
CHƯƠNG 1. PHÁT BIỂU BÀI TOÁN .....	1
1.1. Giới thiệu về thị giác máy tính .....	1
1.2. Giới thiệu bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera .....	5
1.2.1. Giới thiệu khái quát .....	5
1.2.2. Phân tích bài toán .....	7
1.2.3. Mô tả dữ liệu đầu vào, đầu ra của bài toán .....	10
1.2.4. Ứng dụng .....	11
CHƯƠNG 2. MỘT SỐ KỸ THUẬT HIỆN CÓ VÀ GIẢI PHÁP ĐỀ XUẤT .....	13
2.1. Kỹ thuật cho bài toán nhận diện đối tượng .....	13
2.1.1. Phương pháp dựa trên vùng đề xuất .....	13
2.1.2. Phương pháp không dựa vào vùng đề xuất .....	16
2.2. Kỹ thuật cho bài toán theo dõi đối tượng .....	18
2.2.1. Phương pháp SORT .....	19
2.2.2. Phương pháp DeepSORT .....	21
2.2.3. Phương pháp ByteTrack .....	22
2.3. Kỹ thuật cho bài toán tái nhận diện .....	24
2.3.1. Mô hình VehicleNet .....	25
2.3.2. Mô hình Resnet IBN .....	27
2.3.3. Mô hình ResNeXt .....	29
2.4. Kỹ thuật cho bài toán liên kết đối tượng giữa các camera .....	32
2.4.1. Tìm các phương tiện ứng cử .....	32

2.4.2. Tính toán ma trận khoảng cách $D$ giữa đặc trưng của các box .....	33
2.4.3. Liên kết phương tiện sử dụng k-láng giềng tương hỗ .....	36
2.5. Giải pháp đề xuất cho bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera .....	37
<b>CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM .....</b>	<b>39</b>
3.1. Kết quả huấn luyện mô hình ResNext-50 (32x4d).....	39
3.1.1. Dữ liệu huấn luyện .....	39
3.1.2. Kết quả huấn luyện mô hình.....	41
3.2. Kết quả thực nghiệm trên bộ dữ liệu AI City Challenge .....	44
3.2.1. Giới thiệu bộ dữ liệu.....	44
3.2.2. Kết quả thực nghiệm.....	48
<b>CHƯƠNG 4. SẢN PHẨM DEMO .....</b>	<b>56</b>
4.1. Giới thiệu về công cụ Tkinter.....	56
4.2. Phân tích hệ thống .....	58
4.2.1. Biểu đồ use case tổng quát .....	58
4.2.2. Mô tả chi tiết các use case .....	58
4.3. Giao diện hệ thống.....	61
4.4. Các chức năng của hệ thống.....	62
4.4.1. Chức năng theo dõi phương tiện giao thông từ nhiều camera .....	62
4.4.2. Chức năng tìm kiếm phương tiện giao thông .....	63
<b>KẾT LUẬN .....</b>	<b>66</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>67</b>

## LỜI CẢM ƠN

Trước khi đến với nội dung chính của bản báo cáo đồ án tốt nghiệp này, em xin được gửi lời cảm ơn đến thầy Nguyễn Mạnh Cường khoa công nghệ thông tin. Bản báo cáo này được hoàn thiện hơn với sự giúp đỡ, đóng góp không nhỏ của thầy. Các đóng góp, ý kiến của thầy luôn phản ánh đúng được các vấn đề tồn đọng trong bản báo cáo. Từ các đóng góp, ý kiến của thầy, em đã rút ra được những kinh nghiệm khi thực hiện bản báo cáo và hoàn thành bản báo cáo này với mức độ hoàn chỉnh tốt hơn.

Bên cạnh đó, em cũng gửi lời cảm ơn đến những người đã giúp đỡ, đóng góp để có một bản báo cáo hoàn chỉnh. Cảm ơn anh Trần Văn Gạo và Nguyễn Thành Nam ở bên công ty Samsung R&D đã cung cấp bộ dữ liệu và đưa ra những góp ý để hoàn chỉnh cho bản báo cáo này.

Trong quá trình làm chắc chắn khó tránh khỏi những thiếu sót. Do đó, kính mong nhận được những lời góp ý của thầy/cô để bản báo cáo này ngày càng hoàn thiện hơn.

Em xin trân thành cảm ơn!

Sinh viên thực hiện

*Đại*

*Đinh Minh Đại*

## LỜI NÓI ĐẦU

Thị giác máy tính và trí tuệ nhân tạo là hai lĩnh vực công nghệ đang trải qua sự biến đổi đột phá, đặc biệt trong thời kỳ số hóa mạnh mẽ và sự phát triển không ngừng của công nghệ thông tin. Sự hình thành của hai lĩnh vực đã mở ra một loạt các cơ hội mới và thách thức thú vị, ảnh hưởng mạnh mẽ đến nhiều khía cạnh của cuộc sống hàng ngày.

Tính đến hiện nay, lĩnh vực thị giác máy tính đã tiến xa hơn bao giờ hết trong giải quyết các tác vụ, yêu cầu khác nhau như nhận diện người, nhận diện xe cộ, v.v. Không dừng lại ở đó, các công nghệ, kỹ thuật ngày càng được hoàn thiện nhằm cải tiến, giải quyết các yêu cầu của bài toán. Việc tiếp tục phát triển công nghệ tạo ra một cơ hội đột phá trong việc cải thiện cuộc sống và công việc, từ tăng cường an ninh cá nhân đến cải thiện quy trình sản xuất và loại bỏ công việc lặp đi lặp lại.

Bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera là chủ đề đang được thịnh hành trong thời gian gần đây với cuộc thi như AI City Challenge. Lý do cho sự phổ biến là nhờ tiến bộ lớn của lĩnh vực thị giác máy tính trong những năm gần đây, đặc biệt sau sự phát triển của học sâu (deep learning) và đặc biệt là mạng nơ-ron tích chập (CNN) đã mang lại kết quả vượt trội so với các phương pháp cũ. Điều này mở ra khả năng phát triển, ứng dụng rộng rãi hơn cho bài toán vào cuộc sống và cùng là trọng tâm của báo cáo này.

Với mong muốn giải quyết bài toán và đưa ra những ứng dụng trong thực tiễn đó là tìm kiếm phương tiện trong các camera giám sát, em quyết định đã lựa chọn đề tài “**Xây dựng hệ thống theo dõi phương tiện giao thông từ nhiều nguồn camera**”.

Trong bản báo cáo này sẽ gồm có 4 chương, mỗi chương sẽ tập trung trình bày về một vấn đề cụ thể. Nội dung chính của các chương là:

- **Chương 1: Phát biểu bài toán**

Chương này sẽ trình bày tóm tắt về thị giác máy tính, giới thiệu, mô tả chung về bài toán cũng như những bài toán liên quan để giải quyết và một số ứng dụng phổ biến của bài toán.

- **Chương 2: Một số kỹ thuật hiện có và giải pháp đề xuất**

Chương 2 sẽ trình bày và phân tích ưu, nhược điểm một số kỹ thuật có thể áp dụng để giải bài toán, đặc biệt là các bài toán con bên trong.

- **Chương 3: Kết quả thực nghiệm**

Báo cáo kết quả thực nghiệm sử dụng kỹ thuật các kỹ thuật để giải quyết bài toán. Ngoài ra, đánh ra các kết quả để chọn ra kỹ thuật tốt nhất để thực hiện áp dụng xây dựng hệ thống.

- **Chương 4: Sản phẩm demo**

Nội dung của chương sẽ trình bày về sản phẩm demo, các công cụ để xây dựng sản phẩm, các giao diện của sản phẩm và kết quả

Hi vọng thông qua 4 chương của báo cáo này, em sẽ làm rõ được bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera, một số kỹ thuật có thể sử dụng để giải quyết bài toán và đưa ra ứng dụng cụ thể trong thực tiễn.

## DANH MỤC HÌNH VẼ

Hình 1.1 So sánh giữa thị giác máy tính và con người .....	2
Hình 1.2 Minh họa các tác vụ trong thị giác máy tính.....	3
Hình 1.3 Ví dụ về thị giác máy tính trong chẩn đoán .....	4
Hình 1.4 Minh họa cho bài toán theo dõi phương tiện từ một camera .....	5
Hình 1.5 Ảnh chụp từ nhiều góc độ khác nhau trên một đoạn đường .....	6
Hình 1.6 Các bài toán con của bài toán “Theo dõi phương tiện giao thông từ nhiều nguồn camera”.....	7
Hình 1.7 Ví dụ về nhận diện cầu thủ trên sân bóng.....	8
Hình 1.8 Ví dụ về bài toán theo dõi .....	8
Hình 1.9 Ví dụ về bài toán tái nhận diện người .....	9
Hình 1.10 Ví dụ về cảnh giao thông từ nhiều camera.....	10
Hình 2.1 Ví dụ minh họa cho kỹ thuật selective search.....	14
Hình 2.2 Minh họa cho phương pháp Fast R-CNN .....	15
Hình 2.3 Minh họa cho phương pháp Faster R-CNN .....	16
Hình 2.4 Minh họa cho phương pháp YOLO .....	17
Hình 2.5 Ví dụ về độ đo IOU .....	18
Hình 2.6 Luồng hoạt động của phương pháp SORT .....	19
Hình 2.7 Minh họa phương pháp DeepSORT .....	21
Hình 2.8 Sơ đồ luồng hoạt động của ByteTrack .....	22
Hình 2.9 Minh họa phương pháp ByteTrack .....	23
Hình 2.10 Minh họa mô hình VehicleNet.....	26
Hình 2.11 Ví dụ về dữ liệu miền thực và miền ảo .....	27
Hình 2.12 Sơ đồ các khối IBN .....	28
Hình 2.13 So sánh giữa hai khối của hai mô hình ResNet và ResNext.....	30
Hình 2.14 So sánh giữa ResNet-50 và ResNeXt-50 (32x4d) .....	31
Hình 2.15 Vùng được định nghĩa trước trong camera thứ 42 và 43 .....	32
Hình 2.16 Mô tả trực quan quá trình liên kết đối tượng giữa các camera .....	37

Hình 2.17 Sơ đồ hệ thống cho bài toán .....	38
Hình 3.1 Minh họa cho bộ dữ liệu VeRi-776 .....	39
Hình 3.2 Biểu đồ thể hiện sự phân bố của các phương tiện trong tập train của bộ dữ liệu VeRi-776 .....	40
Hình 3.3 Biểu đồ thể hiện sự phân bố của các phương tiện trong tập test của bộ dữ liệu VeRi-776 .....	40
Hình 3.4 Minh họa quy trình kiểm định K-fold.....	41
Hình 3.5 Kết quả loss trong quá trình kfold mô hình ResNext-50 (32x4d) .....	42
Hình 3.6 Kết quả accuracy trong quá trình kfold mô hình ResNext-50 (32x4d)43	
Hình 3.7 Kết quả accuracy và loss trong quá trình huấn luyện mô hình ResNext-50 (32x4d) .....	44
Hình 3.8 Sự bố trí các camera trên bản đồ của tập S01 .....	46
Hình 3.9 Sự bố trí các camera trên bản đồ của tập S02 .....	46
Hình 3.10 Sự bố trí các camera trên bản đồ của các tập S03, S04 và S05 .....	47
Hình 3.11 Sự bố trí các camera trên bản đồ của tập S06 .....	47
Hình 3.12 Cảnh của các camera thuộc tập S03.....	48
Hình 3.13 Kết quả theo dõi trong một khung hình trên camera 11 .....	49
Hình 3.14 Các vùng được đánh dấu trên các camera thuộc tập S03.....	49
Hình 3.15 Kết quả liên kết giữa các camera của các mô hình trên IDF1, IDP, IDR sử dụng Re-ranking với nhiều tham số k khác nhau .....	51
Hình 3.16 Kết quả liên kết giữa các camera của các mô hình trên IDF1, IDP, IDR sử dụng Euclid với nhiều tham số k khác nhau.....	52
Hình 3.17 Kết quả liên kết giữa các camera của các mô hình trên IDF1, IDP, IDR sử dụng Cosine với nhiều tham số k khác nhau .....	53
Hình 4.1 Sơ đồ use case tổng quát .....	58
Hình 4.2 Màn hình chức năng theo dõi phương tiện giao thông từ nhiều camera .....	61
Hình 4.3 Màn hình thực hiện tìm kiếm phương tiện giao thông.....	62
Hình 4.4 Kết quả theo dõi trên từng camera ở dạng txt.....	63

Hình 4.5 Màn hình kết quả cho chức năng tìm kiếm phương tiện giao thông khi tìm thấy phương tiện (1).....	64
Hình 4.6 Màn hình kết quả cho chức năng tìm kiếm phương tiện giao thông khi tìm thấy phương tiện (2).....	64
Hình 4.7 Màn hình kết quả cho chức năng tìm kiếm phương tiện giao thông khi không tìm thấy phương tiện .....	65

## DANH MỤC BẢNG BIỂU

Bảng 2-1 Kết quả ByteTracking trích từ tài liệu tham khảo [6] .....	24
Bảng 3-1 Mô tả tập dữ liệu con trong bộ dữ liệu AI City Challenge .....	45
Bảng 3-2 Kết quả sử dụng mô hình ResNet101-IBN liên kết camera trên tập S03 với số lần lặp là 10, k=15 .....	54
Bảng 3-3 Kết quả sử dụng mô hình ResNet50 liên kết camera trên tập S03 với số lần lặp là 10, k=14 .....	54
Bảng 3-4 Kết quả sử dụng mô hình ResNeXt50 liên kết camera trên tập S03 với số lần lặp là 10, k=14 .....	55

## DANH MỤC TỪ NGỮ VIẾT TẮT

CNN	Convolutional Neural Network
NMS	Non-max Suppression
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
R-CNN	Regional-Convolutional Neural Network
YOLO	You Only Look Once
Re-ID	Re-identification
BN	Batch Normalization
IB	Instance Normalization

## CHƯƠNG 1. PHÁT BIỂU BÀI TOÁN

Nội dung của chương này sẽ trình bày về tổng quan về thị giác máy tính và chi tiết về bài toán bài toán “*Theo dõi phương tiện giao thông từ nhiều nguồn camera*”. Lý do là bài toán trên là một bài toán thuộc lĩnh vực thị giác máy tính. Do đó, có cái nhìn tổng quan về thị giác máy tính sẽ giúp cho việc phân tích bài toán trở nên rõ ràng hơn.

### 1.1. Giới thiệu về thị giác máy tính

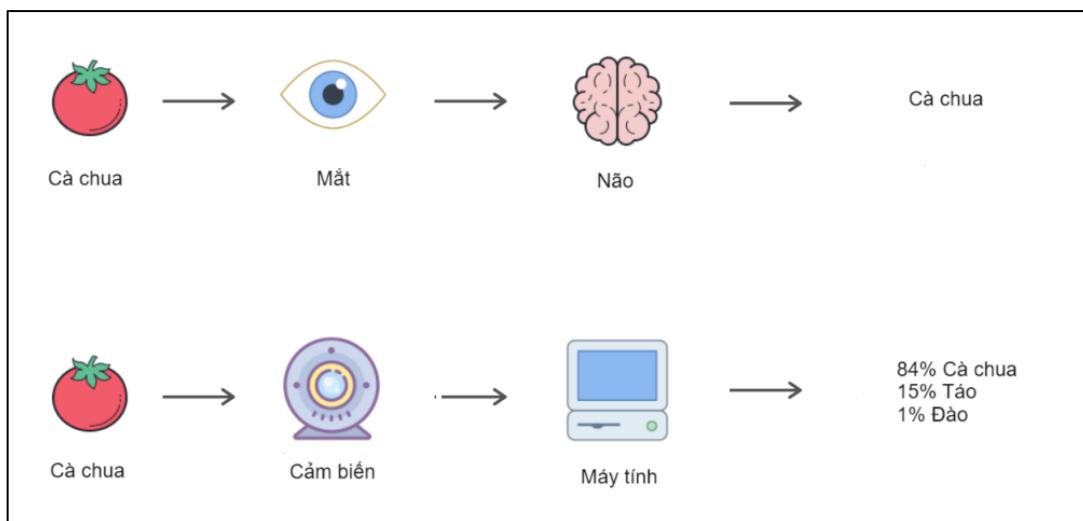
**Thị giác máy tính** là một lĩnh vực trí tuệ nhân tạo cho phép máy tính và hệ thống lấy được thông tin có ý nghĩa từ dữ liệu dạng hình ảnh kỹ thuật số - đồng thời thực hiện hành động hoặc đưa ra đề xuất dựa trên thông tin đó. Cụ thể hơn, nhiệm vụ thị giác máy tính bao gồm các phương pháp thu thập, xử lý, phân tích và hiểu hình ảnh kỹ thuật số cũng như trích xuất dữ liệu nhiều chiều từ thế giới thực để tạo ra thông tin số hoặc ký hiệu. Các thông tin số là đại lượng có thể được đo lường hoặc đếm, ví dụ: Tọa độ của một vật trong ảnh, số lượng các vật được phát hiện, v.v. Thông tin dạng ký hiệu là đại lượng được biểu diễn bởi kí hiệu hoặc danh mục, các thông tin đó có thể là nhãn gán cho đối tượng, mô tả của ảnh v.v. Tóm lại, nếu trí tuệ nhân tạo cho phép máy tính suy nghĩ thì thị giác máy tính cho phép máy tính nhìn, quan sát và hiểu.

**Nguyên lý** của thị giác máy tính tương đối giống như thị giác của con người, chính xác hơn là bắt chước thị giác của con người. Máy tính tiếp nhận thông tin dạng hình ảnh về thế giới thông qua các thiết bị cảm biến và các thông tin sẽ được xử lý tại bộ phận xử lý trung tâm. Cách thức trên khá giống với thị giác của con người, đều có bộ phận tiếp nhận thông tin và bộ phận xử lý.

Tuy nhiên, thị giác của con người có lợi thế vượt trội hơn là trải qua quá trình tiến hóa sinh học và quá trình học hỏi không ngừng mỗi ngày. Điều này làm cho thị giác của con người trở nên nhạy bén hơn do có kinh nghiệm tích lũy từ cuộc sống. Ngoài ra, thị giác của con người có khả năng phân tích và hiểu

ngữ cảnh, nhận diện vật thể từ nhiều góc độ, ánh sáng khác nhau – điều mà thị giác máy tính chưa thể hiện tốt.

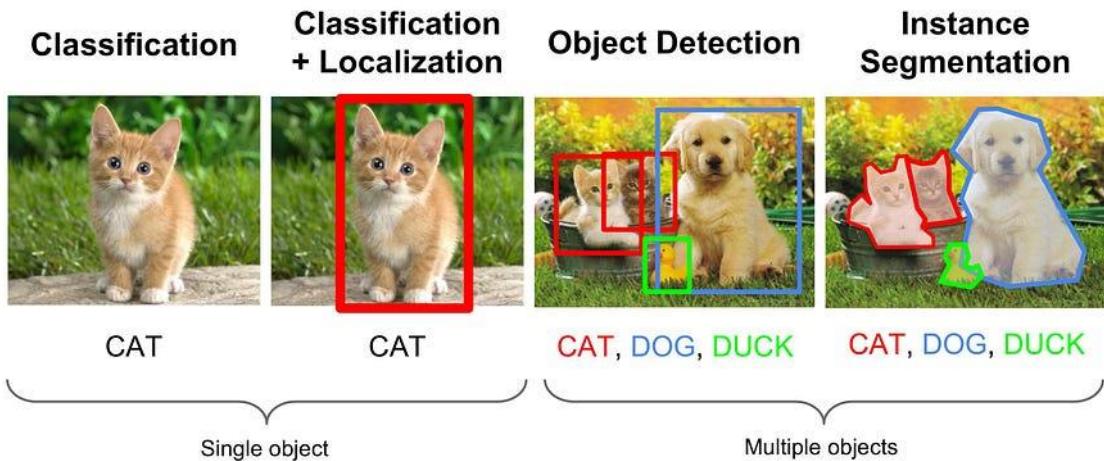
Song, thị giác máy tính lại có khả năng xử lý một cách nhanh chóng, vượt trội ở các tác vụ đếm các đối tượng trong ảnh. Như vậy, thị giác máy trở nên khả thi là có những hiểu biết về thị giác của con người. Mặc dù chưa thể hiện tốt như thị giác của con người nhưng có những tác vụ mà thị giác máy tính thể hiện vượt trội hoàn toàn so với con người.



*Hình 1.1 So sánh giữa thị giác máy tính và con người*

Trong thị giác máy tính, có rất nhiều các tác vụ được đề ra nhằm giải quyết một yêu cầu xác định. Một số các tác vụ phổ biến có thể kể đến có trong thị giác máy tính là: *Phân loại ảnh* (Image Classification), *chọn vùng đối tượng* (Object Localization), *phát hiện đối tượng* (Object Detection), *phân đoạn ảnh* (Semantic Segmentation), *theo dõi* (Tracking), v.v.

Đây là các tác vụ những tác vụ cơ bản trong lĩnh vực thị giác máy tính và các tác vụ này có thể được giải quyết bằng nhiều các kỹ thuật khác nhau, trong đó Học sâu (Deep learning) và Mạng nơ-ron tích chập (CNNs) là hai kỹ thuật thiết yếu.



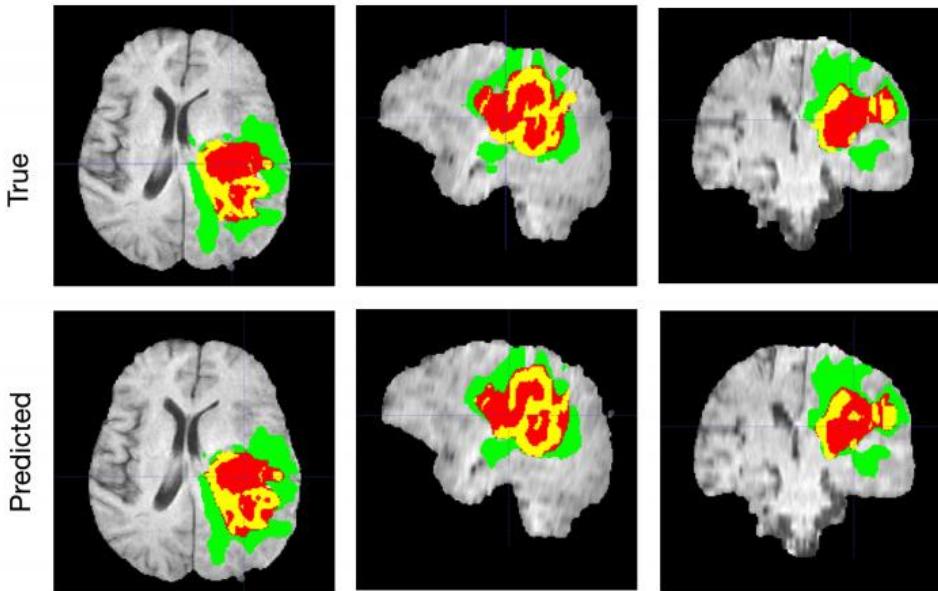
*Hình 1.2 Minh họa các tác vụ trong thị giác máy tính*

(*Nguồn: <https://www.mdpi.com/2227-7080/9/1/2/htm#>*)

Tính đến thời điểm hiện tại, thị giác máy tính đã phát triển vượt bậc và được ứng dụng trong nhiều lĩnh vực khác nhau như:

- Trong lĩnh vực sản xuất, thị giác máy tính được sử dụng cho việc giám sát chất lượng, quản lý từ xa và tự động hóa hệ thống.
- Trong lĩnh vực y tế, các quá trình phân tích hình ảnh X-ray, MRI nhằm chẩn đoán được thực hiện một cách tự động nhờ có thị giác máy tính và đem lại kết quả tốt.
- Trong việc bảo mật, các hệ thống nhận dạng, phát hiện các đối tượng xâm nhập một cách tự động trở nên khả thi là nhờ có thị giác máy tính.

Ngoài ra, vô số các ứng dụng khác của thị giác máy tính có thể được đưa vào trong đời sống hằng ngày, cải thiện, nâng cao chất lượng cuộc sống.



*Hình 1.3 Ví dụ về thị giác máy tính trong chẩn đoán*

(*Nguồn: <https://www.altexsoft.com/blog/computer-vision-healthcare/>*)

Mặc dù đã có những sự phát triển vượt bậc nhất định và được ứng dụng nhiều trong đời sống nhưng thị giác máy tính vẫn còn những thử thách, khó khăn nhất định. Ở thời điểm hiện tại giới hạn về chất bán dẫn đang dần được thể hiện rõ nét, hiệu năng của các vi xử lý thế hệ tăng không nhiều so với thế hệ cũ. Một khó khăn khác là thiếu nguồn dữ liệu huấn luyện cả về chất lượng lẫn số lượng do thị giác máy tính cần rất nhiều dữ liệu để học. Ngoài ra còn nhiều những khó khăn khác về tối ưu hóa thuật toán thu thập, lưu trữ và xử lý dữ liệu phục vụ cho mục đích huấn luyện. Tóm lại, thị giác máy tính vẫn còn nhiều những thử thách và cần có giải pháp mang tính toàn diện.

Nhìn chung, thị giác máy tính là một lĩnh vực phát triển nhanh chóng với nhiều ứng dụng vào cuộc sống và nhiều những cơ hội cải tiến, thúc đẩy công nghệ. Với việc tiếp tục nghiên cứu và phát triển, hoàn toàn có thể mong đợi được thấy những ứng dụng thú vị hơn nữa về thị giác máy tính trong tương lai.

## 1.2. Giới thiệu bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera

### 1.2.1. Giới thiệu khái quát

Bài toán “**Theo dõi phương tiện giao thông từ nhiều nguồn camera**” là một bài toán đang được dành sự quan tâm lớn trong khoảng thời gian gần đây. Lý do nằm ở các phương pháp học máy thủ công truyền thống chưa đủ phức tạp để thực hiện bài toán và cũng nằm ở giới hạn của phần cứng khi phải xử lý dữ liệu dạng video, một dạng dữ liệu chứa nhiều thông tin phức tạp. Cho đến khi phương pháp học sâu phát triển và sữ mạnh phần cứng được cải tiến thì việc thực hiện bài toán trở nên khả thi.

Có thể thấy, bài toán trên là một bài toán mở rộng của bài toán “**Theo dõi phương tiện giao thông từ một camera**”. Các yêu cầu cơ bản của bài toán theo dõi đối tượng trong một camera có thể kể đến là:

- Nhận diện được các phương tiện giao thông trong khung hình của video.
- Theo dõi/dánh dấu phương tiện giao thông trên của khung hình của video.
- Phân biệt giữa các phương tiện giao thông trong khung hình với nhau.

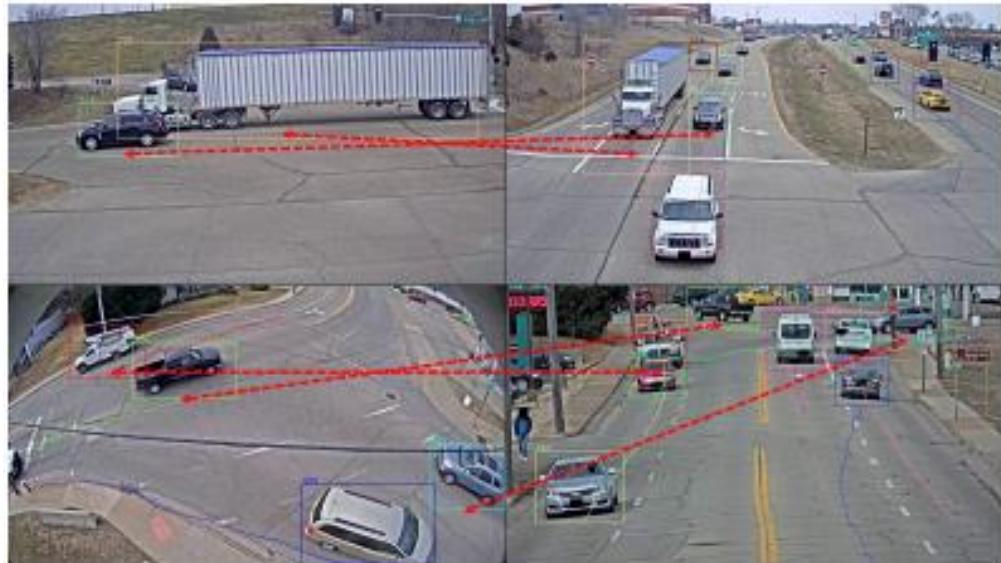


*Hình 1.4 Minh họa cho bài toán theo dõi phương tiện từ một camera*

(Nguồn: <https://blog.roboflow.com/what-is-object-tracking-computer-vision/>)

Tuy nhiên, khác với việc theo dõi phương tiện giao thông từ một nguồn camera, việc theo dõi phương tiện giao thông từ nhiều nguồn camera có yêu cầu phức tạp hơn. Một số yêu cầu có thể kể đến như:

- Nhận diện được các phương tiện giao thông đã từng xuất hiện ở một camera bất kỳ trước đó thay vì xác định đó là một phương tiện giao thông mới.
- Xác định đường đi của phương tiện giao thông giữa, tức là thứ tự xuất hiện ở các camera
- Khả năng mở rộng bài toán với nhiều camera hơn mà không làm ảnh hưởng đến kết quả và hiệu suất của hệ thống.



*Hình 1.5 Ảnh chụp từ nhiều góc độ khác nhau trên một đoạn đường*

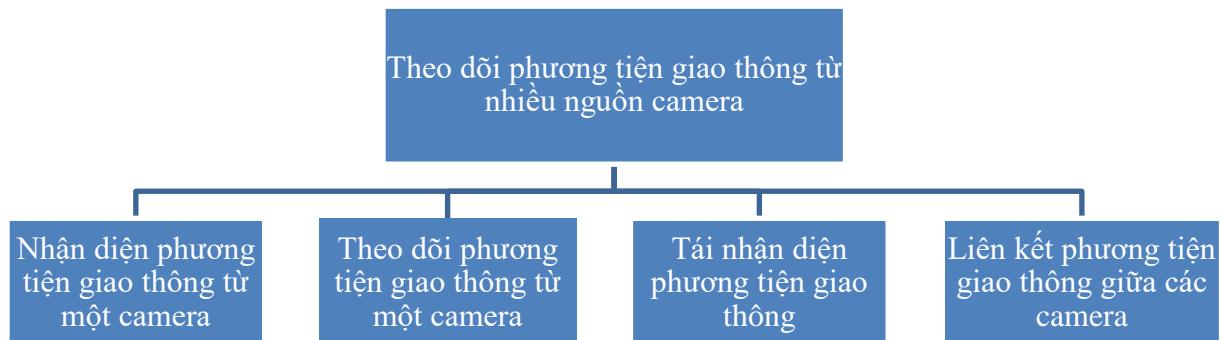
*(Nguồn: <https://paperswithcode.com/dataset/cityflow>)*

Những yêu cầu trên không hoàn toàn mới lạ mà đã xuất hiện ở bài toán liên quan đến theo dõi người từ nhiều nguồn camera nói riêng và theo dõi đối tượng từ nhiều nguồn camera nói chung. Do đó, các phương pháp, hướng tiếp cận để thực hiện bài toán đã sẵn có và có thể áp dụng để giải quyết bài toán.

### 1.2.2. Phân tích bài toán

Thông thường, các bài toán theo dõi đối tượng từ nhiều camera trên được chia thành hai bước đó là: ***Theo dõi đối tượng từ một camera*** và ***liên kết giữa các camera***. Đối với bài toán theo dõi phương tiện giao thông từ một camera, từ các yêu cầu từ bài toán thì có thể thấy rằng các bài toán con cần phải giải quyết là ***phát hiện phương tiện giao thông*** và ***theo dõi phương tiện giao thông***. Còn đối với bài toán liên kết các phương tiện giao thông giữa các camera, các bài toán con cần phải giải quyết là ***tái nhận diện phương tiện giao thông*** và ***liên kết các phương tiện giao thông giữa các camera***.

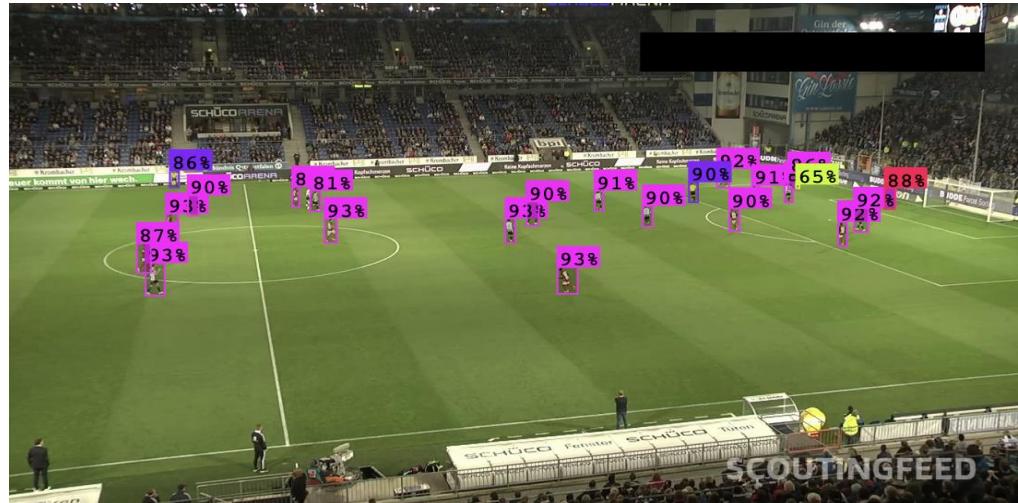
Như vậy, từ bài toán “***Theo dõi phương tiện giao thông từ nhiều nguồn camera***” ban đầu, các bài toán con cần phải giải quyết được thể hiện ở hình dưới đây:



*Hình 1.6 Các bài toán con của bài toán “Theo dõi phương tiện giao thông từ nhiều nguồn camera”*

Ứng với mỗi bài toán trên, đều có các bài toán tổng quát là các bài toán: ***nhận diện đối tượng*** (Object detection), ***theo dõi đối tượng*** (Object tracking), ***tái nhận diện đối tượng*** (Re-identification), ***liên kết giữa các camera*** (Inter-camera association). Với mô tả ngắn gọn của các bài toán như sau:

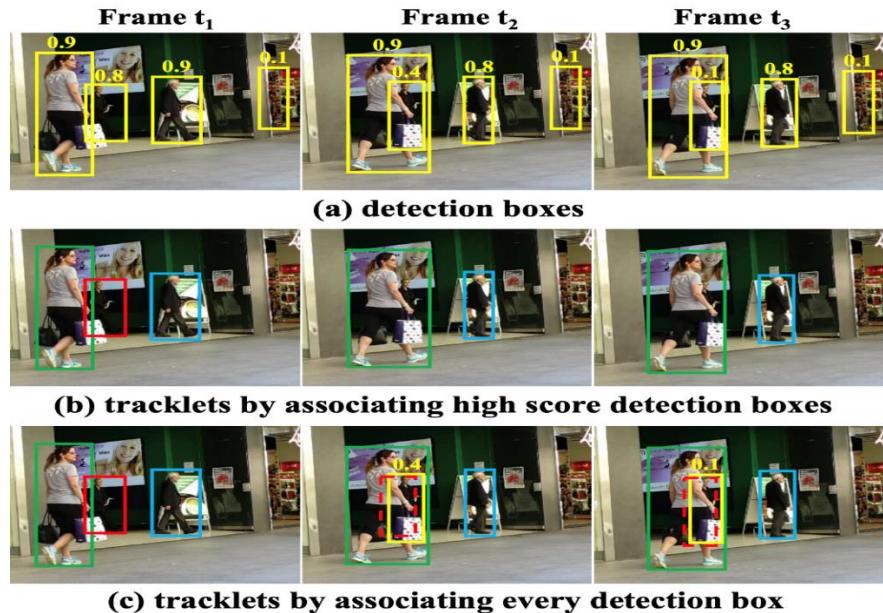
- ***Nhận diện đối tượng*** (Object detection): Nhận diện đối tượng đóng một vai trò trung tâm, với mục tiêu làm cho máy tính có khả năng hiểu và diễn giải thông tin từ hình ảnh hay video giống như con người. Làm cho máy tính có khả năng nhận diện đối tượng trong dữ liệu đầu vào.



Hình 1.7 Ví dụ về nhận diện cầu thủ trên sân bóng

(Nguồn: <https://blog.roboflow.com/object-detection/>)

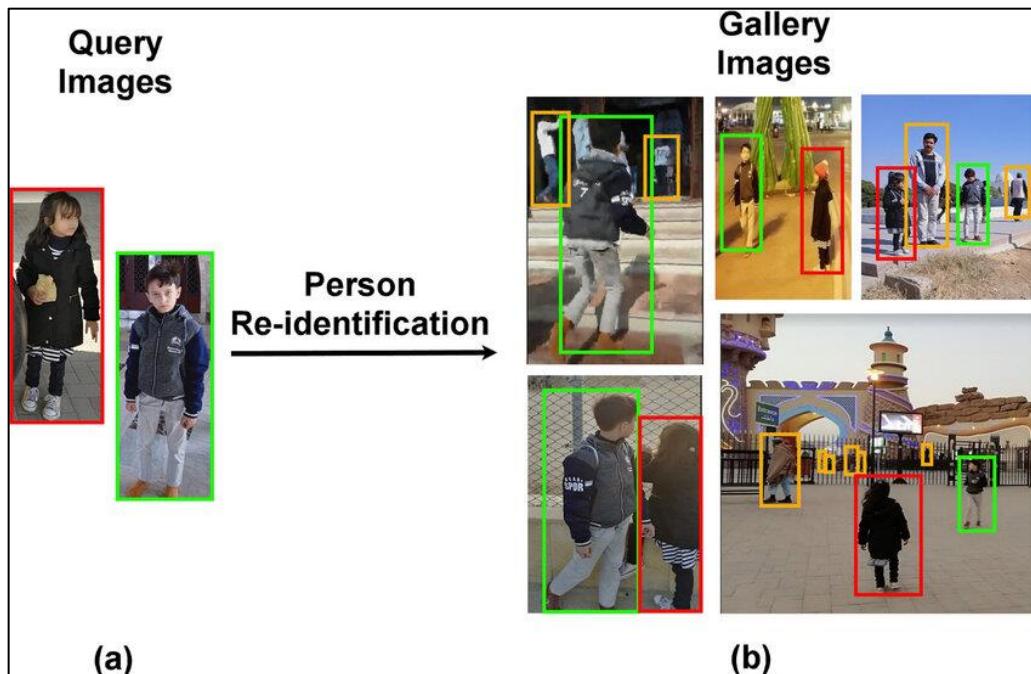
- *Theo dõi đối tượng* (Object tracking): Theo dõi đối tượng liên quan đến việc xác định vị trí và theo dõi chuyển động của các đối tượng qua thời gian trong một chuỗi video. Đây là một thách thức đặc biệt vì việc theo dõi đòi hỏi sự chính xác cao trong việc phát hiện và dự đoán vị trí của đối tượng trong các điều kiện đa dạng và thay đổi liên tục.



Hình 1.8 Ví dụ về bài toán theo dõi

(Nguồn: <https://github.com/ifzhang/ByteTrack>)

- *Tái nhận diện* (re-identification): Tái nhận diện trong thị giác máy tính là một tác vụ phức tạp nhằm mục đích nhận ra các đối tượng trên các cảnh hoặc chế độ xem camera khác nhau. Nói cách khác, mỗi một đối tượng sẽ có một ID nhất định, khi gặp lại đối tượng này sau một thời gian đối tượng đó không xuất hiện hoặc trong một bức ảnh, hoàn cảnh khác thì phải cho ra ID ban đầu thay vì gán ID mới cho đối tượng.



*Hình 1.9 Ví dụ về bài toán tái nhận diện người*

(*Nguồn: [https://www.researchgate.net/figure/An-intelligent-person-re-identification-system-identifies-different-people-across\\_fig1\\_351761936](https://www.researchgate.net/figure/An-intelligent-person-re-identification-system-identifies-different-people-across_fig1_351761936)*)

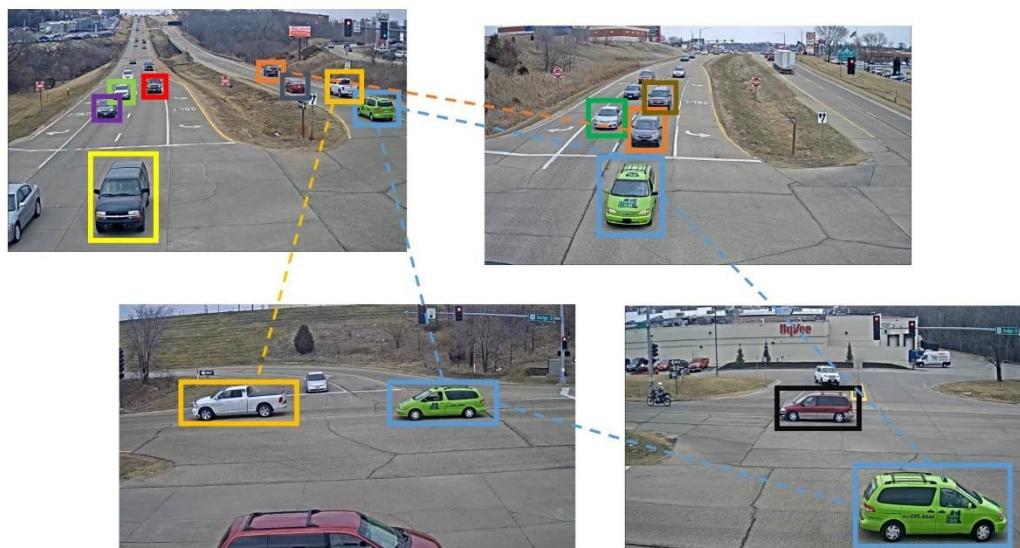
- *Liên kết giữa các camera* (Inter-camera association): Là quá trình liên kết các đối tượng giữa các camera khi đối tượng đó di chuyển từ một camera sang một camera khác. Có thể nói, bài toán mở rộng của bài toán tái nhận diện đối tượng. Yêu cầu của bài toán cao hơn so với việc tái nhận diện là phải thực hiện liên kết các đối tượng giữa các camera, tạo thành một đối tượng toàn cục.

Như vậy, bài toán “**Theo dõi phương tiện giao thông từ nhiều nguồn camera**” yêu cầu nhận diện được các phương tiện giao thông, theo dõi các phương tiện giao thông, tái nhận diện phương tiện giao thông và liên kết các

phương tiện giao thông giữa các camera. Việc nhận diện phải nhận diện được các phương tiện giao thông có trong ảnh và phân biệt các đối tượng không phải phương tiện giao thông. Việc theo dõi yêu cầu phải đưa ra đường đi cũng như tọa độ của các phương tiện giao thông. Cuối cùng, việc tái nhận diện và liên kết phương tiện giao thông yêu cầu đưa ra một phương tiện đã từng xuất hiện trước đó và thực hiện liên kết.

### 1.2.3. Mô tả dữ liệu đầu vào, đầu ra của bài toán

Với các yêu cầu về bài toán đã trình bày ở trên thì dữ liệu đầu vào theo yêu cầu sẽ là những video được trích xuất từ camera trên đường giao thông. Những video có thể là từ một địa điểm cụ thể nhưng nhiều góc quay khác nhau hoặc là từ nhiều những địa điểm khác nhau nhưng có liên kết giao thông.



*Hình 1.10 Ví dụ về cảnh giao thông từ nhiều camera*

(*Nguồn: <https://www.mdpi.com/2079-9292/11/7/1008>*)

Kết quả trả về khi qua hệ thống theo dõi phương tiện giao thông sẽ bao gồm về tọa độ của phương tiện trong ảnh (bounding box), id của phương tiện giao thông, tọa độ trong ảnh mà phương tiện đã đi qua giống như ở Hình 1.10.

#### 1.2.4. Ứng dụng

Việc theo dõi phương tiện giao thông từ nhiều nguồn camera có nhiều ứng dụng trong lĩnh vực giao thông. Nếu chỉ xét riêng việc theo dõi từ một nguồn camera thì ứng khả năng ứng dụng đã rất rộng mở. Dưới đây là một số ứng dụng có thể dễ dàng nhận thấy:

- **Giám sát và quản lý giao thông:** Sử dụng hệ thống theo dõi nhiều camera để giám sát luồng giao thông, phát hiện tắc nghẽn và quản lý tín hiệu giao thông để cải thiện khả năng di chuyển trong đô thị. Nâng cao hiệu quả của mạng lưới giao thông bằng cách phân tích chuyển động của phương tiện, tối ưu hóa tuyến đường và giảm thời gian di chuyển.
- **Hệ thống đường cao tốc thông minh:** Triển khai tính năng theo dõi phương tiện giữa các khu vực trên đường cao tốc để tăng cường các biện pháp an toàn, quản lý tốc độ và hỗ trợ công nghệ lái xe tự động.
- **Thực thi pháp luật và an ninh:** Hỗ trợ nhận dạng và theo dõi các phương tiện vì mục đích an ninh, bao gồm giám sát các hoạt động đáng ngờ hoặc thực thi luật giao thông. Đặc biệt là theo dõi một phương tiện cụ thể nào đó.
- **Quy hoạch đô thị:** Hỗ trợ các nhà quy hoạch thành phố hiểu mô hình giao thông và đưa ra quyết định sáng suốt về phát triển và sửa đổi cơ sở hạ tầng.
- **Phân tích đám đông:** Trong các sự kiện hoặc không gian công cộng, phân tích luồng phương tiện để quản lý đám đông và đảm bảo an toàn công cộng.

Trong thực tế, việc triển khai hệ thống theo dõi phương tiện đòi hỏi phải xem xét cẩn thận vị trí đặt camera vì trường nhìn và góc có thể ảnh hưởng đáng kể đến hiệu suất của hệ thống. Ví dụ: Camera đặt tại các giao lộ có thể cung cấp kết quả theo dõi khác so với camera trên đường cao tốc để đo tốc độ và hướng

của các phương tiện khác nhau. Hơn nữa, điều kiện thời tiết cũng có thể ảnh hưởng đến độ chính xác của việc theo dõi.

Tóm lại, theo dõi phương tiện từ nhiều camera trong thị giác máy tính là một lĩnh vực năng động và đang phát triển, kết hợp các thuật toán phát hiện và theo dõi tiên tiến với thiết kế mạng camera chiến lược để tạo ra các hệ thống thông minh cho nhu cầu an ninh và giao thông hiện đại.

## CHƯƠNG 2. MỘT SỐ KỸ THUẬT HIỆN CÓ VÀ GIẢI PHÁP ĐỀ XUẤT

Như đã trình bày ở Chương 1 về các bài toán con được chia ra từ bài toán ban đầu đó là bài toán nhận diện đối tượng, bài toán theo dõi đối tượng, bài toán tái nhận diện đối tượng và bài toán liên kết giữa các camera. Dưới đây trình bày về các kỹ thuật phổ biến trong đó một vài kỹ thuật được sử dụng và đạt được kết quả cao trong cuộc thi Ai City Challenge [1].

### 2.1. Kỹ thuật cho bài toán nhận diện đối tượng

Trước khi phương pháp học sâu được sử dụng rộng rãi, các kỹ thuật học máy cơ bản được sử dụng, cải tiến để có thể thực hiện được bài toán. Ví dụ như Scale Invariant Feature Transform (SIFT) hay Histogram of Oriented Gradient (HOG) được sử dụng rộng rãi để trích chọn các đặc trưng có trong ảnh. Sau đó, các đặc trưng thu được sẽ là đầu vào cho quá trình nhận diện các đối tượng có trong ảnh. Mặc dù đạt được một số kết quả khả quan, các phương pháp học máy đều chỉ giải quyết được một số bài toán nhất định, không có khả năng giải quyết bài toán một cách tổng quát.

Với các ưu điểm vượt trội so với các phương pháp học máy thủ công, mạng nơ ron tích chập (CNN) đã trở nên phổ biến hơn cho các tác vụ của thị giác máy tính. Đối với việc nhận diện đối tượng, có hai phương pháp chính để thực hiện, đó là **phương pháp dựa trên vùng đề xuất** (region proposal-based methods) và **phương pháp không dựa vào vùng đề xuất** (proposal-free methods).

#### 2.1.1. Phương pháp dựa trên vùng đề xuất

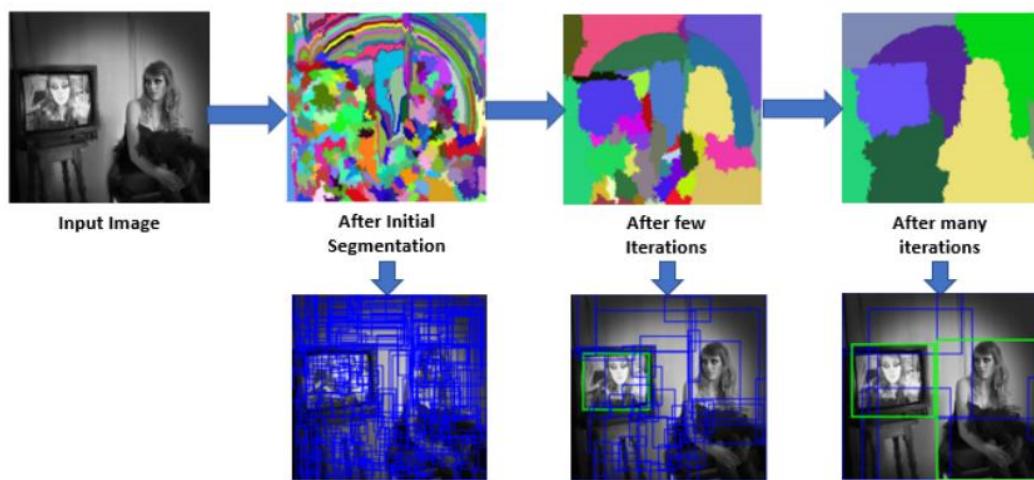
Hay còn lại là phương pháp *hai giai đoạn* (two-stage), là phương pháp nổi trội trong việc nhận diện đối tượng có trong ảnh. Với ý tưởng dựa vào kỹ thuật tìm kiếm lựa chọn (Selective search) bao gồm 2 giai đoạn:

- *Thứ nhất*, tạo ra tập các bounding box của các vùng đề xuất mà chứa toàn bộ đối tượng và loại bỏ những vùng thừa.

- *Thứ hai*, kết hợp các bounding box thành các bounding box lớn hơn.

Việc kết hợp các bounding box sẽ dựa theo các độ đo giữa hai vùng như: Sự tương đồng về màu sắc, sự tương đồng về kết cấu, v.v. Kết quả cuối cùng là các bounding box của vùng đề xuất trong ảnh ban đầu. Số các vùng đề xuất bởi thuật toán Selective search có thể lên đến 2000 vùng - một con số không nhỏ.

Sau đó, các vùng sẽ được đưa vào mô hình để phân loại như SVM, Decision Tree, v.v. Các kỹ thuật thuộc nhóm dựa trên vùng đề xuất có ưu điểm là cho ra độ chính xác cao, đặc biệt là đối với những đối tượng nhỏ trong ảnh.



*Hình 2.1 Ví dụ minh họa cho kỹ thuật selective search*

(*Nguồn: <https://www.geeksforgeeks.org/selective-search-for-object-detection-r-cnn/>*)

## Phương pháp R-CNN

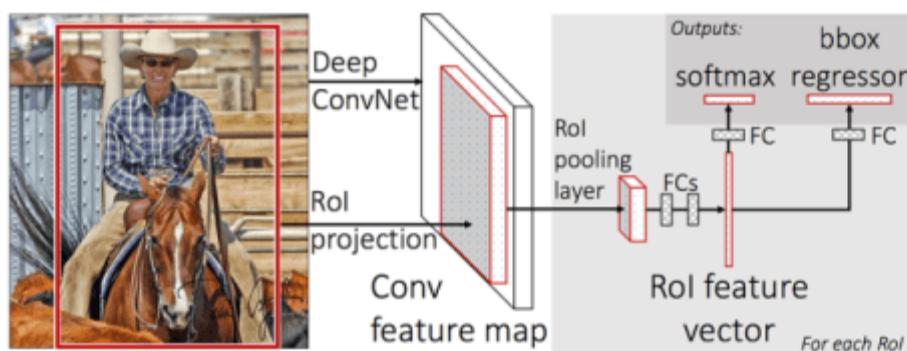
Phương pháp R-CNN khác so với phương pháp ban đầu nằm ở việc sử dụng bộ phân lớp đó là mạng nơ-ron tích chập (CNN). Việc sử dụng mạng CNN mang lại kết quả tốt hơn đáng kể so với các mô hình học máy truyền thống. Tuy nhiên, R-CNN lại yêu cầu chi phí tính toán cao và mỗi vùng đề xuất lại được xử lý một cách riêng biệt, với tối đa có thể lên đến 2000 vùng như đã đề cập ở trên.

## Phương pháp Fast R-CNN

Để giải quyết vấn đề liên quan đến tốc độ xử lý, phương pháp Fast R-CNN được ra đời để khắc phục điểm yếu này của mạng R-CNN. Tương tự như R-CNN thì Fast R-CNN vẫn dùng *selective search* để lấy ra các khu vực đề xuất.

Tuy nhiên, Fast R-CNN không tách các vùng ra khỏi ảnh để đưa vào mô hình phân loại. Thay vào đó, Fast R-CNN cho cả bức ảnh vào một mạng CNN để lấy ra các bản đồ đặc trưng (feature map). Sau đó, các vùng đề xuất được lấy ra từ feature map tương ứng. Tiếp đó được Flatten và thêm 2 lớp kết nối đầy đủ (Fully connected layer) để dự đoán lớp của vùng đề xuất và giá trị offset values của bounding box.

Khi thực hiện Flatten để cho ra một vector có kích thước cố định, Region of Interest (ROI) pooling được ra đời. Khác so với max pooling hay average pooling thì kết quả đầu ra của ROI pooling luôn có kích thước cố định.



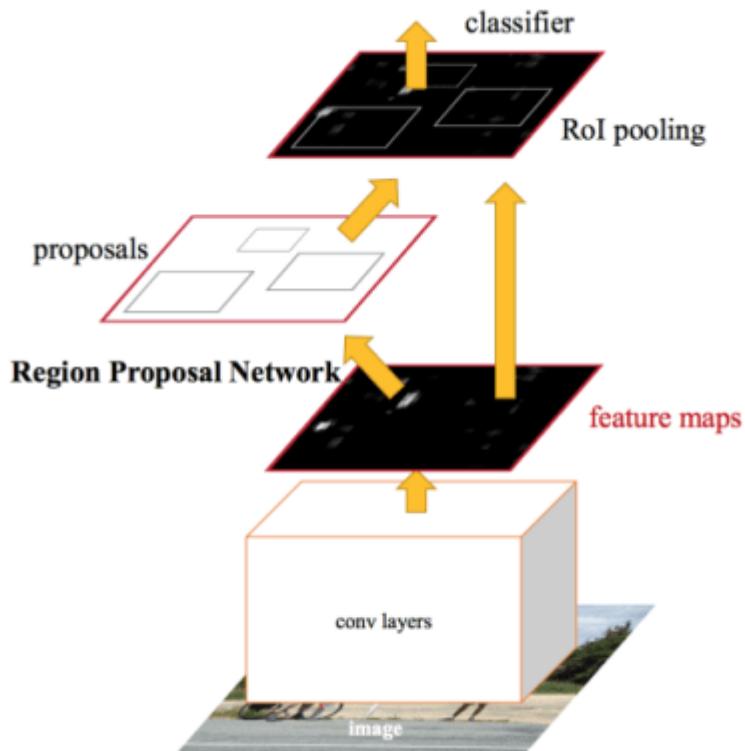
Hình 2.2 Minh họa cho phương pháp Fast R-CNN

(Nguồn: trích dẫn từ tài liệu tham khảo [2])

## Phương pháp Faster R-CNN

Chưa dừng lại ở đó, Faster R-CNN được ra đời nhằm tiếp tục cải tiến phương pháp Fast R-CNN với việc không sử dụng thuật toán selective search để lấy ra các vùng đề xuất. Thay vào đó, sử dụng một mạng CNN mới gọi là *Region Proposal Network* (RPN) để tìm các vùng đề xuất.

Việc đầu tiên cả bức ảnh được cho qua pre-trained model để lấy feature map. Sau đó feature map được dùng cho *Region Proposal Network* để lấy được các vùng đề xuất. Sau khi lấy được vị trí các vùng đề xuất thì thực hiện tương tự như Fast R-CNN.



Hình 2.3 Minh họa cho phương pháp Faster R-CNN

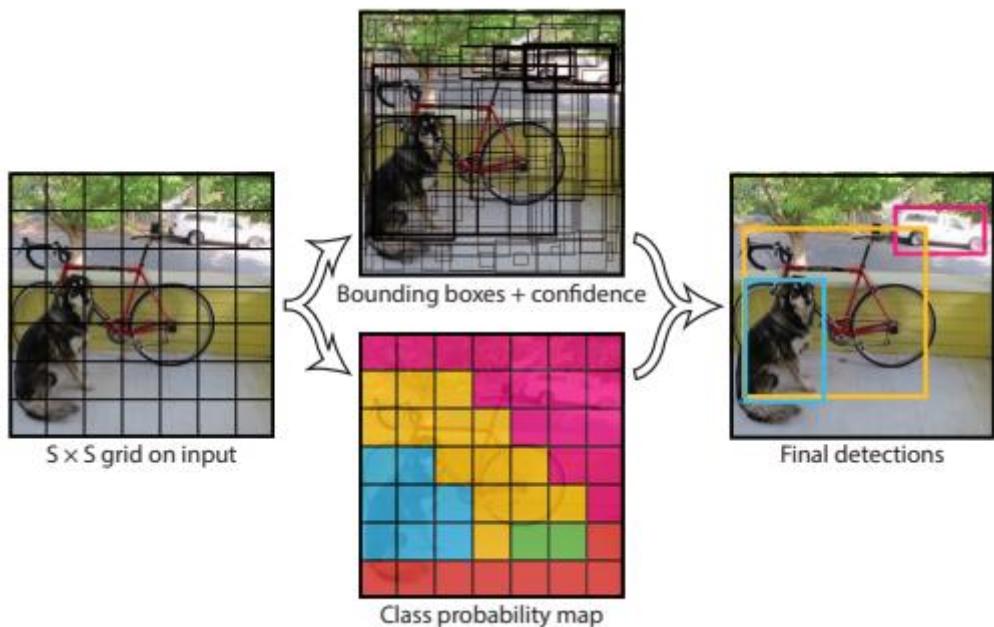
(Nguồn: trích dẫn từ tài liệu tham khảo [3])

#### 2.1.2. Phương pháp không dựa vào vùng đề xuất

Phương pháp không dựa vào vùng đề xuất hay còn gọi là phương pháp *một giai đoạn* (one-stage) hướng đến việc loại bỏ giai đoạn lấy ra các vùng đề xuất và trực tiếp huấn luyện một mô hình hình nhận diện duy nhất.

Với việc không sử dụng vùng đề xuất, phương pháp này có tiềm năng là nhanh hơn và đơn giản hơn, nhưng có độ chính xác thấp hơn so với phương pháp còn lại. Tiêu biểu cho phương pháp này có họ mô hình YOLO (You Only Look Once).

YOLO sử dụng một mô hình CNN truyền thăng để dự đoán trực tiếp đối tượng và vị trí của đối tượng trong ảnh. Dựa bài toán ban đầu về dạng bài toán hồi quy, các giá trị hồi quy là tọa độ, kích thước của đối tượng có trong ảnh. Dưới đây là mô tả ngắn gọn mô hình YOLOv1.



Hình 2.4 Minh họa cho phương pháp YOLO

(Nguồn: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Redmon\\_You\\_Only\\_Look\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf))

Với ý tưởng là chia bức ảnh ban đầu thành các ô dạng lưới nhỏ hơn có kích thước  $S \times S$  như ở Hình 2.4. Trong đó, kết quả mỗi ô đều cho ra  $B$  bounding box với các thông: *độ tin cậy* (confidence), *tọa độ*, *kích thước* của từng box và một vector thể hiện xác suất đối tượng trong box rồi vào một lớp cụ thể. Độ tin cậy thể hiện sự chắc chắn của mô hình cho khả năng mà box có chứa đối tượng. Vector đầu ra có dạng one-hot có độ dài là  $C$ , có một giá trị là 1 và các giá trị còn lại là 0. Như vậy, kết quả đầu ra cho toàn bộ bức ảnh sẽ bao gồm  $S \times S \times (B * 5 + C)$  giá trị.

Một vấn đề xuất hiện trong quá trình thực hiện đó là trong một ô có thể chứa nhiều bounding box của cùng đối tượng. Khi đó, một độ đo là IOU

(Intersection Over Unions) được sử dụng để thể hiện mức độ chồng của box được dự đoán so với box nhãn và được thể hiện bằng công thức:

$$IOU = \frac{\text{Phần diện tích giao}}{\text{Phần diện tích hợp}}$$



Hình 2.5 Ví dụ về độ đo IOU

(Nguồn: <https://www.datacamp.com/blog/yolo-object-detection-explained>)

Khi đó, thực hiện lấy bounding box có độ tin cậy cao nhất và bỏ các bounding box còn lại. Phương pháp này có tên là NMS (Non-Max Suppression). Cuối cùng, khi đã thực hiện tính toán trên từng ô, kết quả thu được sẽ được kết hợp lại để cho ra kết quả cuối cùng như ở Hình 2.4.

Hiện nay, các mô hình YOLO đã được cải tiến với nhiều phiên bản, mới nhất là YOLOv9. Các phiên bản mới mang đến cải thiện về độ hiệu quả của mô hình khi cùng số lượng tham số.

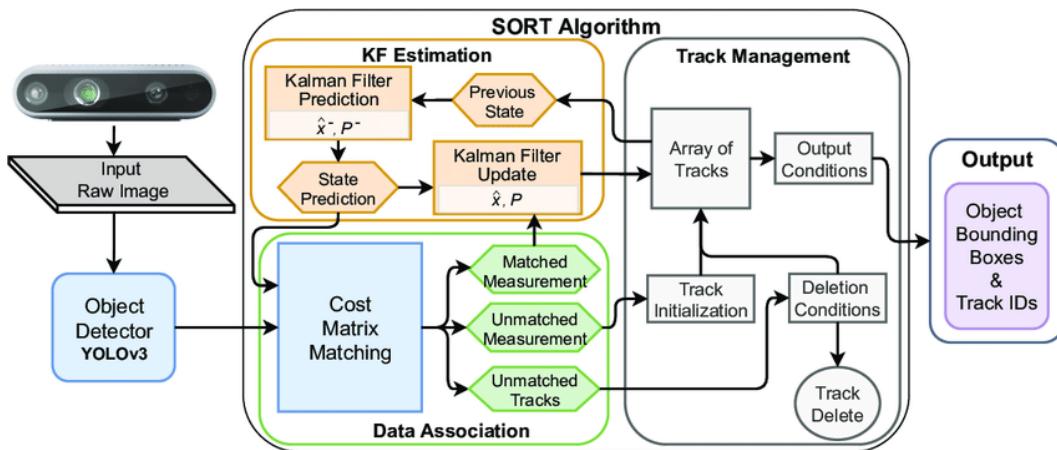
## 2.2. Kỹ thuật cho bài toán theo dõi đối tượng

Đối với bài toán theo dõi nhiều đối tượng (MOT), có hai hướng tiếp cận đó là theo dõi **Dựa vào nhận diện** (detection-based tracking) và **Theo dõi**

**không dựa vào nhận diện** (detection free tracking). Trong đó, DBT (Detection-Based Tracking) tập trung vào mối liên kết chặt chẽ giữa nhận diện đối tượng và theo dõi đối tượng, mỗi lần lặp xử lý theo dõi đối tượng đều có bước nhận diện đối tượng được thực hiện trước. Còn đối với DFB (Detection Free Tracking) chỉ yêu cầu bước nhận diện đối tượng thực hiện một lần. Trong đó, DBT đang trở thành hướng tiếp cận chính để giải quyết bài toán vì đem lại kết quả tốt hơn so với DFB, nhất là những cảnh có độ phức tạp cao. Hiện nay có rất nhiều phương pháp, kỹ thuật để giải quyết bài toán theo dõi đối tượng có thể kể đến như SORT, DeepSORT, ByteTrack, MDNet, CSTrack, v.v.

### 2.2.1. Phương pháp SORT

SORT là một phương pháp cho bài toán theo dõi đa đối tượng trong đó tập trung vào việc liên kết các đối tượng sẵn có với nhận diện một cách hiệu quả nhất. Phương pháp bao gồm các thành phần như sau:



Hình 2.6 Luồng hoạt động của phương pháp SORT

(Nguồn: [https://www.researchgate.net/figure/Overview-of-the-object-tracking-SORT-algorithm\\_fig2\\_358134782](https://www.researchgate.net/figure/Overview-of-the-object-tracking-SORT-algorithm_fig2_358134782))

- *Nhận diện* (Detection): Là phương pháp theo dõi dựa trên nhận diện nên việc nhận diện tốt sẽ mang lại kết quả tốt hơn [4]. Kết quả của nhận diện sẽ là các bounding box chứa đối tượng.
- *Mô hình dự đoán* (Estimation Model): Là một mô hình biểu diễn trạng thái của đối tượng để ước lượng trạng thái tiếp theo của đối tượng đã

được theo dõi. Trạng thái của đối tượng có thể được biểu diễn bằng một vector có dạng:

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$$

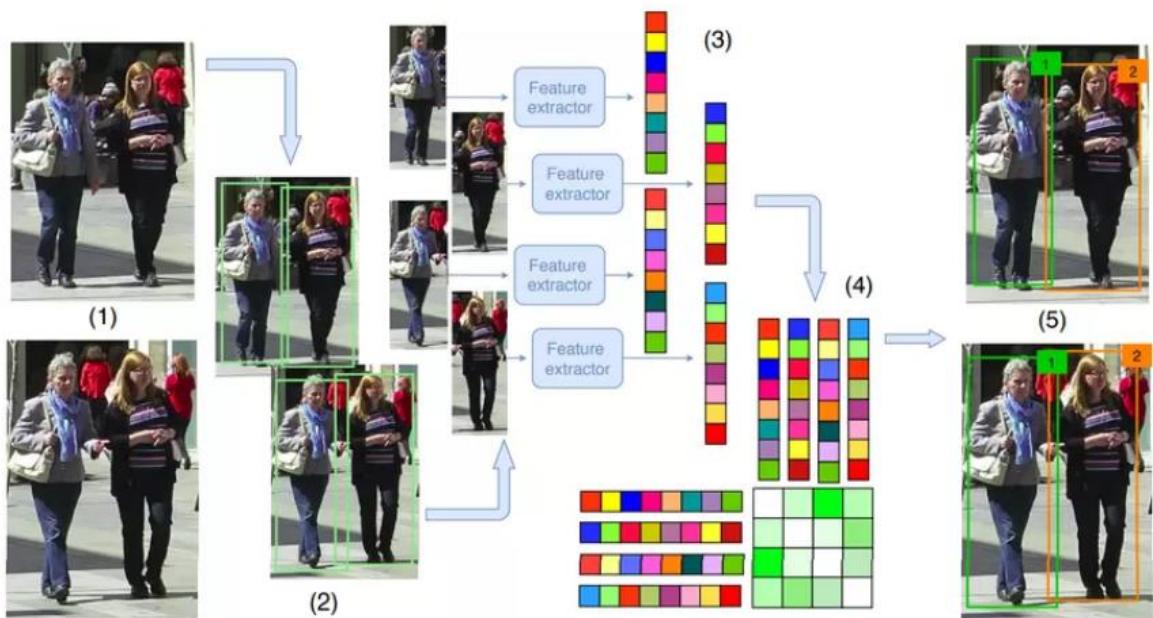
trong đó  $u$  và  $v$  biểu diễn tọa độ tâm của vật,  $s$  và  $r$  biểu diễn diện tích và tỉ lệ chiều dài và rộng của box chứa vật. Khi đối tượng được liên kết với nhận diện thì trạng thái của vật được cập nhật thông qua bộ lọc Kalman. Ngược lại, trạng thái tiếp theo sẽ được dự đoán dựa trên trạng thái vận tốc hiện tại [4], tức là ước lượng vị trí của vật trong khung hình tiếp theo.

- *Liên kết dữ liệu* (Data Association): Trong quá trình liên kết kết quả nhận diện cho một đối tượng có sẵn, vị trí bounding box tiếp theo của từng đối tượng sẽ được dự đoán. Khi đó, ma trận thể hiện lỗi liên kết (cost matrix matching) giữa một kết quả nhận diện và toàn bộ bounding box của các đối tượng có sẵn được tính toán dựa vào độ đo IOU. Sau đó, việc liên kết đối tượng sẽ sử dụng thuật toán Hungarian để liên kết sao cho tổng lỗi từ ma trận là nhỏ nhất. Cuối cùng, nếu giá trị IOU nhỏ hơn một ngưỡng  $IOU_{min}$  thì sẽ được loại bỏ [4]. Điều này nhằm loại bỏ những yếu tố gây nhiễu, ảnh hưởng đến kết quả.
- *Tạo và xóa Id của đối tượng* (Creation and Deletion of Track Identities): Khi các đối tượng vào và ra khỏi ảnh, Id cho đối tượng cần được tạo hoặc hủy tương ứng. Mỗi khi có một đối tượng mới được nhận diện, thông số trạng thái về vận tốc được đặt thành 0. Vì vận tốc không được quan sát tại thời điểm này nên phương sai của thành phần vận tốc được khởi tạo với các giá trị lớn, phản ánh sự không chắc chắn. [4]

SORT là một phương pháp đơn giản nhưng hiệu quả cao và là nền tảng để các phương pháp mới như DeepSORT, ByteTrack, BotSORT phát triển. Tính đến thời điểm hiện tại, các phương pháp có hướng tiếp cận như SORT vẫn được sử dụng phổ biến. Nhược điểm có thể thấy của phương pháp SORT đó chính là kết quả sẽ bị phụ thuộc vào chất lượng của việc nhận diện đối tượng.

### 2.2.2. Phương pháp DeepSORT

DeepSORT được phát triển ngay SORT nhằm giải quyết các vấn đề thiếu sót liên quan đến số lượng ID switches cao. Hướng giải quyết mà deepSORT đề xuất dựa trên việc sử dụng deep learning để trích xuất các đặc trưng của đối tượng nhằm tăng độ chính xác trong quá trình liên kết dữ liệu. Ngoài ra, một chiến lược liên kết cũng được xây dựng mang tên Matching Cascade giúp việc liên kết các đối tượng sau khi đã biến mất 1 thời gian được hiệu quả hơn.



Hình 2.7 Minh họa phương pháp DeepSORT

(Nguồn: <https://viblo.asia/p/sort-deep-sort-mot-goc-nhin-ve-object-tracking-phan-2-djeZ1m78ZWz>)

Trong deep SORT, nhóm tác giả giải quyết vấn đề data association dựa trên thuật toán Hungarian (tương tự như SORT), tuy nhiên, việc liên kết không chỉ dựa trên IOU mà còn quan tâm đến các yếu tố khác: Độ tương đồng giữa kết quả nhận diện (detection) và các đối tượng đang theo dõi.

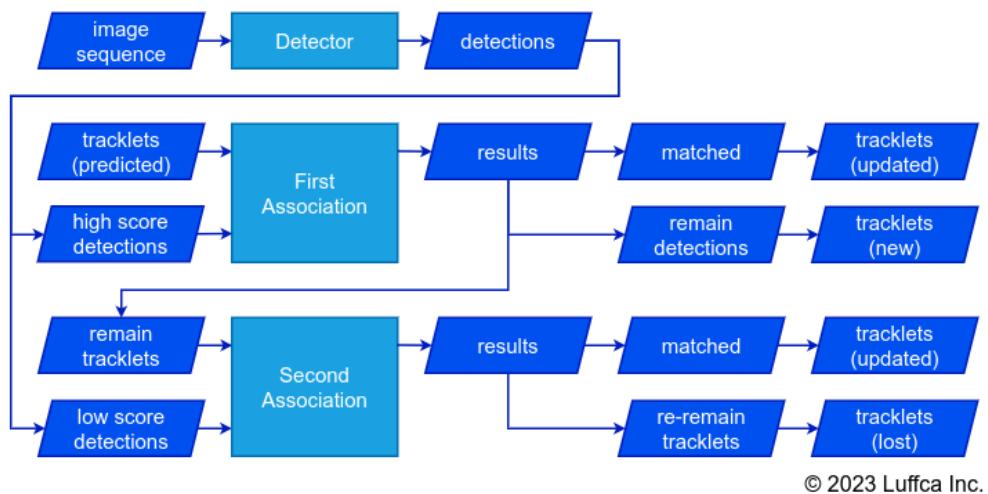
Để có các đặc trưng thu được từ các detection mang tính phân loại cao, Nicolai Wojke, Alex Bewley đã phát triển một kiến trúc mạng phần dư mở rộng (Wide Residual Network), huấn luyện riêng trên các bộ dữ liệu lớn về định danh

người (large scale re-id dataset) như: Market 1501, MARS, v.v. Tác vụ này còn được gọi là Cosine Metric Learning vì sử dụng hàm classifier mới là cosine softmax classifier. Chi tiết cụ thể có thể được tìm thấy ở bài báo gốc [5].

### 2.2.3. Phương pháp ByteTrack

Khác với các phương pháp SORT khác, chỉ giữ những box có độ tin cậy cao, ByteTrack sẽ giữ lại toàn bộ những box đã nhận diện được và chia thành 2 nhóm là thấp và cao dựa vào độ tin cậy. Dưới đây là một sơ đồ mô tả thuật toán BYTE nhằm thực hiện liên kết giữa nhận diện và đối tượng. Mã giả của thuật toán có thể được tìm thấy ở bài báo gốc [6].

ByteTrack: Multi-Object Tracking by Associating Every Detection Box

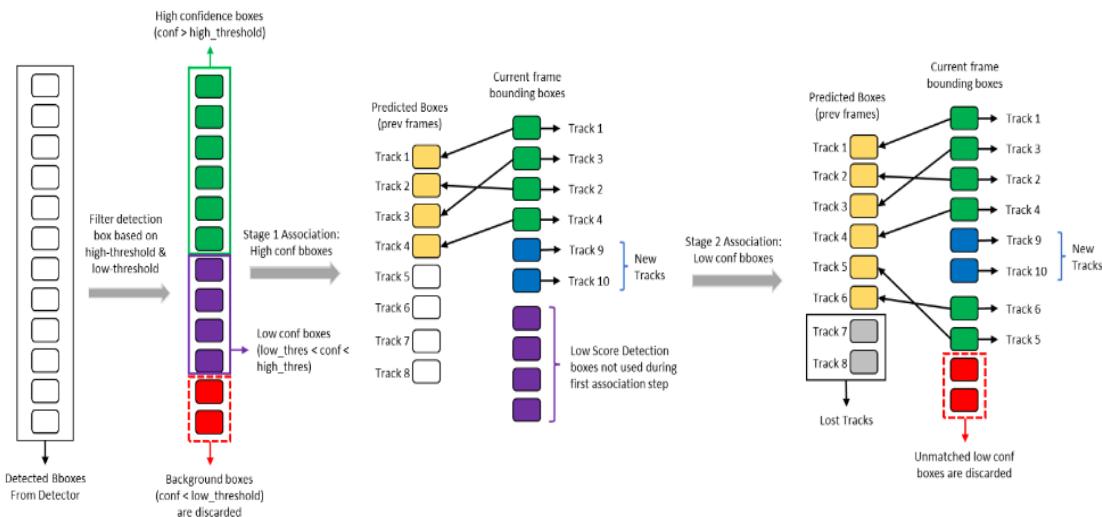


Hình 2.8 Sơ đồ luồng hoạt động của ByteTrack

(Nguồn: <https://www.luffca.com/2023/06/multiple-object-tracking-bytetrack/>)

Dựa vào sơ đồ trên, các kết quả nhận diện từ bộ nhận diện sẽ được phân chia thành 2 nhóm như đã đề cập. Có thể mô tả quá trình liên kết kết quả nhận diện với các đối tượng sẵn có, tương ứng với bước Data Association trong SORT, như sau:

- Khi đã có kết quả của 2 nhóm, kết quả từ nhóm có độ tin cậy cao (high score detections) sẽ được thực hiện việc liên kết với các đối tượng đã sẵn có trong những khung hình trước (tracklet).
- Tiếp theo đó, các kết tracklet chưa được liên kết sẽ được thực hiện với nhóm có độ tin cậy thấp hơn (low score detections).
- Nếu vẫn còn high score detections thì thực hiện tạo ra các tracklet mới.
- Nếu có tracklet chưa được liên kết thì cho tracklet đó vào tập tracklet thất lạc (tracklet lost), tức quản lý các đối tượng không còn xuất hiện trong khung hình. Các tracklet trong tập thất lạc sẽ được loại bỏ nếu sau hơn 30 khung hình không có sự liên kết với tracklet đó.
- Cuối cùng, cập nhật các tracklet theo các detection đã liên kết, tức cập nhật trạng thái của các đối tượng theo bộ lọc Kalman.



Hình 2.9 Minh họa phương pháp ByteTrack

(Nguồn: <https://www.datature.io/blog/introduction-to-bytetrack-multi-object-tracking-by-associating-every-detection-box.>)

Trong bài báo gốc [6], các tác giả có thực hiện việc thử độ đo khoảng cách giữa tracklet và detection với các độ đo IOU và Re-ID. Với Re-ID là một độ đo cosine sử dụng các đặc trưng của đối tượng. Hơn thế, với 2 giai đoạn liên kết dữ liệu, các tác giả đã thử từng độ độ ứng từng giai đoạn và kết quả như sau:

Bảng 2-1 Kết quả ByteTracking trích từ tài liệu tham khảo [6]

Độ đo giao đoạn 1	Độ đo giao đoạn 2	Bộ dữ liệu MOT17			Bộ dữ liệu BDD100K		
		MOTA	IDF1	IDs	mMOTA	mIDF1	IDs
IoU	Re-ID	75.8	77.5	231	39.2	48.3	29172
IoU	IoU	<b>76.6</b>	79.3	<b>159</b>	39.4	48.9	27902
Re-ID	Re-ID	75.2	78.7	276	45.0	53.4	10425
Re-ID	IoU	76.3	<b>80.5</b>	216	<b>45.5</b>	<b>54.8</b>	<b>9140</b>

Từ Bảng 2-1, tác giả đưa ra khẳng định IoU và Re-ID đều phù hợp cho giai đoạn 1. Còn đối với giai đoạn 2 thì ưu tiên sử dụng độ đo IoU, lý do cho điều này là nhóm có độ tin cậy thấp thường sẽ chứa nhiều như ảnh mờ, vật bị che lấp. Do đó, sử dụng Re-ID, hay đặc trưng của vật sẽ không phù hợp.

Với những thử nghiệm khác ở bài báo gốc, ByteTrack được khẳng định là một thuật toán tuy đơn giản nhưng hiệu quả, cân bằng giữa tốc độ và độ chính xác.Thêm vào đó, phương pháp Byte có thể được tích hợp vào mọi thuật toán theo dõi khác. Một nhược điểm cũng tương như SORT đó là kết quả của ByteTrack cũng phụ thuộc vào chất lượng của việc nhận diện đối tượng.

### 2.3. Kỹ thuật cho bài toán tái nhận diện

Đã có nhiều kỹ thuật được sử dụng để thực hiện bài toán tái nhận diện, Re-identification (Re-ID) trong thị giác máy tính. Các phương pháp học máy cơ bản có thể mang lại kết quả tốt tùy thuộc vào hoàn cảnh môi trường nhưng kết quả sẽ không chính xác nếu có sự thay đổi của hậu cảnh, ánh sáng, v.v. Do đó, các phương pháp học sâu đang trở thành chủ đề nghiên cứu cho bài toán tái nhận diện. Thêm vào đó, các nghiên cứu về Re-ID từ trước đến nay chủ yếu tập trung

vào tái nhận diện người và tái nhận diện phương tiện giao thông, với kết quả cao nhất là từ kỹ thuật mạng nơ-ron tích chập.

Trích chọn đặc trưng và so sánh giữa hai hình ảnh khác nhau thông qua độ đo khoảng cách đã bắt đầu trở thành phương pháp chính cho Re-ID. Với hình ảnh ban đầu được đưa vào một mạng học sâu có kết quả đầu ra là một vector biểu diễn đặc trưng của đối tượng có trong ảnh (embedding vector). Sau đó, các độ đo khoảng cách như Euclidean, Triplet Loss, k-reciprocal hoặc độ đo Cosine để so sánh sự tương đồng giữa hai vector đặc trưng. Khoảng cách giữa hai đặc trưng càng thấp (hoặc nhỏ hơn) thì các đặc trưng này càng có nhiều khả năng thuộc cùng một đối tượng [7].

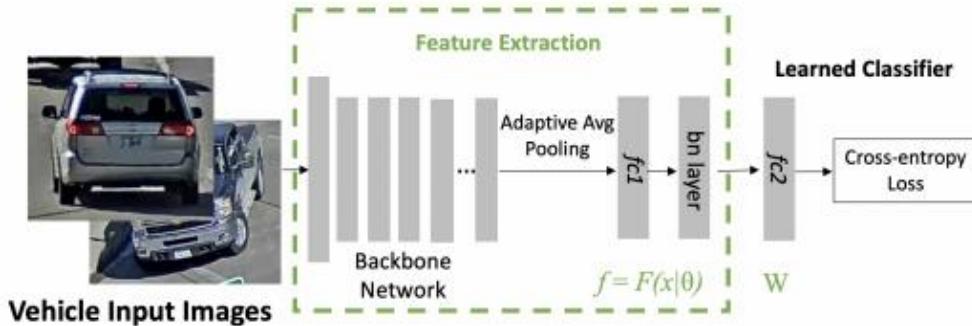
Tính đến thời điểm hiện tại, các bài toán liên quan đến tái nhận diện chỉ được nghiên cứu phổ biến cho bài toán tái nhận diện người và phương tiện giao thông. Mà cách tiếp cận của hai bài toán này lại không có nhiều điểm chung. Cụ thể, hình ảnh về một người nào đó vẫn có những điểm chung nếu nhìn từ nhiều góc độ ví dụ như màu tóc, họa tiết quần áo, v.v. Còn những hình ảnh về một phương tiện giao thông từ nhiều góc độ có thể không có nhiều điểm chung, khó để có thể thực hiện việc tái nhận diện. Do đó, các phương pháp dưới đây có thể sẽ chỉ giới hạn cho bài toán tái nhận diện phương tiện.

### 2.3.1. Mô hình VehicleNet

VehicleNet là một bộ dữ liệu quy mô lớn được thiết kế để nâng cao hệ thống nhận dạng lại phương tiện (re-id). Bộ dữ liệu này tích hợp nhiều bộ dữ liệu về phương tiện công cộng như CityFlow, VeRi-776, CompCar và VehicleID. Việc kết hợp này để giải quyết thách thức về các biến động về hình dạng của phương tiện trong nhiều góc camera khác nhau. Tuy nhiên, bộ dữ liệu CityFlow sẽ là bộ dữ liệu trọng tâm trong quá trình huấn luyện. Ngoài ra, các tác giả trong bài báo gốc [8] gọi mô hình huấn luyện trên bộ dữ liệu này là VehicleNet.

Dưới đây là mô ngắn gọn cấu trúc của mô hình VehicleNet:

- **Bộ trích chọn đặc trưng** (Feature Extractor): Việc sử dụng mô hình đã được huấn luyện (pre-trained model) giúp tiết kiệm thời gian huấn luyện mô hình và có thể đạt được kết quả tốt hơn so với việc huấn luyện một mô hình từ đầu. Do đó, các mô hình pre-trained trên bộ dữ liệu ImageNet như ResNet-50, DenseNet-121, v.v được sử dụng làm bộ trích chọn đặc trưng. Các mô hình pre-trained sẽ được loại bỏ các lớp cuối, thay và thêm các lớp mới để phù hợp cho quá trình huấn luyện. Các lớp mới được thêm là lớp Adaptive Average Pooling, Fully Connected, Batch Normalization được thể hiện ở Hình 2.10. Quan trọng nhất là lớp fc2 trong hình có vai trò phân lớp ảnh đầu vào để phục vụ huấn luyện mô hình.



*Hình 2.10 Minh họa mô hình VehicleNet*

*(Nguồn: trích từ tài liệu tham khảo [8])*

- **Nhúng đặc trưng** (feature embedding): Đây là phần chính để thực hiện bài toán Re-ID nói chung, với cách tiếp cận là so sánh vector đặc trưng của các đối tượng với nhau. Đối với mô hình VehicleNet, phần cho ra một vector mô tả đặc trưng của đối tượng chính là đầu ra của lớp BN sau lớp fc1 được thể hiện ở Hình 2.10.

Quá trình huấn luyện của mô hình Vehicle sẽ bao gồm 2 giai đoạn. Đầu tiên là huấn luyện mô hình theo bài toán phân loại đa lớp trên toàn bộ dữ liệu. Với số lớp là số phương tiện riêng biệt có trong toàn bộ bộ dữ liệu. Tiếp đó, thay thế lớp fc2 để huấn luyện riêng biệt trên bộ CityFlow. Lý do cho điều này là việc huấn luyện trên toàn bộ dữ liệu sẽ giúp mô hình tìm ra các đặc trưng mang

tính phân biệt giữa các phương tiện giao thông [8] nhằm mang lại kết quả phân lớp tốt hơn.

### 2.3.2. Mô hình Resnet IBN

IBN viết tắt của Instance Batch Normalization được ra đời nhằm tới mục đích xử lý hiện tượng khi một mô hình đạt kết quả cao trong một miền (domain) nhưng lại đạt kết quả thấp ở miền khác. Ví dụ cho miền ở đây đó là hình ảnh một phương tiện từ thế giới thực và hình ảnh của phương tiện ở trong trò chơi điện tử hoặc môi trường giả lập.



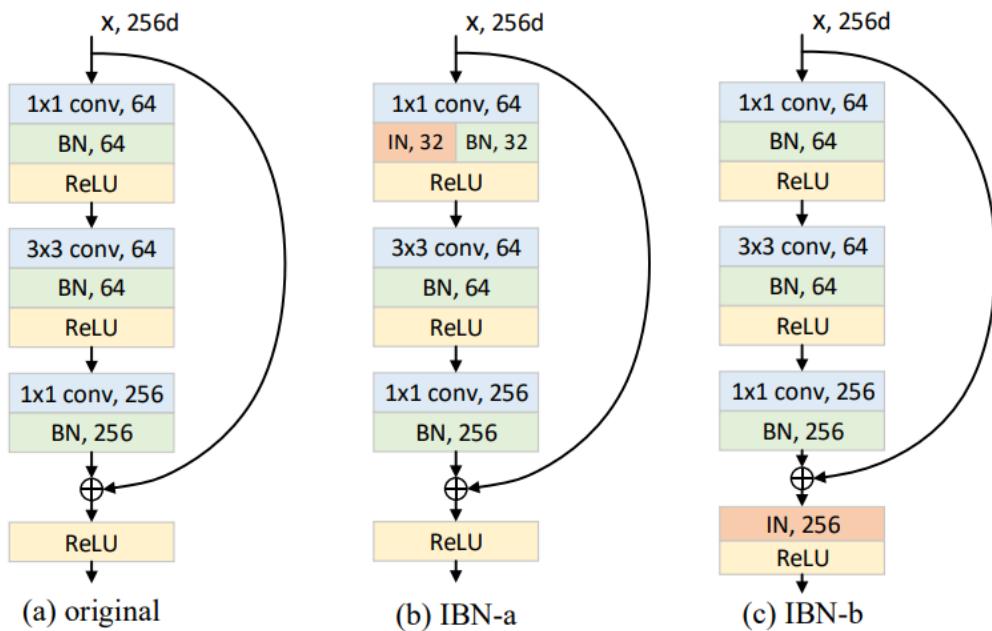
*Hình 2.11 Ví dụ về dữ liệu miền thực và miền ảo*

*(Nguồn: trích dẫn từ tài liệu tham khảo [9])*

Instance Batch Normalization chính là các phép chuẩn hóa Batch và Instance. Chuẩn hóa batch là thực hiện chuẩn hóa trên một batch của quá trình huấn luyện. Cụ thể là chuẩn hóa các vector đầu vào trong các lớp ẩn (hidden layers) sử dụng các đại lượng là kỳ vọng và phương sai. Hiệu quả của quá trình chuẩn hóa batch là làm cho quá trình huấn luyện trở nên ngắn hơn và ổn định hơn [10]. Trong khi đó chuẩn hóa Instance liên quan đến việc chuẩn hóa một đối

tương đầu vào cụ thể. Tác dụng của việc chuẩn hóa Instance là việc chuẩn hóa độ tương phản trong dữ liệu đầu vào [11]. Hơn thế, IN (Instance Normalization) làm cho mô hình có sự bất biến về hình ảnh và ngoại hình, tức bất biến đối với các miền khác nhau. Còn BN (Batch Normalization) có khả giữ được các đặc trưng mang tính phân biệt [12] trong quá trình truyền trong mạng sâu.

Kết hợp lại sẽ được các khối IBN minh họa như hình dưới:

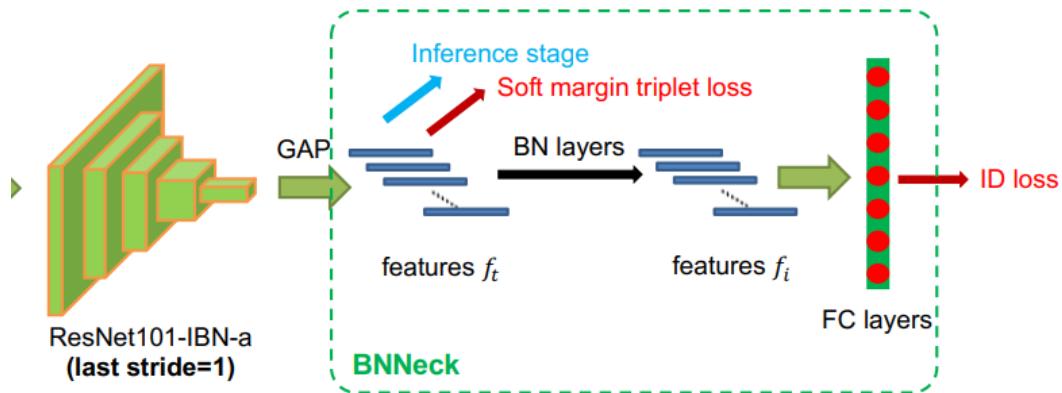


Hình 2.12 Sơ đồ các khối IBN

(Nguồn: trích từ tài liệu tham khảo [12])

Có thể thấy, khái IBN không khác quá nhiều so với một khái residual của mạng Resnet thông thường. Lớp IN được thêm vào sau lớp tích chập 1x1 đầu tiên hoặc ngay trước hàm kích hoạt. Sự khác biệt giữa khái IBN-a và IBN-b được đề cập ở [12], đó là khái IBN-a hướng đến việc lưu giữ các thông tin ở các lớp trong mạng còn khái IBN-b hướng đến việc giảm sự phân tán của đặc trưng, tức thể hiện tốt hơn ở các domain mới.

Các khái IBN trên có thể được tích hợp vào các mô hình mạng nơ-ron tích chập phổ biến như các mô hình ResNet, VGG, v.v. Dưới đây là mô hình ResNet101-IBN-a trọng tâm của phần này:



Hình ảnh 2.1 Kiến trúc mô hình ResNet101-IBN-a

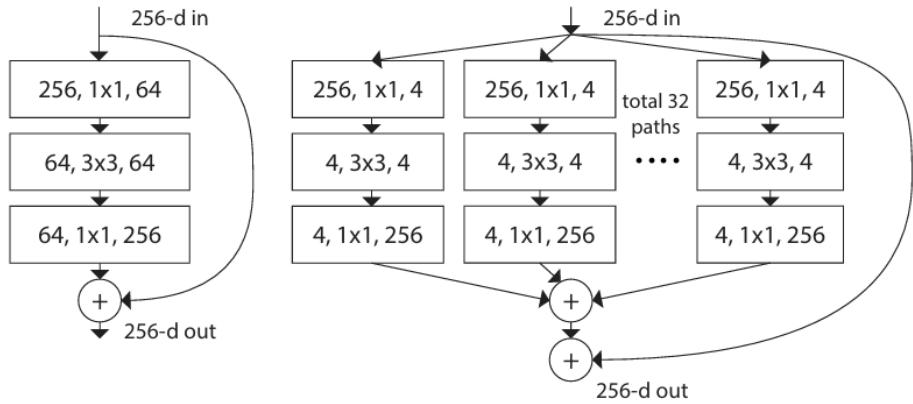
(Nguồn: trích từ tài liệu tham khảo [13])

Một số điểm nổi bật của cấu trúc mô hình đó là last stride = 1, Soft margin triplet loss. Với last stride là bước trượt của lớp tích chập cuối cùng của mô hình. Việc chọn last stride = 1 mang đến kết quả là các đặc trưng thu được có kích thước lớn hơn, chứa nhiều thông tin hơn nên kết quả mang lại sẽ tốt hơn [14]. Còn đối với Soft margin triplet loss là phiên bản lề mềm của triplet loss, là phiên bản cải tiến so với Triplet loss ban đầu.

Điều đặc biệt của mô hình này là nằm ở quá trình huấn luyện với tên gọi là huấn luyện đa miền (multi-domain learning). Dữ liệu huấn luyện được lấy từ bộ ảnh thực và bộ ảnh từ môi trường giả lập với một số lượng nhất định được đề cập ở [13]. Bộ dữ liệu sẽ được tăng cường bằng cách xóa một vùng ngẫu nhiên trong ảnh. Quá trình huấn luyện có hai giai đoạn là huấn luyện và fine-tune. Việc huấn luyện sẽ được thực hiện trên bộ dữ liệu đã đề cập ở trên, quá trình fine-tune chỉ huấn luyện mô hình trên bộ dữ liệu thực.

### 2.3.3. Mô hình ResNeXt

Mô hình ResNeXt là một mô hình được giới thiệu vào năm 2016 trong bài báo “Aggregated Residual Transformations for Deep Neural Networks”. Đây là mô hình cải tiến so với mô hình ResNet thông thường với sự khác biệt nằm ở các khối trong mô hình.



*Hình 2.13 So sánh giữa hai khối của hai mô hình ResNet và ResNext*

(*Nguồn: trích từ tài liệu tham khảo [15]*)

ResNeXt giới thiệu "cardinality block", đây là một phần mới được thêm vào trong kiến trúc mạng ResNeXt. Cardinality block có nhiệm vụ tạo ra sự phân chia (split) của các kênh đầu vào thành nhiều "nhóm cardinality" (cardinality groups) như ở hình Hình 2.13. Mỗi nhóm cardinality đại diện cho một tập hợp các đặc trưng (features) cụ thể mà mô hình sẽ học. Với số nhóm cardinality trong một khối sẽ được ký hiệu là C.

Các khối trong mô hình có cấu trúc tương đồng nhau và được tuân thủ theo hai quy tắc giống như mô hình VGG/ResNet:

- (i) Nếu kết quả đầu ra của các khối có kích thước là như nhau thì các khối sẽ có cùng các siêu tham số (chiều rộng và kích thước filter).
- (ii) Mỗi khi kích thước đầu ra của khối có kích thước giảm đi một nửa, chiều rộng của khối tiếp theo sẽ có kích thước gấp 2 lần.

Với các quy tắc trên, mô hình mạng ResNeXt sẽ được xác định một cách dễ dàng hơn vì các khối sử dụng chung một cấu trúc, chỉ khác nhau ở độ rộng của từng khối. Ví dụ cụ thể hiện cho cấu trúc ResNeXt-50 (32x4d) được thể hiện ở hình dưới:

stage	output	ResNet-50	<b>ResNeXt-50 (32x4d)</b>
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C=32 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C=32 \\ 1\times1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C=32 \\ 1\times1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C=32 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		<b>25.5×10<sup>6</sup></b>	<b>25.0×10<sup>6</sup></b>
FLOPs		<b>4.1×10<sup>9</sup></b>	<b>4.2×10<sup>9</sup></b>

Hình 2.14 So sánh giữa ResNet-50 và ResNeXt-50 (32x4d)

(Nguồn: trích từ tài liệu tham khảo [15])

Trong Hình 2.14, cấu trúc của mạng ResNeXt-50 (32x4d) có siêu tham số  $C=32$ , 4d thể hiện chiều rộng của các nhóm cardinality; khói trong các ô có các tham số có dạng như ‘1x1, 128’ thể hiện lớp tích chập có kích thước 1x1 và số kênh đầu ra là 128; số ngay bên cạnh các khói thể hiện số lần lặp lại của khói.

Là sự phát triển so với mô hình ResNet thông thường, mô hình ResNeXt mang lại kết quả tốt hơn mà không yêu cầu thêm về chi phí tính toán. Các thử nghiệm ở bài báo [15] đã chỉ ra điều đó.

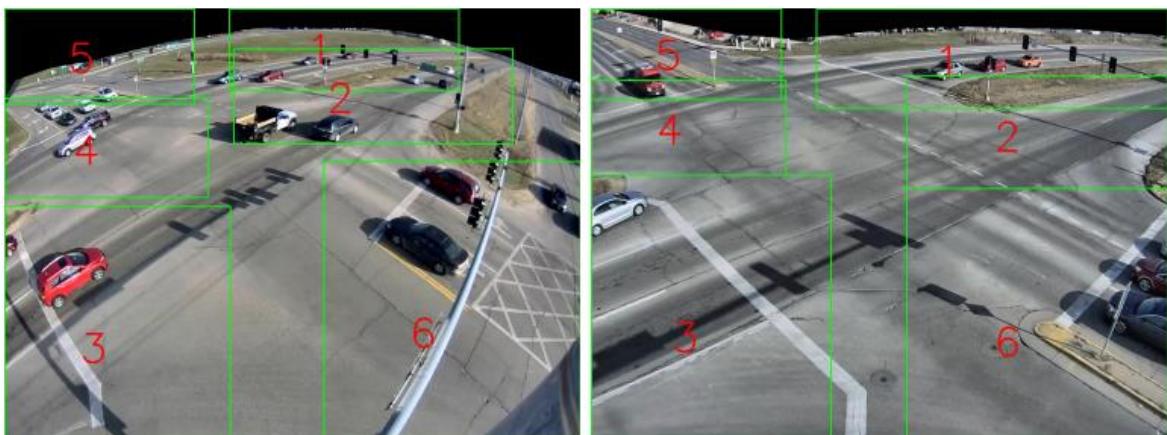
Việc áp dụng mô hình ResNeXt cho bài toán tái nhận diện đối tượng trong giới hạn báo cáo này được thực hiện tương tự như mô hình VehicleNet đã nêu ở trên. Cụ thể hơn là loại bỏ các lớp classifier, thay bằng các lớp fc1, BN và fc2 như ở Hình 2.10.

## 2.4. Kỹ thuật cho bài toán liên kết đối tượng giữa các camera

Dưới đây là một phương pháp cụ thể được sử dụng trong cuộc thi AI City Challenge 2022 bởi nhóm 28 [16] có tên là **Box-grained Reranking Matching**. Khác với các kỹ thuật khác là sử dụng các trung bình đặc trưng từ phương tiện (tracklet), kỹ thuật này sử dụng đặc trưng của toàn bộ box trong tracklet để thực hiện liên kết giữa các camera. Và việc liên kết được thực hiện lần lượt trên cặp camera liền kề mà có phương tiện có thể di chuyển qua lại. Cặp camera được xác định thành 2 vùng là *vùng vào* và *vùng ra*.

### 2.4.1. Tìm các phương tiện ứng cử

Đầu tiên, các khung cảnh trong camera sẽ được đánh dấu các vùng một cách thủ công, ứng với mỗi vùng là các đoạn đường mà phương tiện di chuyển có thể di chuyển đến. Dưới đây là minh họa cụ thể cho camera thứ 42 và 43:



*Hình 2.15 Vùng được định nghĩa trước trong camera thứ 42 và 43*

(*Nguồn: trích từ tài liệu tham khảo [16]*)

Với các vùng được đánh dấu ở trên, khi thực hiện liên kết giữa hai camera, việc đầu tiên là lấy ra các tracklet có vùng ra và vùng vào tương ứng. Ví dụ trong Hình 2.15, vùng 1 của camera 42 liên kết với vùng 4 của camera 43 nên sẽ thực hiện lấy các tracklet có điểm kết thúc ở vùng 4 đối với camera 43 và tracklet có điểm bắt đầu ở vùng 1 của camera 42. Việc tìm ra các tracklet như

trên giúp giảm thời gian tìm kiếm cũng như giảm việc liên kết sai giữa các phương tiện (tracklet).

Thêm vào đó, cặp camera sẽ có thông tin về thời gian khoảng di chuyển hợp lệ của phương tiện giao thông. Xét một cặp tracklet thuộc vùng ra và vùng vào, nếu thời gian di chuyển của cặp tracklet giữa hai camera không thuộc trong khoảng hợp lệ thì sẽ được loại bỏ, không xét liên kết. Như vậy sẽ cần đến thông tin về thời gian xuất hiện và biến mất của một phương tiện (tracklet) trong một camera.

Từ đó, các thông tin của một phương tiện (tracklet) về vùng vào, vùng ra, thời gian xuất hiện, thời gian biến mất được ký hiệu như sau:

$$Traj_i = \{[Z_{in}, Z_{out}], [time_{in}, time_{out}]\}$$

#### 2.4.2. Tính toán ma trận khoảng cách $D$ giữa đặc trưng của các box

Khi đã có các tracklet được lấy từ hai camera với  $Z_{out} = [T_1, \dots, T_n]$  và  $Z_{in} = [\bar{T}_1, \dots, \bar{T}_m]$  ký hiệu cho tập tracklet từ vùng ra và vùng vào. Trong đó,  $T_i = [B_i^1, \dots, B_i^{h_i}]$  và  $\bar{T}_j = [\bar{B}_j^1, \dots, \bar{B}_j^{h_j}]$  là các tracklet từ vùng ra và vùng vào.  $B_i^h$  là đặc trưng của box thứ  $h$  của tracklet thứ  $i$ . Ma trận khoảng cách giữa các box của vùng vào và vùng ra được ký hiệu như sau với cos là độ đo cosine:

$$S = \begin{bmatrix} \cos(B_1^1, \bar{B}_1^1) & \dots & \cos(B_1^1, \bar{B}_1^{h_m}) \\ \dots & \dots & \dots \\ \cos(B_1^{h_n}, \bar{B}_1^1) & \dots & \cos(B_1^{h_n}, \bar{B}_1^{h_m}) \end{bmatrix}_{\sum_{i=1}^n h_i \times \sum_{j=1}^m h_j}$$

Từ ma trận  $S$  trên, ma trận khoảng cách  $D$  sẽ được tính theo phương pháp **Re-ranking with k-reciprocal Encoding** [17]. Với Re-ranking có thể hiểu đơn giản là bước tìm kiếm với 2 giai đoạn, giai đoạn 1 là tìm ra những phần tử liên quan, giai đoạn 2 là sử dụng những phần tử đã tìm được để thực hiện tìm kiếm tiếp. Còn k-reciprocal (k-láng giềng tương hỗ) là thuật toán tìm ra  $k$  phần tử gần nhất với một phần tử đang xét (giống KNN). Ví dụ cho ý tưởng là phần tử  $a$  gần

với phần tử  $b$  khi cả  $a$  và  $b$  nằm trong danh sách k phần tử gần nhất của nhau theo thuật toán tìm kiếm nào đó như KNN.

Đầu tiên, tạo ma trận  $S' = \begin{bmatrix} S_{out} & S \\ S^T & S_{in} \end{bmatrix}$  thể hiện khoảng cách từ một box đến toàn bộ các box có trong vùng ra và vùng vào với  $S_{out}, S_{in}$  được tính giống như  $S$  nhưng chỉ thực hiện trên vùng ra/vùng vào.

Ma trận  $S'$  còn được biến đổi từ độ đo Cosine sang độ đo Euclid theo các công thức  $S' = (2 - 2 * S')^2$  và thực hiện chuẩn hóa sao cho các cột có giá trị lớn nhất là 1.

Ứng với mỗi box  $B_i^{h_t}$  trong thuộc vùng ra, tính toán danh sách ranking đầu tiên theo KNN trên ma trận  $S'$  đã được biến đổi:

$$N(B_i^{h_t}, k_1) = \{b_1^0, b_2^0, b_3^0 \dots, b_{k_1}^0\}, |N(B_i^{h_t}, k_1)| = k_1$$

Với  $b_i^0$  là đặc trưng của box có thể là từ vùng ra hoặc vùng vào. Nếu  $b_i^0$  chỉ là đặc trưng box thuộc vùng ra thì có thể dẫn đến hiện tượng liên kết sai trong trường hợp tracklet  $i$  đang xét không có trong vùng vào.

Tiếp đó, tính toán ra danh sách  $R(B_i^{h_t}, k_1)$  theo K-reciprocal nearest neighbors theo định nghĩa:

$$R(B_i^{h_t}, k_1) = \{(b_i \in N(B_i^{h_t}, k_1)) \cap (B_i^{h_t} \in N(b_i, k_1))\}$$

Phần tử trong  $R(B_i^{h_t}, k_1)$  hiểu đơn giản là các  $b_i$  có  $N(b_i, k_1)$  chứa  $B_i^{h_t}$ .

Tiếp tục, tính ra danh sách  $R^*(B_i^{h_t}, k_1)$  thỏa mãn điều kiện:

$$R^*(B_i^{h_t}, k_1) \leftarrow R(B_i^{h_t}, k_1) \cup R\left(q, \frac{1}{2}k_1\right)$$

$$s.t. \quad \left| R(B_i^{h_t}, k_1) \cap R\left(q, \frac{1}{2}k_1\right) \right| \geq \frac{2}{3} \left| R\left(q, \frac{1}{2}k_1\right) \right|$$

$$\forall q \in R(B_i^{h_t}, k_1)$$

Với  $R^*(B_i^{ht}, k_1)$  có thể hiểu là danh sách được mở rộng từ các phần tử có trong  $R(B_i^{ht}, k_1)$  sao cho các phần tử thêm vào có mức độ độc liên quan nhất định đối với các phần tử đã có sẵn trong  $R(B_i^{ht}, k_1)$ .

Sau đó, độ đo khoảng cách Jaccard thể hiện độ tương đồng giữa  $B_i^{ht}$  và  $b_i$  được tính bằng công thức:

$$d_J(B_i^{ht}, b_i) = 1 - \frac{|R^*(B_i^{ht}, k_1) \cap R^*(b_i, k_1)|}{|R^*(B_i^{ht}, k_1) \cup R^*(b_i, k_1)|}$$

Cuối cùng khoảng cách  $d^*$  giữa  $B_i^{ht}$  và  $b_i$  được tính theo công thức:

$$d^*(B_i^{ht}, b_i) = (1 - \lambda)d_J(B_i^{ht}, b_i) + \lambda d(B_i^{ht}, b_i)$$

Với  $\lambda$  là siêu tham số,  $d(B_i^{ht}, b_i)$  là khoảng cách giữa  $B_i^{ht}$  và  $b_i$  được tính sẵn ở ma trận  $S'$ .

Thực hiện tính toán như trên với toàn bộ  $B_i^{ht}$  thì sẽ thu được ma trận D.

Và ma trận D sẽ tiếp tục được xử lý liên quan thời gian di chuyển của tracklet giữa các camera:

$$D_{i,j} = \begin{cases} e^{\frac{\alpha_t \times (t_{low} - t_{i,j})}{\beta_t}} \times D_{i,j}, & t_{i,j} < t_{low} \\ e^{\frac{\alpha_t \times (t_{i,j} - t_{upp})}{\beta_t}} \times D_{i,j}, & t_{i,j} > t_{upp} \end{cases} \quad (1)$$

Với  $\alpha_t, \beta_t$  là các siêu tham số,  $t_{low}, t_{upp}$  là ngưỡng dưới và trên của thời gian di chuyển giữa hai camera và  $t_{i,j}$  là thời gian di chuyển của cặp tracklet giữa hai camera.

Cuối cùng, ma trận D được xử lý với tỷ lệ che khuất của box:

$$D_{i,j} = \begin{cases} e^{\alpha_t \times (1+r_o)} \times D_{i,j}, & r_o > r_{thre} \\ D_{i,j} & \text{otherwise} \end{cases} \quad (2)$$

Với  $r_o$  là tỷ lệ che khuất của box, được tính dựa trên IOU với các box khác trong một khung hình,  $r_{thre}$  là ngưỡng tỷ lệ che khuất.

Việc tính với (1) có ý nghĩa khá rõ ràng, nếu cặp tracklet có thời gian di chuyển nằm ngoài khoảng xác định thì đánh trọng số cho cặp tracklet đó trong ma trận  $D$ . Và với (2) cũng tương tự, khi một box bị che khuất nhiều thì cũng sẽ bị đánh trọng số trong ma trận  $D$ .

#### 2.4.3. Liên kết phương tiện sử dụng k-láng giềng tương hõ

Khi đã có ma trận khoảng cách  $D$ , một chiến lược dựa trên [17] được sử dụng đó là tất cả các tracklet được liên kết dựa trên nguyên lý của **k-láng giềng tương hõ** (k-reciprocal nearest neighbors).

Đầu tiên, xét một box  $B_i^h$ , tức box thứ  $h$  của tracklet thứ  $T_i$  thuộc vùng ra, có tập  $N(B_i^h, k)$  là tập chứa  $k$  box gần nhất, tức k-láng giềng gần nhất (KNN), so với  $B_i^h$ :

$$N(B_i^h, k) = (\bar{B}_1, \bar{B}_2, \dots, \bar{B}_k), |N(B_i^h, k)| = k.$$

Từ đó tìm ra:

$$R(B_i^h, k) = \{(\bar{B}_j \in N(B_i^h, k)) \cap (B_i^h \in N(\bar{B}_j, k_1))\}$$

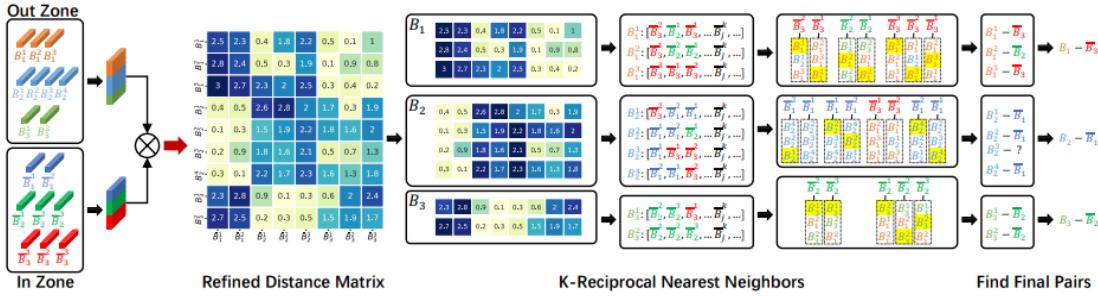
Tức là lọc các  $\bar{B}_j$  thuộc  $N(B_i^h, k)$  sao cho có  $B_i^h$  nằm trong k phần tử gần nhất so với  $\bar{B}_j$ .

Trong  $R(B_i^h, k)$ , mỗi  $\bar{B}_j$  sẽ thuộc một tracklet nào đó và tìm tracklet trong  $R(B_i^h, k)$  sao cho tracklet đó xuất hiện nhiều lần nhất. Bước trên được gọi là phép gán box, ký hiệu là:

$$M(B_i^h, \bar{T}_j) = MaxCount\{N(B_i^h, B_j) \cap N(\bar{B}_j, B_i^h)\}$$

Khi tất cả các box  $B_i^h$  của tracklet  $T_i$  đã tìm được tracklet  $\bar{T}_h$  gần nhất, tracklet  $T_i$  sẽ được gán cho tracklet  $\bar{T}_h$  khi  $\bar{T}_h$  xuất hiện nhiều nhất trong tập các tracklet gần nhất với  $B_i^h$ .

Dưới đây là sơ đồ tổng kết lại quá trình tính toán ma trận khoảng cách  $D$ , thực hiện liên kết tracklet giữa cặp camera dựa trên k-láng giềng tương hỗ.



Hình 2.16 Mô tả trực quan quá trình liên kết đối tượng giữa các camera

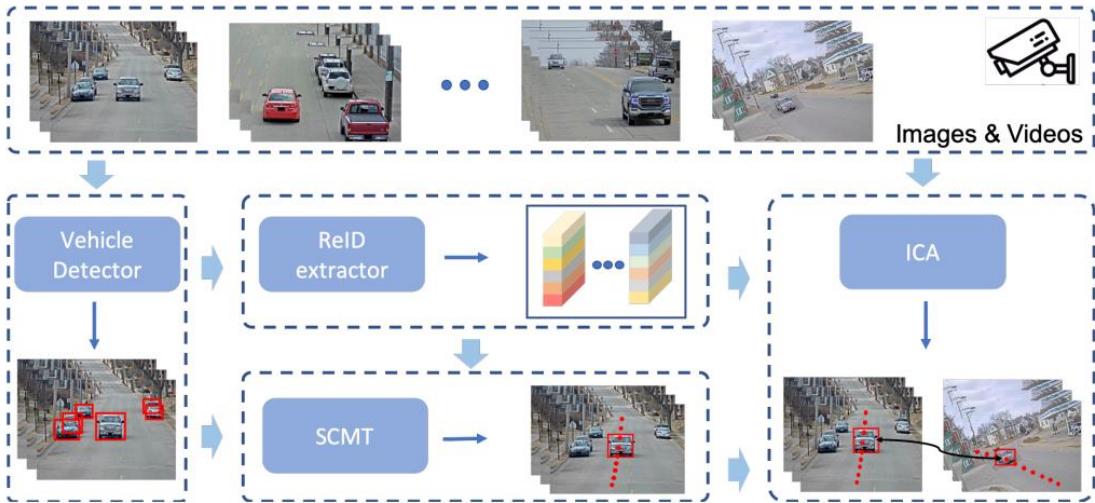
(Nguồn: trích từ tài liệu tham khảo [16])

Ưu điểm của phương pháp sử dụng tất cả các box để thực hiện việc liên kết là sẽ giữ được các đặc trưng riêng biệt của các phương tiện giao thông. Nếu sử dụng phương pháp lấy trung bình các đặc trưng của phương tiện giao thông thì các đặc trưng riêng biệt sẽ không còn, làm giảm khả năng liên kết.

Tuy nhiên, nhược điểm của phương pháp này sẽ yêu cầu tính toán nhiều hơn do sử dụng toàn bộ đặc trưng từ phương tiện giao thông. Từ đó, làm giảm khả năng mở rộng của phương pháp khi có nhiều camera.

## 2.5. Giải pháp đề xuất cho bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera

Có thể nhận thấy, bài toán theo dõi phương tiện giao thông từ nhiều camera là một bài toán lớn, để giải quyết cần chia nhỏ thành các bài toán con để giải quyết như đã đề cập. Các bài toán con đóng vai trò như những mắt xích không thể thiếu trong quá trình giải quyết bài toán ban đầu.Thêm vào đó, các bài toán con có thể được coi là những mô-đun trong hệ thống để giải quyết bài toán ban đầu. Dưới đây là sơ đồ tổng quát cho quá trình giải quyết bài toán theo dõi phương tiện giao thông từ nhiều camera:



*Hình 2.17 Sơ đồ hệ thống cho bài toán*

(*Nguồn: trích từ tài liệu tham khảo [16]*)

Hoạt động của hệ thống bắt đầu với dữ liệu đầu vào các video từ các camera khác nhau. Dữ liệu được đi qua mô-đun nhận diện phương tiện trong từng khung hình của video, kết quả là các bounding box chứa phương tiện trong từng khung hình. Các bounding box được sử dụng để lấy ra hình ảnh cụ của các phương tiện rồi đưa qua mô-đun trích chọn đặc trưng ReID. Sau đó, các đặc trưng và bounding box được đưa đến quá trình xử lý theo dõi trong một camera, kết quả là các tracklet địa phương. Tracklet chứa thông tin về các đặc trưng, bounding box về một phương tiện trong một camera theo thứ tự về thời gian. Khi đã có được kết quả của từng camera, bước cuối cùng chính là việc liên kết tracklet trong các camera lại với nhau tạo thành một tracklet toàn cục.

Trong giới hạn báo cáo này, các chi tiết cụ thể về các mô-đun đó là:

- *Mô-đun nhận diện đối tượng:* Sử dụng mô hình YOLOv8.
- *Mô-đun trích xuất đặc trưng Re-ID:* Sử dụng ba mô hình ResNet101-IBN-a, Resnet50, ResNext-50 (32x4d).
- *Mô-đun theo dõi đối tượng từ một camera:* Sử dụng thuật toán ByteTrack.
- *Mô-đun liên kết giữa các camera:* Sử dụng kỹ thuật Box-grained Reranking Matching.

## CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM

### 3.1. Kết quả huấn luyện mô hình ResNext-50 (32x4d)

#### 3.1.1. Dữ liệu huấn luyện

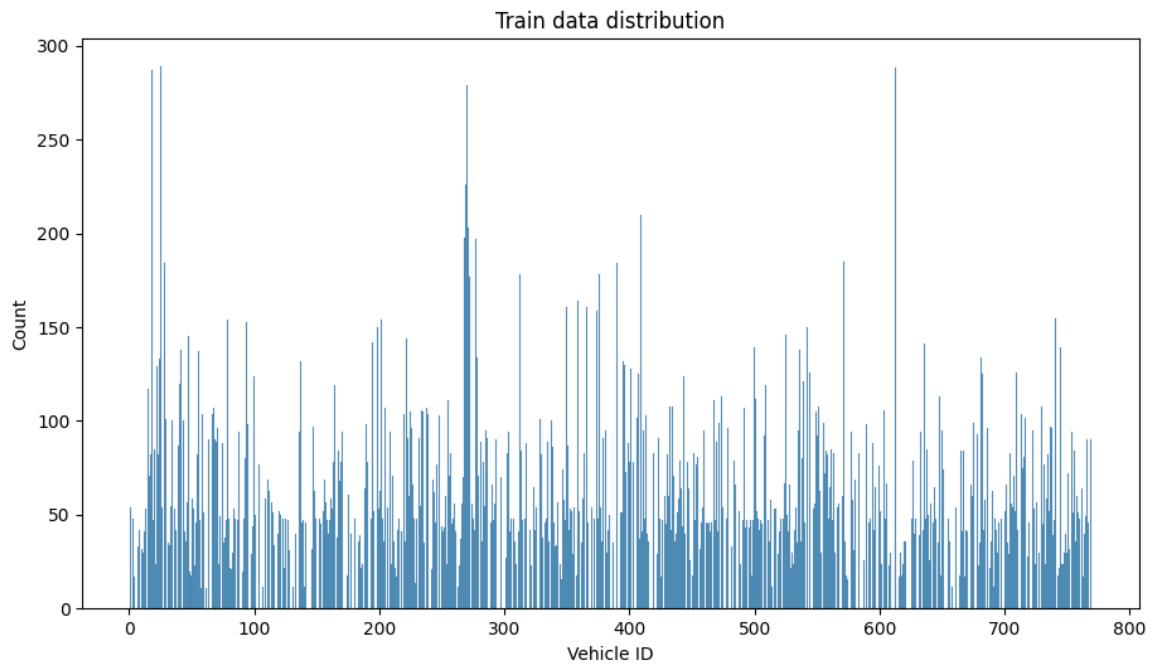
Bộ dữ liệu được sử dụng dùng để huấn luyện mô hình ResNext-50 (32x4d) là bộ dữ liệu VeRi-776 [18]. Bộ dữ liệu chứa hơn 50.000 ảnh của 776 phương tiện giao thông riêng biệt được ghi hình bởi 20 camera.



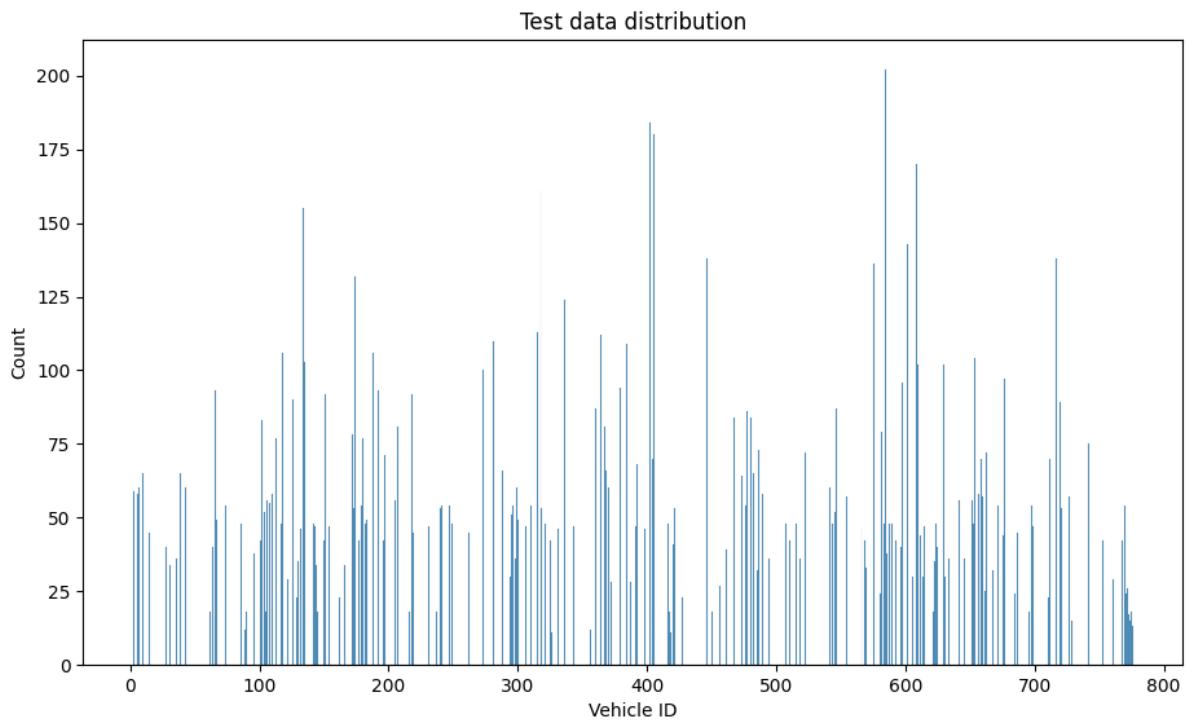
*Hình 3.1 Minh họa cho bộ dữ liệu VeRi-776*

(*Nguồn: <https://github.com/JDAI-CV/VeRidataset>*)

Ngoài ra, bộ dữ liệu được chia sẵn thành 2 tập train và test. Tập train chứa 37.746 ảnh, tập test chứa 12.254 ảnh. Cụ thể hơn, sự phân bố của các phương tiện giao thông trong các tập được thể hiện ở hình dưới:



*Hình 3.2 Biểu đồ thể hiện sự phân bố của các phương tiện trong tập train của bộ dữ liệu VeRi-776*



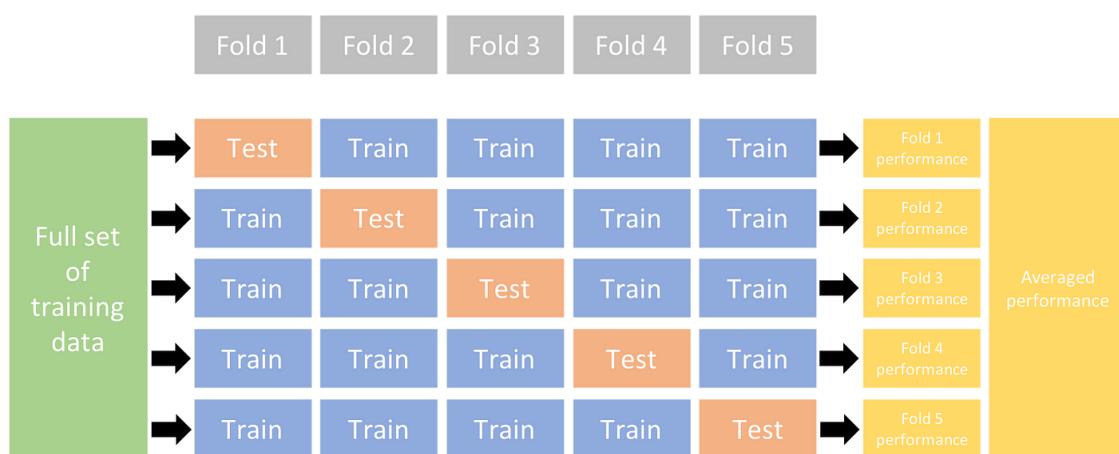
*Hình 3.3 Biểu đồ thể hiện sự phân bố của các phương tiện trong tập test của bộ dữ liệu VeRi-776*

Dựa vào hai hình trên, có thể thấy dữ liệu được phân bố khá đồng đều, chỉ có một lượng nhỏ phương tiện có số hình ảnh nhiều hơn so với các phương tiện

còn lại trong bộ dữ liệu. Nhìn chung, bộ dữ liệu trên có thể được sử dụng luôn để huấn luyện, không nhất thiết phải thực hiện tiền xử lý.

### 3.1.2. Kết quả huấn luyện mô hình

Quá trình huấn luyện mô hình sẽ sử dụng quy trình K-fold để đánh giá kết quả huấn luyện của mô hình. Quy trình K-fold chia bộ dữ liệu huấn luyện thành k phần như nhau. Quá trình thực hiện K-fold được lặp lại k lần, mỗi lần lấy lần lượt một phần trong k phần là bộ dữ liệu đánh giá, các phần còn lại dùng để huấn luyện mô hình.

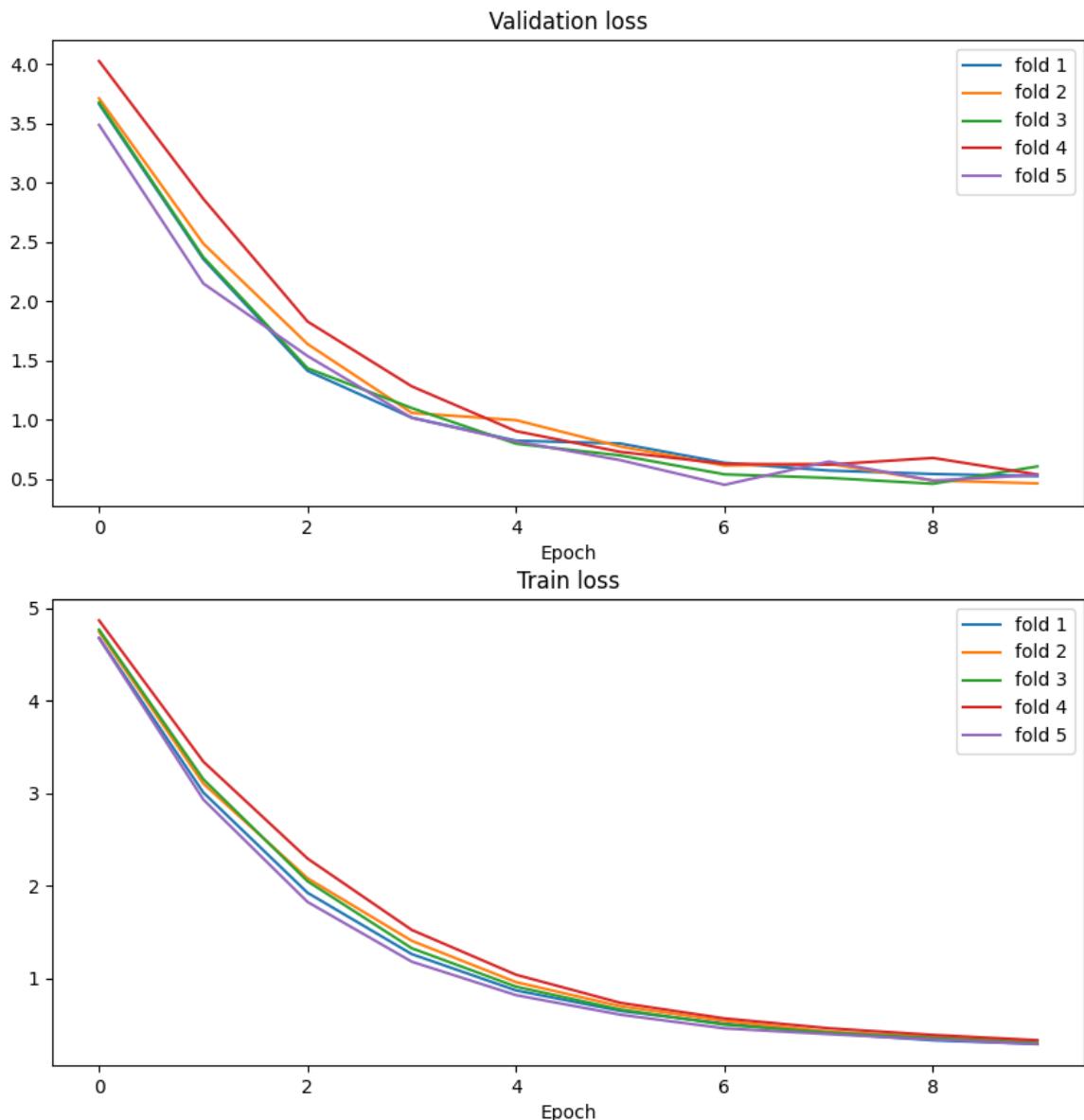


Hình 3.4 Minh họa quy trình kiểm định K-fold

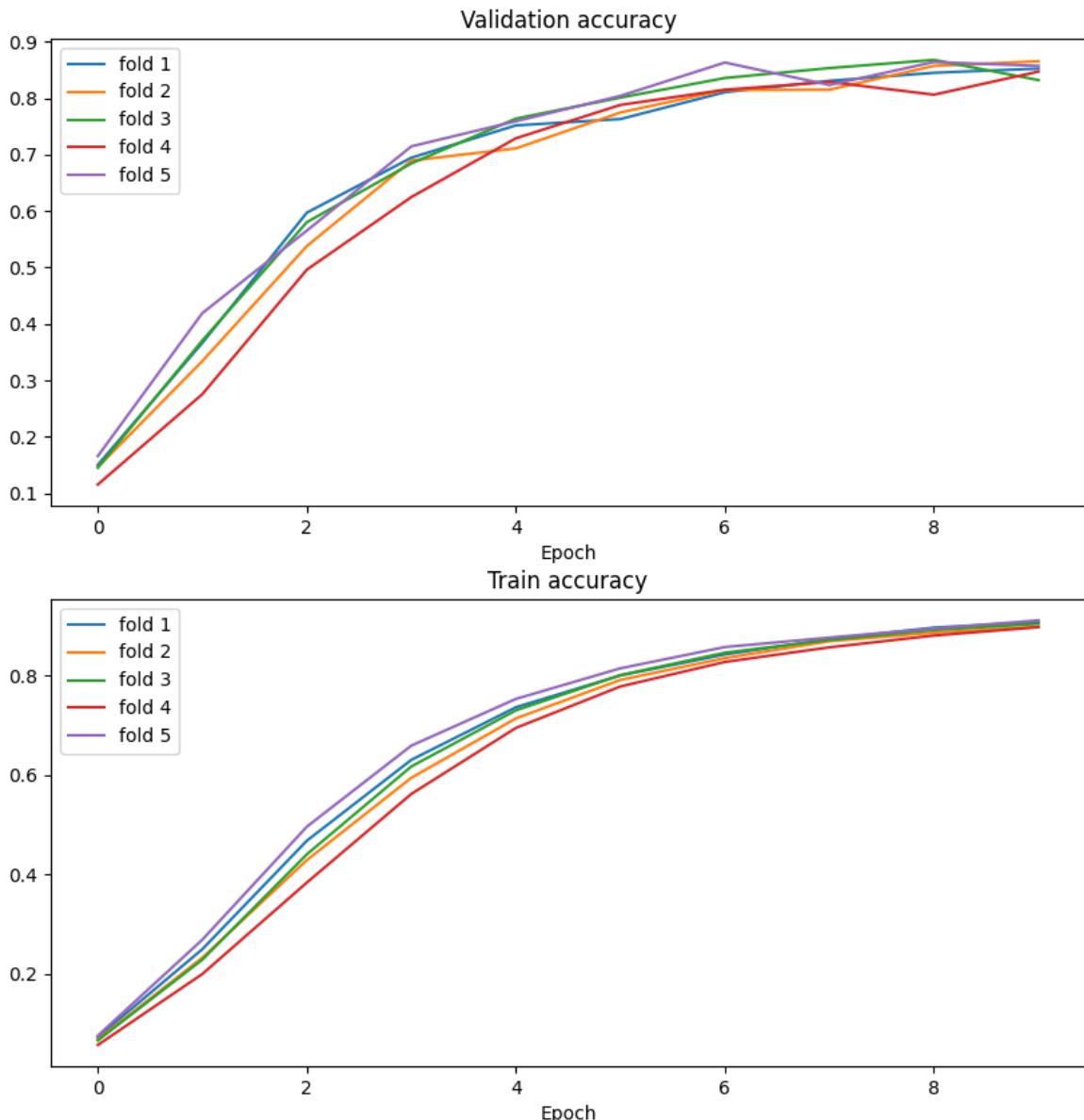
(Nguồn: <https://machinelearningmastery.com/k-fold-cross-validation/>)

Thực hiện quy trình K-fold để đánh giá mô hình sẽ cho ra kết quả khách quan hơn, đánh giá hiện tượng overfitting của mô hình.

Kết quả huấn luyện mô hình sẽ được trình bày được đánh giá theo các độ đo là loss, accuracy cho cả quá trình train và validate với giá trị các tham số epoch = 10, batch\_size = 32 và kfold = 5.



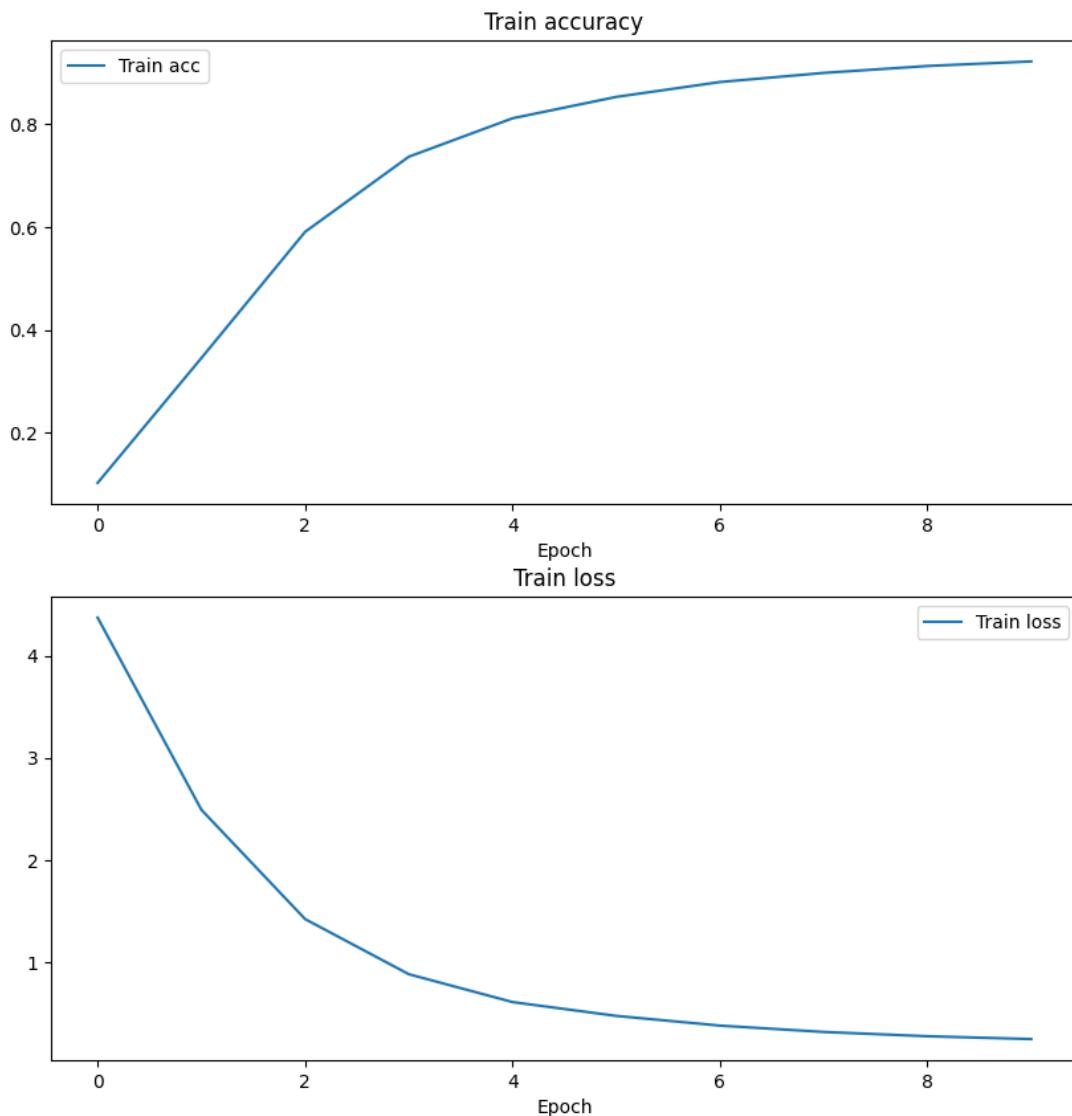
Hình 3.5 Kết quả loss trong quá trình kfold mô hình ResNext-50 (32x4d)



Hình 3.6 Kết quả accuracy trong quá trình kfold mô hình ResNext-50 (32x4d)

Dựa vào Hình 3.5 và Hình 3.6, có thể nhận thấy mô hình chưa có hiện tượng overfitting. Giá trị của các đại lượng loss và accuracy trên tập validate đồng biến theo giá trị loss và accuracy trên tập train. Như vậy, có thể thực hiện huấn luyện mô hình với epochs nhiều hơn để đạt được kết quả tốt hơn.

Dưới đây là kết quả huấn luyện trên toàn bộ tập huấn luyện của bộ dữ liệu:



Hình 3.7 Kết quả accuracy và loss trong quá trình huấn luyện mô hình ResNext-50 (32x4d)

Kết quả huấn luyện của mô hình trên sẽ được sử dụng để đánh giá cho phương pháp đã nêu.

### 3.2. Kết quả thực nghiệm trên bộ dữ liệu AI City Challenge

#### 3.2.1. Giới thiệu bộ dữ liệu

Bộ dữ liệu được sử dụng trong bản báo cáo này là bộ dữ liệu thuộc vòng 1 của cuộc thi AI City Challenge 2022 [1]. Bộ dữ liệu có video của 45 camera trong đó 40 dữ liệu của 40 camera đã có sẵn nhãn, 5 video còn lại được sử dụng để đánh giá kết quả của cuộc thi. Độ dài của các video không quá 5 phút với số khung hình trên giây là 10fps, riêng camera 15 là 8fps và thời điểm ghi hình của

các camera chênh lệch nhau không quá 10 giây. Ngoài ra, các video được chia sẵn vào các tập train, validate và test với số lượng video khác nhau.

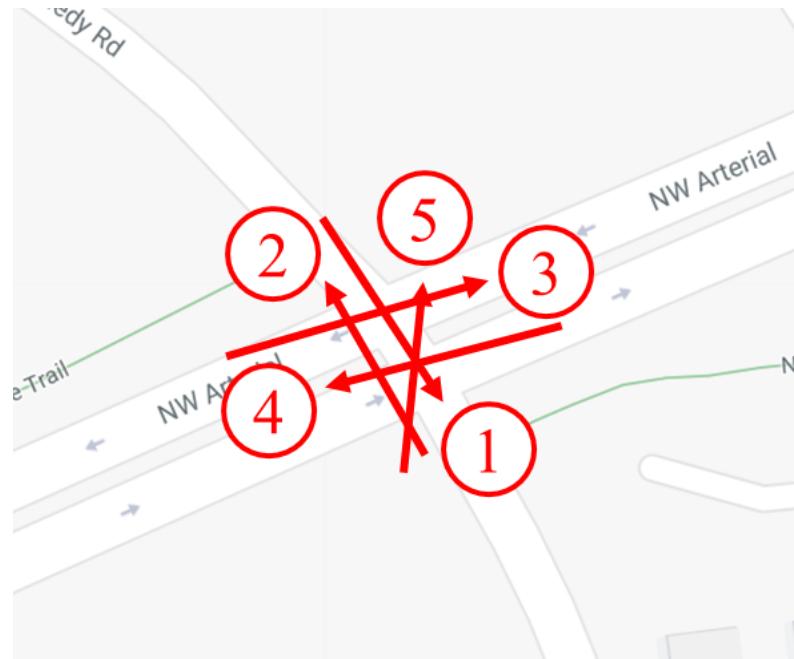
Dưới đây là bảng mô tả về bộ dữ liệu AI City Challenge:

*Bảng 3-1 Mô tả tập dữ liệu con trong bộ dữ liệu AI City Challenge*

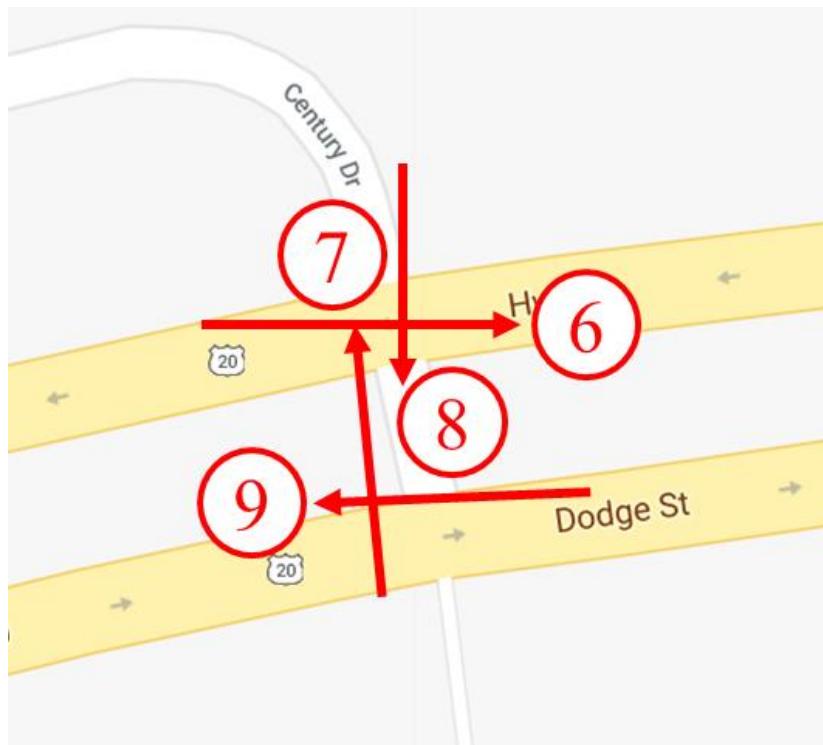
Tập dữ liệu con	Số video	Camera ID
S01	5	1->5
S02	4	6->9
S03	6	10->15
S04	25	16->40
S05	27	10->36
S06	6	41->46

Với Camera ID thể hiện các video trong tập con thuộc các camera có ID tương ứng, ví dụ như: Tập S01 sẽ có 5 video, 5 video này thuộc các camera có ID từ 1 cho đến 5. Trong các tập con, S01, S03, S04 thuộc tập train, S02, S05 thuộc tập validate và tập test chứa duy nhất tập S04.

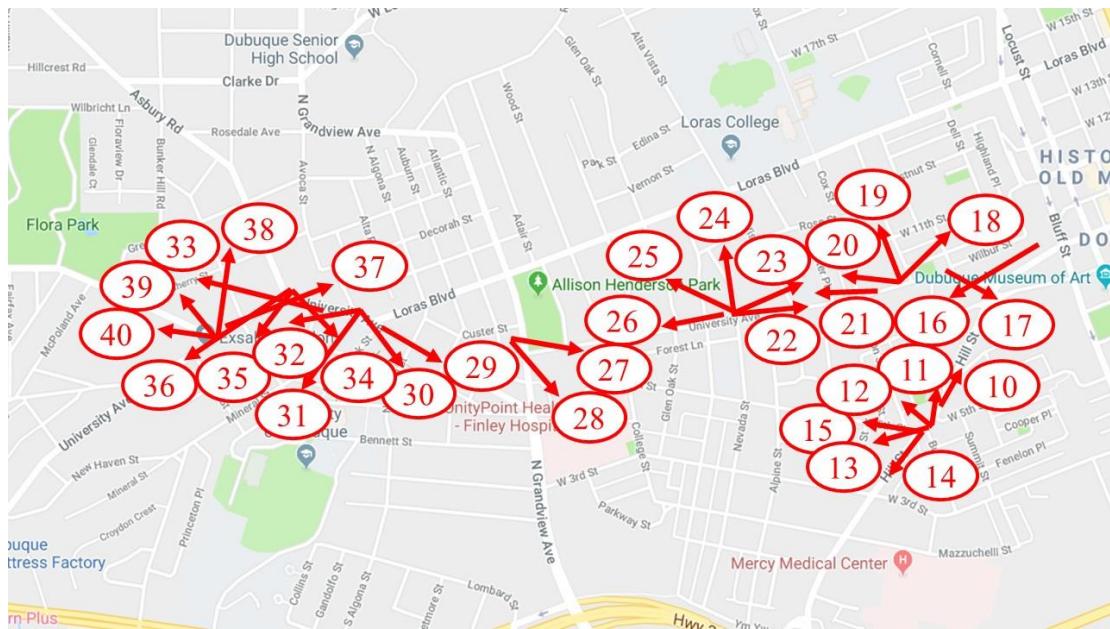
Các video được sử dụng để đánh giá thực nghiệm thuộc tập S03, có các camera có ID là 10, 11, 12, 13, 14, 15. Lý do cho điều này là tập S03 chứa các video có góc nhìn camera ít bị chồng lên nhau và có góc nhìn từ một phương tiện đa dạng hơn. Điều này giúp cho việc đánh giá giải pháp được khách quan hơn. Dưới đây là hình ảnh thể hiện sự bố trí của các camera thuộc các tập khác nhau ở trên bản đồ:



Hình 3.8 Sơ đồ bố trí các camera trên bản đồ của tập S01



Hình 3.9 Sơ đồ bố trí các camera trên bản đồ của tập S02



Hình 3.10 Sự bố trí các camera trên bản đồ của các tập S03, S04 và S05



Hình 3.11 Sự bố trí các camera trên bản đồ của tập S06

Trong các hình, hướng mũi tên thể hiện hướng nhìn của các camera trên bản đồ với gốc là địa điểm đặt camera. Còn số gần với đầu mũi tên thể hiện ID của camera. Dưới đây là hình ảnh cụ thể từ các camera thuộc tập S03 được sử dụng để đánh giá phương pháp:



*Hình 3.12 Cảnh của các camera thuộc tập S03*

### 3.2.2. Kết quả thực nghiệm

Kết quả theo dõi từ camera riêng lẻ được thực hiện trên các mô hình được huấn luyện sẵn có, bao gồm các mô hình YOLOv8, ResNet101-IBN-a, ResNet50 (VehicleNet). Với mô hình YOLOv8 được phát triển bởi công ty Ultralytics. Mô hình ResNet101-IBN-a được huấn luyện bởi các tác giả trong bài báo “City-Scale Multi-Camera Vehicle Tracking Guided by Crossroad Zones” [19]. Lý do cho điều này là do các mô hình có tính phức tạp cao, dữ liệu huấn luyện lớn nên việc sử dụng các mô hình đã được huấn luyện nhằm tiết kiệm thời gian.

Bước đầu tiên để theo dõi phương tiện giao thông từ nhiều camera là theo dõi phương tiện giao thông từ các camera một cách riêng lẻ như đã trình bày ở trên. Tuy nhiên, do bộ dữ liệu được sử dụng không có nhãn cho việc theo dõi cho camera riêng lẻ nên không thể thực hiện đánh giá kết quả trên một camera. Dưới đây là kết quả trực quan khi thực hiện việc theo dõi trên một camera:



*Hình 3.13 Kết quả theo dõi trong một khung hình trên camera 11*

Sau khi thực hiện việc theo dõi cho từng camera riêng lẻ sẽ cho ra kết quả là các *thời điểm*, *id phương tiện*, *bounding box*, *độ tin cậy* và *đặc trưng* của phương tiện trong từng khung hình. Kết quả này sẽ được lưu vào một file có định dạng là “.pkl” để thực hiện việc liên kết giữa các camera.

Như đã trình bày ở phần 2.4, việc liên kết sử dụng các vùng đánh dấu trên các camera nhằm lấy ra các phương tiện (tracklet) để thực hiện liên kết. Hình ảnh dưới đây là các vùng được đánh dấu trên các camera thuộc tập S03:



*Hình 3.14 Các vùng được đánh dấu trên các camera thuộc tập S03*

Ngoài ra, một thao tác được thực hiện thủ công nữa là định sẵn thời gian di chuyển hợp lệ của các phương tiện (tracklet) giữa các cặp camera.

Cuối cùng, thực hiện việc liên kết phương tiện giao thông giữa các camera. Việc thực hiện liên kết được thử trên nhiều các độ đo khoảng cách khác nhau giữa hai đặc trưng và thuật toán liên kết là **k-láng giềng tương hõ** (k-reciprocal nearest neighbors) [17].

Các thang đo được sử dụng để đánh giá kết quả của phương pháp là các thang đo IDF1, IDP, IDR.

$$\text{IDF1} = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}}$$

$$\text{IDP} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFP}}$$

$$\text{IDR} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}}$$

Với,

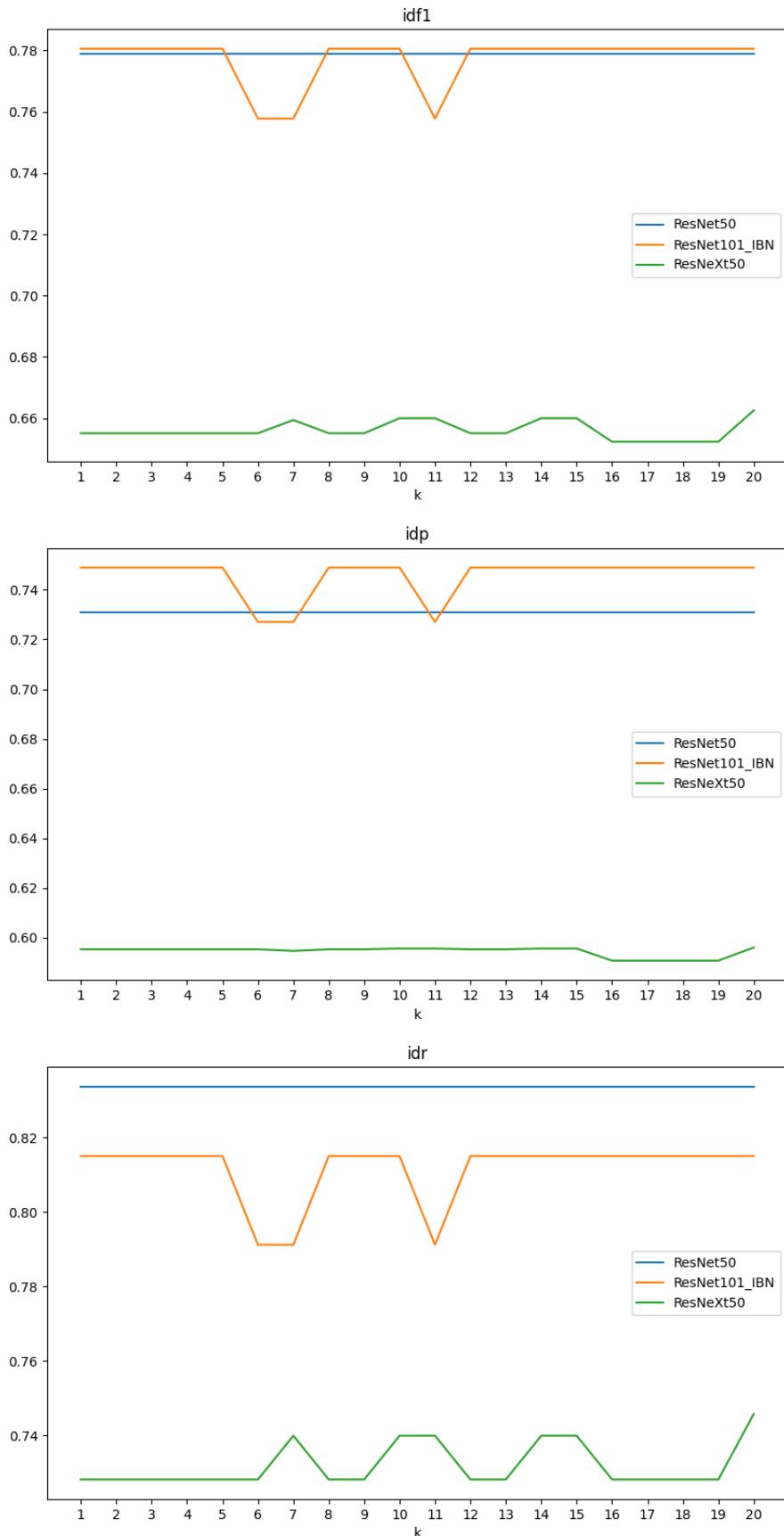
$$\text{IDFN} = \sum_{\tau \in AT} \sum_{t \in T_\tau} m(\tau, \gamma_m(\tau), t, \Delta)$$

$$\text{IDFP} = \sum_{\gamma \in AC} \sum_{t \in T_\gamma} m(\tau_m(\gamma), \gamma, t, \Delta)$$

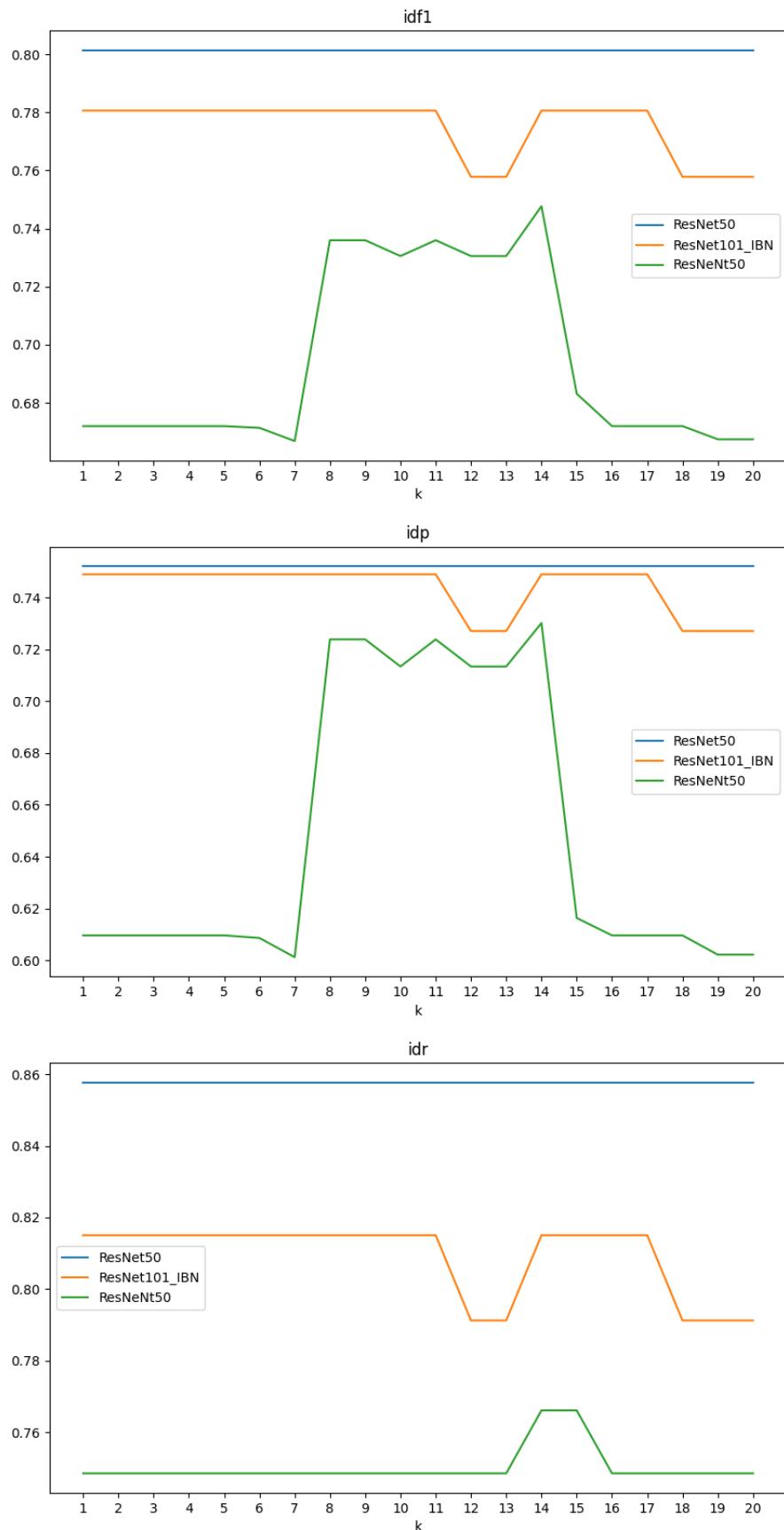
$$\text{IDTP} = \sum_{\gamma \in AC} \text{len}(\gamma) - \text{IDFP} = \sum_{\tau \in AT} \text{len}(\tau) - \text{IDFN}$$

Chi tiết về AT, AC,  $\tau$ ,  $\gamma$  cùng như hàm  $m(\tau, \gamma, t, \Delta)$  được trình bày ở [20].

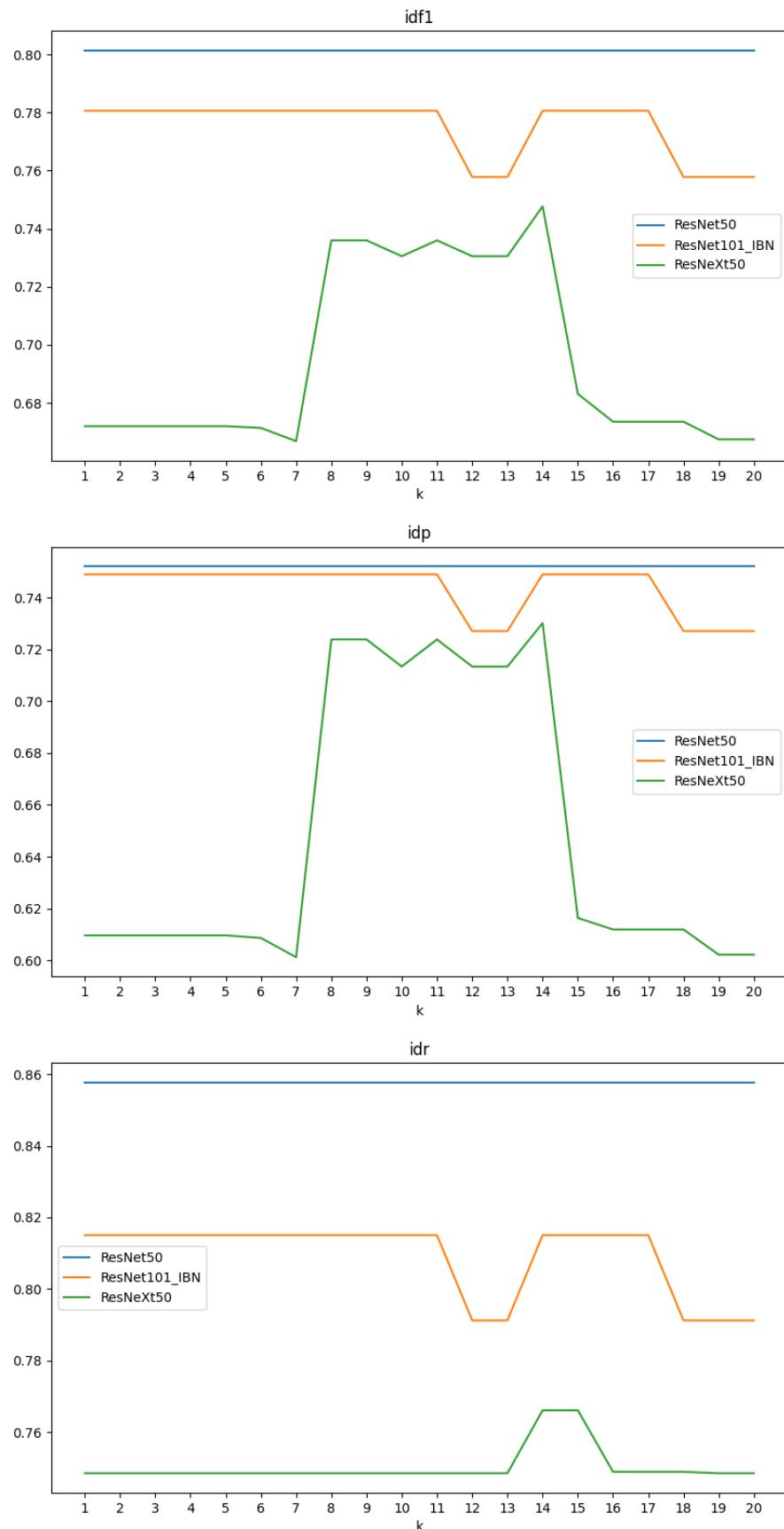
Dưới đây là kết quả IDF1, IDP, IDR khi thực hiện với 20 tham số k khác nhau trên thuật toán k-láng giềng tương hõ với các độ đo để tính ma trận khoảng cách  $D$  là Re-ranking, Euclid và Cosine:



Hình 3.15 Kết quả liên kết giữa các camera của các mô hình trên IDF1, IDP, IDR sử dụng Re-ranking với nhiều tham số k khác nhau



Hình 3.16 Kết quả liên kết giữa các camera của các mô hình trên IDF1, IDP, IDR sử dụng Euclid với nhiều nhiều tham số  $k$  khác nhau



Hình 3.17 Kết quả liên kết giữa các camera của các mô hình trên IDF1, IDP, IDR sử dụng Cosine với nhiều tham số k khác nhau

Dựa vào các hình trên, lựa chọn các tham số k sao cho các mô hình đạt kết quả cao nhất. Dựa vào Hình 3.15, ứng với độ đo khoảng cách Re-ranking, có thể thấy với  $k=15$  thì các mô hình đạt kết quả tốt nhất. Với độ đo Euclid, với  $k=14$  thì các mô hình đạt kết quả tốt nhất. Và đối với độ Cosine,  $k=14$  mang lại kết quả tốt nhất trên các mô hình.

Từ các tham số k đã chọn cho từng độ đo khoảng cách, thực hiện việc liên kết giữa các camera với số lần là 10, các tham số giữ nguyên, dưới đây là kết trung bình quả thu được:

*Bảng 3-2 Kết quả sử dụng mô hình ResNet101-IBN liên kết camera trên tập S03 với số lần lắp là 10, k=15*

Độ đo khoảng cách	IDF1 (%)	IDP (%)	IDR (%)
Cosine	78.05	74.89	81.50
Euclid	78.05	74.89	81.50
Re-ranking	78.05	74.89	81.50

*Bảng 3-3 Kết quả sử dụng mô hình ResNet50 liên kết camera trên tập S03 với số lần lắp là 10, k=14*

Độ đo khoảng cách	IDF1 (%)	IDP (%)	IDR (%)
Cosine	80.14	75.20	85.77
Euclid	80.14	75.20	85.77
Re-ranking	77.89	73.10	83.37

Bảng 3-4 Kết quả sử dụng mô hình ResNeXt50 liên kết camera trên tập S03 với số lần lặp là 10, k=14

Độ đo khoảng cách	IDF1 (%)	IDP (%)	IDR (%)
Cosine	74.76	73.01	76.60
Euclid	74.76	73.01	76.60
Re-ranking	66.00	59.56	73.99

Nhìn chung, kết quả đạt được khá khả quan, thấp nhất là được 66.00% và cao nhất là 80.15% đối với thang đo IDF1. Điều ngạc nhiên là tuy các độ đo Euclid và Cosine đơn giản hơn nhưng kết quả lại cao hơn so với độ đo Re-ranking ở mô hình ResNeXt50 và ResNet101-IBN. Điều này có thể là do trong quá trình thực hiện tính ma trận khoảng cách với Re-ranking, các tham số chưa được tối ưu.

Về phần so sánh kết quả giữa các mô hình, có thể thấy rằng mô hình ResNeXt50 chưa thể hiện tốt so với các mô hình còn lại. Điều này không quá bất ngờ do bộ dữ liệu dùng để huấn luyện mô hình ResNet50 và ResNet101-IBN nhiều hơn so với bộ dữ liệu để huấn luyện mô hình ResNeXt50.

Một lưu ý là kết quả trên chỉ được thực hiện trên một phần nhỏ của tập huấn luyện ban đầu. Hơn thế nữa, trong phần dữ liệu được thực hiện, các phương tiện giao thông ít bị che khuất nên chất lượng đặc trưng thu được cao. Vì thế, kết quả trên chỉ phản ánh được một phần về sự hiệu quả của phương pháp đối với bài toán.

## CHƯƠNG 4. SẢN PHẨM DEMO

### 4.1. Giới thiệu về công cụ Tkinter

Tkinter là một thư viện được tích hợp sẵn trong Python, chuyên dành cho việc phát triển giao diện người dùng đồ họa (GUI). Được xây dựng trên nền tảng toolkit đồ họa Tk, Tkinter cung cấp một cách đơn giản và hiệu quả để tạo ra các ứng dụng có giao diện đồ họa trực quan và dễ sử dụng.

Dù đã được phát hành vào nhiều năm trước, nhưng tkinter đã sớm trở nên nổi tiếng nhờ những điểm mạnh của mình:

- **Dễ học và sử dụng:** Tkinter được biết đến với cú pháp đơn giản và dễ học, làm cho nó là một lựa chọn lý tưởng cho người mới bắt đầu với lập trình GUI trong Python.
- **Tích hợp sẵn trong python:** Tkinter là một phần của thư viện tiêu chuẩn của Python, không yêu cầu cài đặt bổ sung. Điều này giúp giảm độ phức tạp và giữ cho mã nguồn có thể chạy trên nhiều nền tảng mà không cần thay đổi.
- **Đa dạng trong xây dựng giao diện:** Tkinter hỗ trợ nhiều loại widget như nút, ô văn bản, hộp thoại, menu, và nhiều widget khác, cho phép bạn xây dựng giao diện người dùng phong phú và đa dạng.
- **Tương thích đa nền tảng:** Ứng dụng sử dụng Tkinter có thể chạy trên nhiều hệ điều hành khác nhau mà không cần sửa đổi mã nguồn, tăng tính tương thích và di động.

Lợi ích của việc sử dụng Tkinter trong python:

- **Phát triển nhanh chóng:** Tkinter giúp người phát triển nhanh chóng xây dựng giao diện người dùng mà không cần nhiều đoạn mã phức tạp.
- **Phù hợp cho ứng dụng nhỏ và trung bình:** Đối với các ứng dụng nhỏ và trung bình, Tkinter cung cấp một giải pháp hiệu quả và đủ mạnh mẽ để tạo ra giao diện người dùng thân thiện.

- Cộng đồng lớn và tài nguyên phong phú: Tkinter có sự hỗ trợ từ cộng đồng lớn người phát triển Python, điều này có nghĩa là có nhiều tài liệu, ví dụ và hỗ trợ trực tuyến.
- Sự kết hợp với python: Tkinter tích hợp tốt với Python, cho phép bạn kết hợp lập trình hàm và lập trình hướng đối tượng một cách linh hoạt.
- Ứng dụng phổ quát: Tkinter thích hợp cho nhiều loại ứng dụng, từ các ứng dụng desktop đơn giản đến các ứng dụng quản lý dự án và các công cụ biểu đồ.

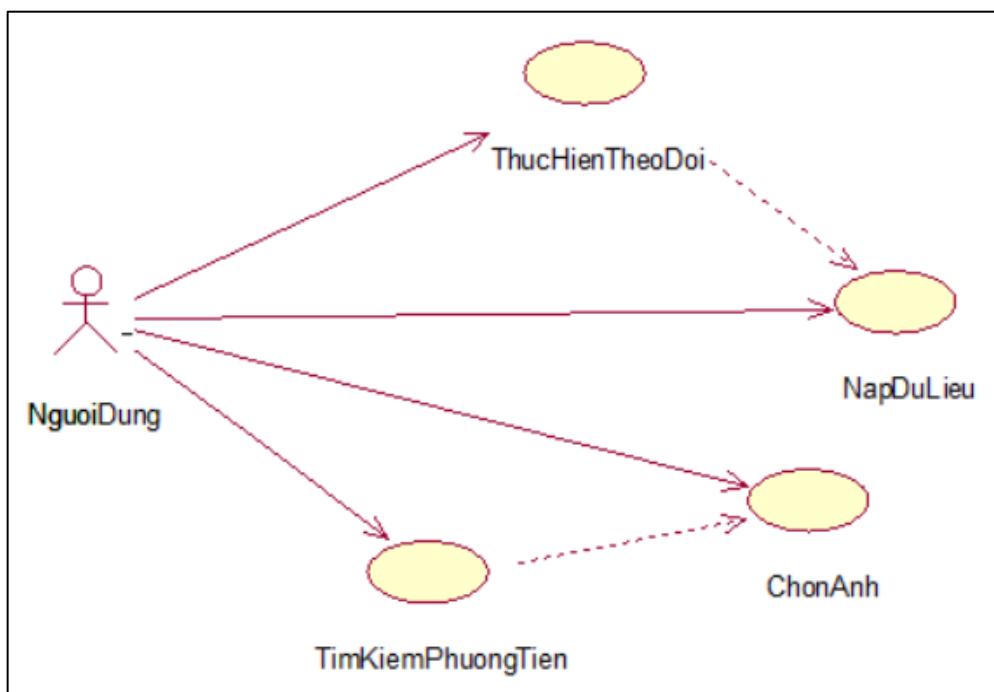
Các bước xây dựng phần mềm nhận diện giới tính và tuổi qua khuôn mặt

1. Đầu tiên, cài đặt thư viện Tkinter bằng lệnh pip install tkinter và import tkinter.
2. Tạo cửa sổ chính của ứng dụng GUI bao gồm các nút chức năng và giao diện tương tác
3. Tạo các màn hình con tương ứng với từng chức năng để thực hiện một chức năng cụ thể.
4. Gọi các màn hình con để liên kết với các chức năng của màn hình chính để tạo thành một ứng dụng hoàn chỉnh.

## 4.2. Phân tích hệ thống

Hệ thống demo cho bài toán trên là một chương trình có giao diện với chức năng chính là thực hiện theo dõi phương tiện giao thông từ nhiều nguồn camera. Ngoài ra, hệ thống còn có chức năng là tìm kiếm phương tiện giao thông trong các camera đã được thực hiện theo dõi.

### 4.2.1. Biểu đồ use case tổng quát



*Hình 4.1 Sơ đồ use case tổng quát*

### 4.2.2. Mô tả chi tiết các use case

#### Use case Nạp dữ liệu:

<b>Tên use case:</b> Nạp dữ liệu
<b>Mô tả tóm tắt:</b> Use case cho phép người dùng nạp dữ liệu vào hệ thống
<b>Luồng sự kiện</b>
<ul style="list-style-type: none"> <li>- Luồng cơ bản:           <ol style="list-style-type: none"> <li>1. Người dùng đặt các tệp tin theo yêu cầu của ứng dụng, ấn nút “Load” trong giao diện của ứng dụng, hệ thống sẽ tự động nạp dữ liệu trong các thư mục đã định sẵn. Khi</li> </ol> </li> </ul>

nạp thành công, hệ thống sẽ hiển thị cảnh của các camera trong dữ liệu lên màn hình.

- **Luồng rẽ nhánh:** Khi hệ thống nạp dữ liệu thất bại, đưa ra thông báo lỗi cụ thể trên màn hình.

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Không có

**Hậu điều kiện:** Không có

**Các điểm mở rộng:** Không có

## Use case Thực hiện theo dõi

**Tên use case:** Thực hiện theo dõi

**Mô tả tóm tắt:** Use case cho phép người dùng thực hiện chạy thuật toán theo dõi phương tiện giao thông từ nhiều camera

### Luồng sự kiện

- **Luồng cơ bản:**
  - Sau khi đã nạp dữ liệu, người dùng nhấn nút “Run” trong giao diện của ứng dụng, hệ thống sẽ tự động chạy chương trình và lưu kết quả ra tệp tin.
- **Luồng rẽ nhánh:**
  - Khi hệ thống chưa dữ liệu, đưa ra thông báo lỗi.
  - Trong quá trình chạy chương trình nếu có lỗi sẽ đưa ra thông báo

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Dữ liệu đã được nạp

**Hậu điều kiện:** Không có

**Các điểm mở rộng:** Không có

## Use case Chọn ảnh

<b>Tên use case:</b> Chọn ảnh
<b>Mô tả tóm tắt:</b> Use case cho phép người dùng chọn ảnh nạp vào hệ thống
<b>Luồng sự kiện</b>
<ul style="list-style-type: none"> <li>- Luồng cơ bản:           <ol style="list-style-type: none"> <li>1. Người dùng ấn nút chọn ảnh trên giao diện của hệ thống. Hệ thống mở cửa sổ để chọn ảnh có trong máy.</li> <li>2. Người dùng chọn ảnh mong muốn và ấn nút “Select”. Hệ thống sẽ nạp ảnh từ máy vào trong ứng dụng và hiển thị ảnh đã chọn lên màn hình.</li> </ol> </li> <li>- Luồng rẽ nhánh: Nếu người dùng chọn tệp tin không phải dạng ảnh, hệ thống sẽ không nạp và hiển thị lên màn hình.</li> </ul>
<b>Các yêu cầu đặc biệt:</b> Không có
<b>Tiền điều kiện:</b> Không có
<b>Hậu điều kiện:</b> Không có
<b>Các điểm mở rộng:</b> Không có

## Use case Tìm kiếm phương tiện

<b>Tên use case:</b> Tìm kiếm phương tiện
<b>Mô tả tóm tắt:</b> Use case cho phép người dùng tìm kiếm phương tiện giao thông trong các dữ liệu dạng video đã nạp và thực hiện việc theo dõi
<b>Luồng sự kiện</b>
<ul style="list-style-type: none"> <li>- Luồng cơ bản:           <ol style="list-style-type: none"> <li>1. Người dùng ấn nút “Search” trên giao diện màn hình, hệ thống thực hiện tìm kiếm phương tiện và trả về kết quả là các video và thời điểm xuất hiện của phương tiện đã chọn để thực hiện tìm kiếm.</li> </ol> </li> </ul>

- Luồng rẽ nhánh: Nếu không có kết quả, hệ thống sẽ báo là không tìm thấy.

**Các yêu cầu đặc biệt:** Không có

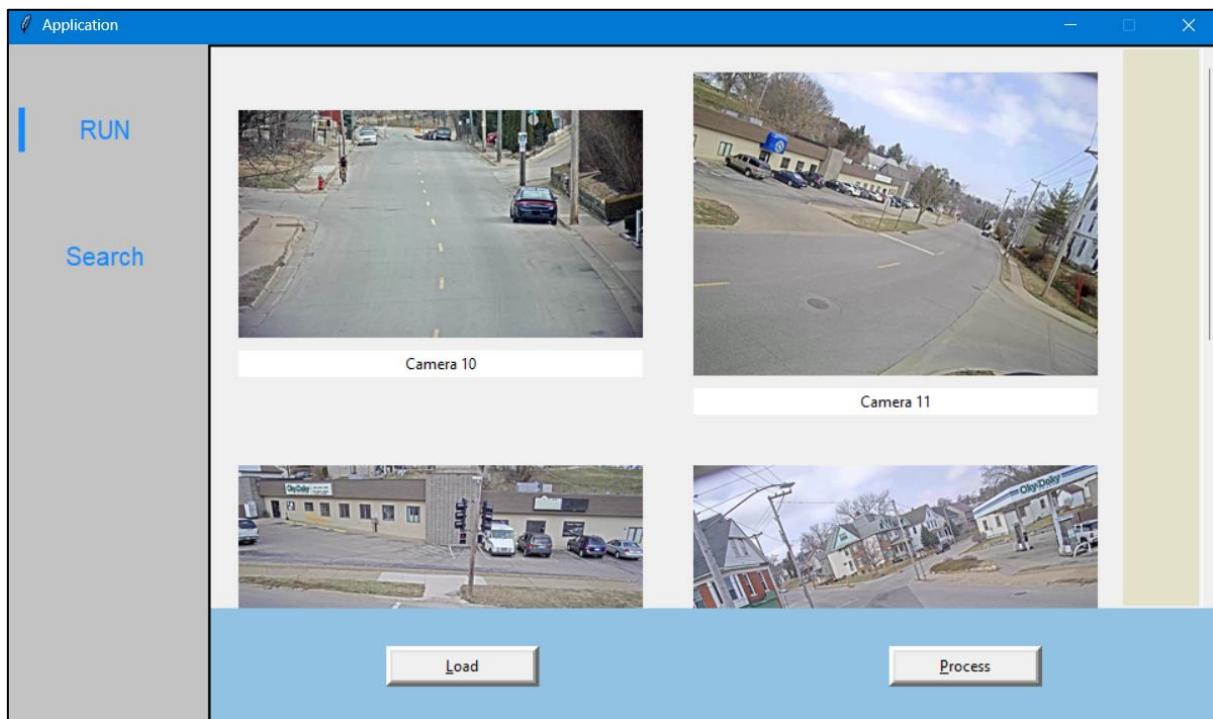
**Tiền điều kiện:** Có tệp tin kết quả của việc theo dõi, ảnh phương tiện đã được chọn.

**Hậu điều kiện:** Không có

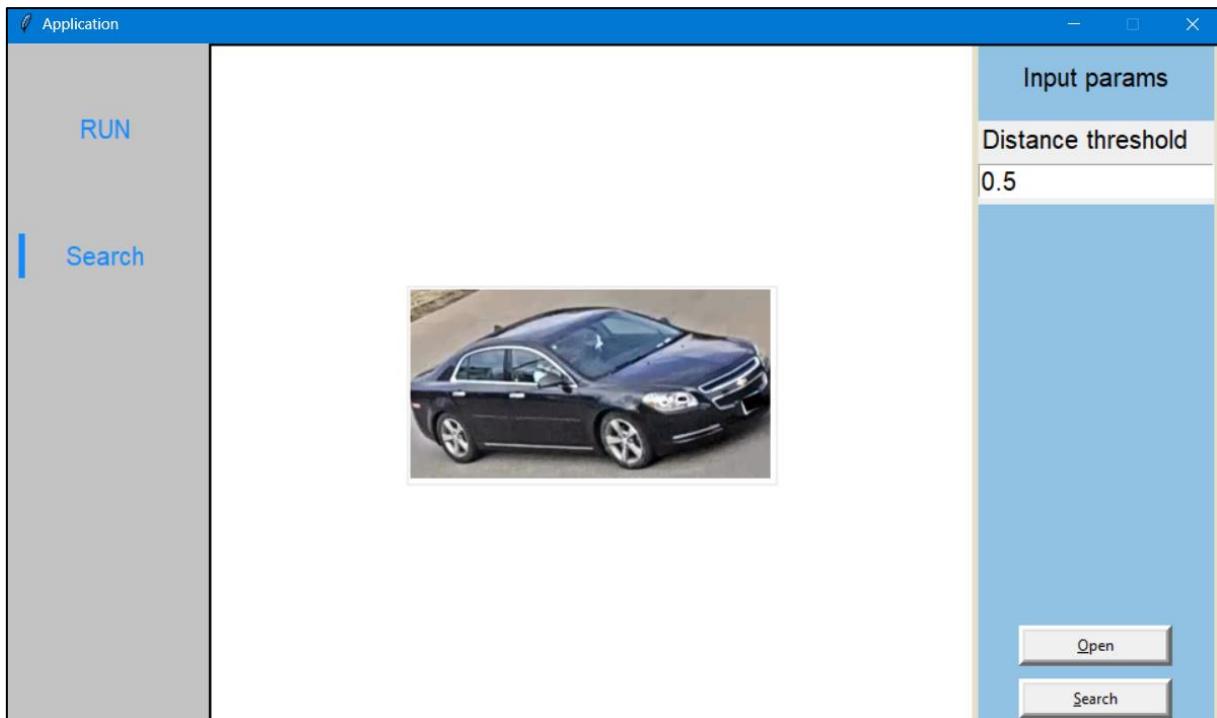
**Các điểm mở rộng:** Không có

### 4.3. Giao diện hệ thống

Dưới đây là giao diện của chương trình demo cho bài toán “Theo dõi phương tiện giao thông từ nhiều nguồn camera” trong phạm vi báo cáo này:



*Hình 4.2 Màn hình chức năng theo dõi phương tiện giao thông từ nhiều camera*



*Hình 4.3 Màn hình thực hiện tìm kiếm phương tiện giao thông*

#### **4.4. Các chức năng của hệ thống**

##### 4.4.1. Chức năng theo dõi phương tiện giao thông từ nhiều camera

Đây là chức năng chính của hệ thống, chức năng sẽ xử lý các video được cấp bởi người dùng và một vài thông tin kèm theo. Các thông tin đó là các vùng đánh dấu trong từng video, danh sách kè giờ giữa các camera, thời gian di chuyển giữa các camera có đường đi nối nhau. Các thông tin này được đặt trong một thư mục có tên là ‘input’.

Kết quả đầu ra của chức này bao gồm kết quả theo dõi trên từng camera và kết quả sau khi thực hiện liên kết giữa các camera. Kết quả theo dõi trên một camera là tệp tin chứa các ID, bounding box, đặc trưng của phương tiện giao thông theo định dạng MOT. Kết quả của việc thực hiện liên kết giữa các camera cũng tương tự như kết quả trên một camera nhưng các ID của phương tiện đã được liên kết lại với nhau, tạo thành ID toàn cục.

15	1	3	351	346	122	119	-1	-1
15	1	4	351	346	122	119	-1	-1
15	1	5	352	346	120	119	-1	-1
15	1	6	352	346	120	119	-1	-1
15	1	7	352	346	120	119	-1	-1
15	1	8	352	346	120	119	-1	-1
15	1	9	353	346	120	118	-1	-1
15	1	10	353	346	120	118	-1	-1
15	1	11	353	346	120	119	-1	-1
15	1	12	353	346	120	119	-1	-1
15	1	13	355	346	118	118	-1	-1
15	1	14	355	346	118	118	-1	-1
15	1	15	355	346	118	118	-1	-1
15	1	16	355	346	118	118	-1	-1
15	1	17	352	346	121	118	-1	-1
15	1	18	352	346	121	118	-1	-1
...								

Hình 4.4 Kết quả theo dõi trên từng camera ở dạng txt

#### 4.4.2. Chức năng tìm kiếm phương tiện giao thông

Chức năng này thực hiện tìm kiếm phương tiện giao thông trên kết quả theo dõi phương tiện giao thông từ nhiều camera. Chức năng yêu cầu người dùng chọn một ảnh chỉ chứa phương tiện giao thông cần tìm và nạp vào hệ thống. Hệ thống sẽ cho ra kết quả là ID của phương tiện, các camera đã đi qua và thời điểm xuất hiện trên video. Ngược lại, hệ thống sẽ báo là không tìm thấy phương tiện.

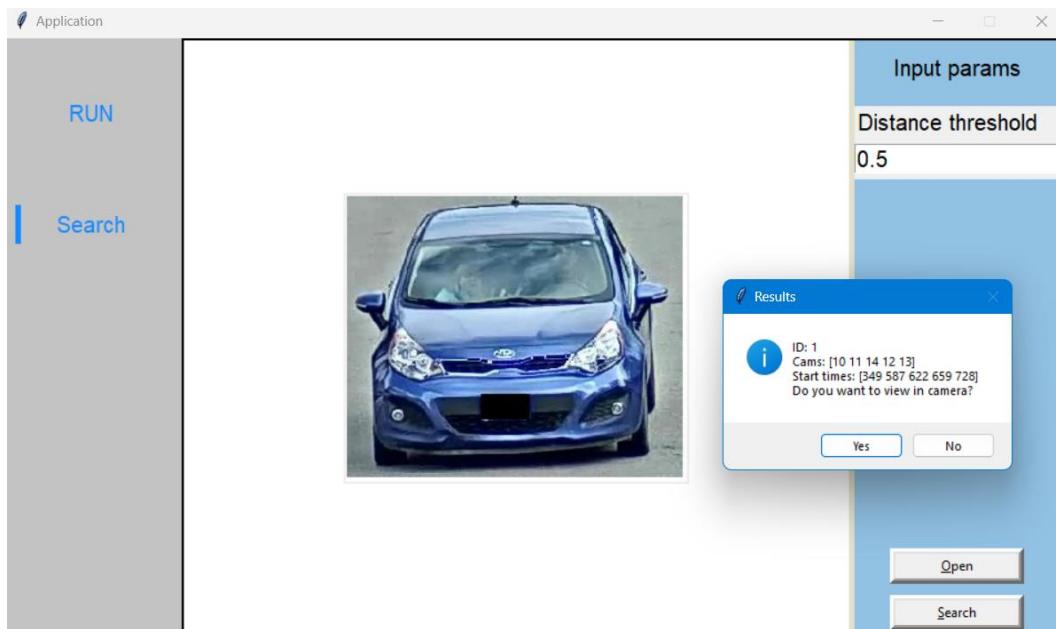
Ngoài ra, người dùng có thể điều chỉnh ngưỡng để khi tìm kiếm phương tiện giao thông. Độ đo để thể hiện sự tương đồng giữa phương tiện cần tìm và phương tiện có trong video là trung bình độ của đo euclid giữa đặc trưng phương tiện cần tìm và đặc trưng của toàn bộ phương tiện đó theo thời gian trong một video:

$$d(q, T) = \frac{1}{n} \sum_{i=1}^n \|q - T_i\|_2$$

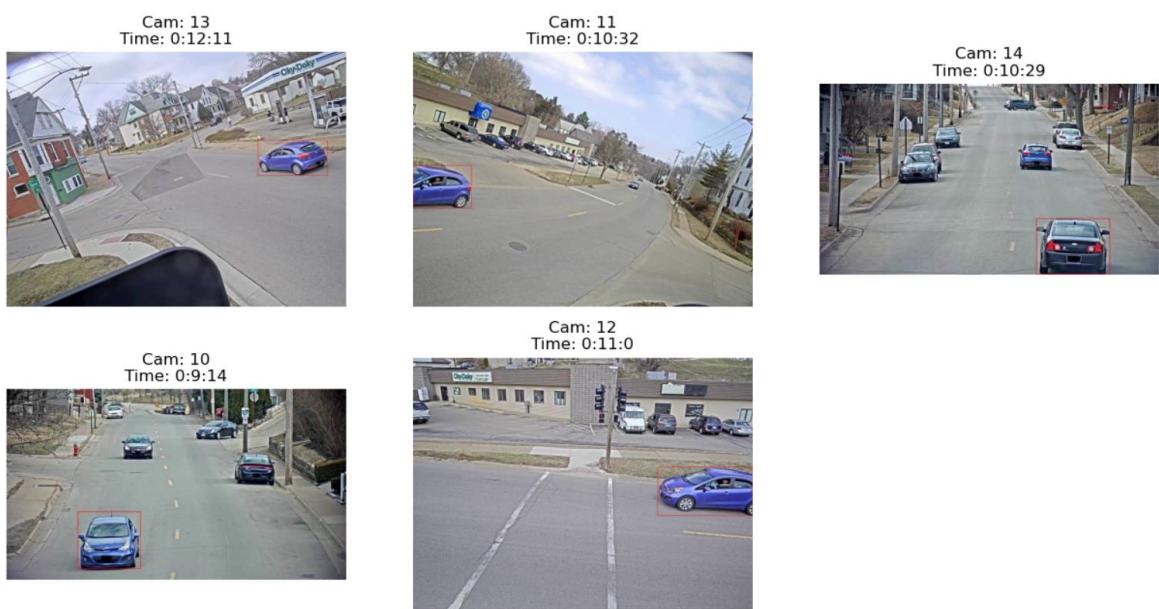
Với, q là đặc trưng của phương tiện trong ảnh cần tìm, T là tập các đặc trưng của một phương tiện có trong video, n là số đặc trưng có trong T.

Từ biểu thức trên, có thể thấy, d càng gần 0 càng thể hiện phương tiện cần tìm và phương tiện đang xét có sự tương đồng cao.

Dưới đây là kết quả khi thực hiện tìm kiếm một phương tiện giao thông có trong bộ dữ liệu đầu vào:



*Hình 4.5 Màn hình kết quả cho chức năng tìm kiếm phương tiện giao thông khi tìm thấy phương tiện (1)*



*Hình 4.6 Màn hình kết quả cho chức năng tìm kiếm phương tiện giao thông khi tìm thấy phương tiện (2)*

Còn đây là thực hiện tìm kiếm phương tiện giao thông không có trong video:



Hình 4.7 Màn hình kết quả cho chức năng tìm kiếm phương tiện giao thông khi không tìm thấy phương tiện

## KẾT LUẬN

Qua nội dung báo cáo trên, có thể nhận thấy các kỹ thuật sử dụng vào trong đề tài “Xây dựng hệ thống theo dõi phương tiện giao thông từ nhiều camera” đều là những kỹ thuật không quá mới. Tuy vậy, kết quả cao nhất thu được khi áp dụng vào bài toán là 80% trên thang đo IDF1, đây là một kết quả tốt và còn có thể được cải thiện thêm.

Tuy đạt được kết quả khả quan, song bài toán vẫn còn những thách thức lớn hơn. Thứ nhất, các phương tiện bị che lấp ảnh hưởng rất nhiều đến kết quả của bài toán. Thứ hai, việc liên kết giữa phương tiện các camera đòi hỏi việc đánh dấu các vùng trên camera, đưa ra các ràng buộc về thời gian một cách thủ công. Hai điều này làm cho khả năng áp dụng của bài toán vào trong thực tế giảm xuống khi có nhiều camera.

Một đề xuất để kết quả có thể tốt hơn là sử dụng những mô hình, kỹ thuật mới và tốt cho bài toán tái nhận diện, bài toán quyết định đến kết quả liên kết giữa các camera. Cụ thể hơn như sử dụng kỹ thuật “Recall@k Surrogate Loss with Large Batches and Similarity Mixup” để huấn luyện mô hình nhằm đạt được kết quả tốt.

Cuối cùng, sản phẩm demo nhỏ có hai chức năng là thực hiện theo dõi phương tiện giao thông từ nhiều nguồn camera và tìm kiếm phương tiện giao thông từ các camera được xây dựng. Sản phẩm là một ứng dụng nhỏ trong vô vàn các ứng dụng khác của bài toán theo dõi phương tiện giao thông từ nhiều nguồn camera.

## TÀI LIỆU THAM KHẢO

- [1] M. Naphade and S. Wang and D. C. Anastasiu and Z. Tang and M. Chang and Y. Yao and L. Zheng and M. Shaiqur Rahman and A. Venkatachalapathy and A. Sharma and Q. Feng and V. Ablavsky and S. Sclaroff and P. Chakraborty and A. Li and S. Li and R. Chellappa, "The 6th AI City Challenge," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE Computer Society, 2022, pp. 3346-3355.
- [2] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440-1448.
- [3] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, 2017, pp. 1137-1149.
- [4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft, "SIMPLE ONLINE AND REALTIME TRACKING," 2016.
- [5] Nicolai Wojke, Alex Bewley, Dietrich Paulus, "SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC," 2017.
- [6] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, Xinggang Wang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," 2021.
- [7] M. Millar, "Review of Current Methods for Re-Identification in Computer Vision," 2019.

- [8] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, Tao Mei, "VehicleNet: Learning Robust Visual Representation," 2020.
- [9] Luo, Hao and Chen, Weihua and Xu, Xianzhe and Gu, Jianyang and Zhang, Yuqi and Liu, Chong and Jiang, Yiqi and He, Shuting and Wang, Fan and Li, Hao, "An Empirical Study of Vehicle Re-Identification on the AI City Challenge," 2021.
- [10] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training b," 2015.
- [11] Dmitry Ulyanov, Andrea Vedaldi, "Instance Normalization:," 2016.
- [12] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang, "Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net," in *ECCV*, 2018.
- [13] He, Shuting and Luo, Hao and Chen, Weihua and Zhang, Miao and Zhang, Yuqi and Wang, Fan and Li, Hao and Jiang, Wei, "Multi-Domain Learning and Identity Mining for Vehicle Re-Identification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2485-2493.
- [14] Luo, Hao and Gu, Youzhi and Liao, Xingyu and Lai, Shenqi and Jiang, Wei, "Bag of Tricks and a Strong Baseline for Deep Person Re-Identification," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1487-1495.
- [15] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, "Aggregated Residual Transformations for Deep Neural Networks," 2016.
- [16] Yang, Xipeng and Ye, Jin and Lu, Jincheng and Gong, Chenting and Jiang, Minyue and Lin, Xiangru and Zhang, Wei and Tan, Xiao and Li, Yingying

- and Ye, Xiaoqing and Ding, Errui, "Box-Grained Reranking Matching for Multi-Camera Multi-Target Tracking," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE Computer Society, 2022, pp. 3095-3105.
- [17] Zhong, Zhun, Zheng, Liang, Cao, Donglin, Li, Shaozi, "Re-ranking Person Re-identification with k-reciprocal Encoding," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3652-3661.
- [18] Xincheng Liu, Wu Liu, Tao Mei, Huadong Ma, "PROVID: Progressive and Multimodal Vehicle Reidentification for Large-Scale Urban Surveillance," *IEEE Trans. Multimedia*, vol. PP, 2017.
- [19] Liu, Chong and Zhang, Yuqi and Luo, Hao and Tang, Jiasheng and Chen, Weihua and Xu, Xianzhe and Wang, Fan and Li, Hao and Shen, Yi-Dong, "City-Scale Multi-Camera Vehicle Tracking Guided by Crossroad Zones," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 4124-4132.
- [20] Ergys Ristani and Francesco Solera and Roger S. Zou and Rita Cucchiara and Carlo Tomasi, "Performance Measures and a Data Set for," in *Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science*, 2016.