**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 10: Xây dựng hệ thống nhận diện biển số xe tự động với OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210235 | Nguyễn Quang Tân | DCCNTT12.10.1 |
| **2** | 20210201 | Nguyễn Hoàng Phong | DCCNTT12.10.1 |
| **3** | 20210136 | Vũ Đình Thịnh | DCCNTT12.10.1 |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 10: Xây dựng hệ thống nhận diện biển số xe tự động với OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210235 | Nguyễn Quang Tân | DCCNTT12.10.1 |
| **2** | 20210201 | Nguyễn Hoàng Phong | DCCNTT12.10.1 |
| **3** | 20210136 | Vũ Đình Thịnh | DCCNTT12.10.1 |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 10**  **Tên học phần:**  **Xử lý ảnh và thị giác máy tính**  **Lớp Tín chỉ: XATGMT.03.K12.01.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  **Lương Thị Hồng Lan** | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | Nguyễn Quang Tân | Nguyễn Hoàng Phong | Vũ Đình Thịnh |
| --- | --- | --- | --- | --- | --- |
| 20210235 | 20210201 | 20210136 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |

**Mục lục**

[LỜI MỞ ĐẦU 10](#_Toc184658344)

[LỜI CẢM ƠN 11](#_Toc184658345)

[Chương 1 : Tổng quan về đề tài 14](#_Toc184658346)

[**1.1 Tổng Quan Về Thị Giác Máy Tính** 14](#_Toc184658347)

[**1.2 Tổng Quan Về Học Máy** 15](#_Toc184658348)

[1.2.2 Học không giám sát (Unsupervised Learning) 17](#_Toc184658349)

[1.3 Các phương pháp sử dụng 19](#_Toc184658350)

[2.1 Yêu cầu bài toán 23](#_Toc184658351)

[2.2.2 Xử lý ảnh và Open CV 27](#_Toc184658352)

[Hướng giải quyết 27](#_Toc184658353)

[2.2.3 Phát hiện vị trí và tách biển số 28](#_Toc184658354)

[2.2.4 Phân đoạn ký tự 31](#_Toc184658355)

[Hướng giải quyết 31](#_Toc184658356)

[Xoay biển số 31](#_Toc184658357)

[Tìm vùng đối tượng 33](#_Toc184658358)

[2.2.5 Tìm và tách kí tự 34](#_Toc184658359)

[Thuật toán KNN (K - Nearest Neighbor) 34](#_Toc184658360)

[Hướng giải quyết 36](#_Toc184658361)

[**3.1 Dữ liệu huấn luyện** 38](#_Toc184658362)

[Kết luận 47](#_Toc184658363)

[Danh mục tham khảo 49](#_Toc184658364)

**DANH MỤC HÌNH ẢNH**

Hình 1.1 Hình ảnh về học máy .............................................................................. 11

Hình 1.2 Ví dụ về học có giám sát ....................................................................... 12

Hình 1.3 Ví dụ về học không giám sát .................................................................. 13

Hình 1.4 Ví dụ về học bán giám sát ...................................................................... 15

Hình 1.5 Ví dụ về KNN......................................................................................... 16

Hình 1.6 Tập dữ liệu huấn luyện........................................................................... 17

Hình 1.7 Biển số trước khi nhận diện................................................................... 18

Hình 1.8 Biển số sau khi nhận diện ..................................................................... 18

Hình 1.9 Hình ảnh bộ giữ liệu ............................................................................. 20

Hình 1.10 Các bước chính trong phân đoạn trình tự............................................ 22

Hình 1.11.1 Ảnh biển số chưa xoay..................................................................... 23

Hình 1.11.2 Ảnh biển số đã xoay......................................................................... 23

Hình 1.12.1 Ảnh nhị phân ................................................................................... 24

Hình 1.12.2 Tìm vùng đối tượng ......................................................................... 24

Hình 1.13 Ảnh kí tự sau khi cắt............................................................................ 25

Hình 1.14 Ngôn ngữ Python................................................................................. 25

Hình 2.1 Mô hình nhận diện biển số xe................................................................ 30

Hình 3.1 Giao diện Input...................................................................................... 31

Hình 3.2 Giao diện Output .................................................................................... 31

Hình 3.3.1 Ảnh gốc ............................................................................................... 40

Hình 3.3.2 Hình ảnh không tìm thấy kí tự............................................................. 40

Hình 3.3.3 Tìm thấy 5 kí tự................................................................................... 40

Hình 3.3.4 Tìm thấy 9 kí tự................................................................................... 40

Hình 3.3.5 Lấy ngưỡng cao /rộng 1.5 .................................................................. 41

Hình 3.3.6 Lấy ngưỡng cao /rộng 1.4................................................................... 41

Hình 3.4.1 Ảnh gốc nhận diện 3 biển số ............................................................ 42

Hình 3.4.2 Biển số 1 ............................................................................................42

Hình 3.4.3 Biển số 2 ........................................................................................... 42

Hình 3.4.4 Biển số 3 ........................................................................................... 42

Hình 3.4.5 Không khoanh được vùng kí tự ........................................................ 42

# LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0, khi các hệ thống tự động hóa và trí tuệ nhân tạo ngày càng được ứng dụng rộng rãi, việc áp dụng thị giác máy tính vào các lĩnh vực thực tiễn đã mang lại nhiều giải pháp đột phá. Một trong những ứng dụng quan trọng của thị giác máy tính là nhận diện biển số xe tự động, hỗ trợ tích cực trong việc quản lý giao thông, giám sát an ninh, và tự động hóa các hệ thống bãi đỗ xe.

Việc xây dựng một hệ thống nhận diện biển số xe không chỉ yêu cầu khả năng xử lý và phân tích hình ảnh mà còn đòi hỏi sự tích hợp các thuật toán mạnh mẽ để nhận diện và trích xuất thông tin chính xác trong thời gian thực. OpenCV, một thư viện mã nguồn mở nổi tiếng trong lĩnh vực xử lý ảnh, cung cấp nhiều công cụ mạnh mẽ và linh hoạt để thực hiện các tác vụ này.

Đề tài “Xây dựng chương trình nhận diện biển số xe tự động bằng OpenCV” không chỉ mang lại cơ hội tìm hiểu và ứng dụng các kỹ thuật xử lý ảnh và nhận diện ký tự, mà còn mở ra tiềm năng xây dựng những giải pháp hiện đại phục vụ thực tế. Mục tiêu của chương trình là phát hiện biển số xe trong hình ảnh hoặc video, trích xuất thông tin ký tự trên biển số, và đảm bảo độ chính xác cao trong các điều kiện môi trường khác nhau.

Trong bài viết này, chúng em sẽ trình bày chi tiết các bước thực hiện, từ việc xử lý ảnh, nhận diện biển số đến trích xuất ký tự, cũng như đánh giá hiệu quả của hệ thống qua các thử nghiệm thực tế. Hy vọng rằng nghiên cứu này sẽ góp phần vào việc phát triển các hệ thống thông thông. Đồng thời, chúng em mong muốn đề tài sẽ là một bước đệm để tiếp tục khám phá những tiềm năng ứng dụng rộng lớn của công nghệ xử lý ảnh và thị giác máy tính trong tương lai.minh, cung cấp những giải pháp hiệu quả trong các lĩnh vực quản lý phương tiện và giao

# LỜI CẢM ƠN

Trong suốt quá trình học môn Xử lý ảnh và Thị giác máy tính, em đã có một trải nghiệm học tập đầy thú vị và bổ ích. Em xin gửi lời cảm ơn sâu sắc đến cô giáo Lương Thị Hồng Lan đã tận tâm giảng dạy và truyền đạt những kiến thức quý báu cho chúng em. Môn học không chỉ cung cấp cho em nền tảng lý thuyết về xử lý ảnh số mà còn mở ra một thế giới phong phú về các thuật toán thị giác máy tính tiên tiến, giúp em nhận thức rõ hơn về sự phát triển mạnh mẽ của công nghệ hiện đại. Những kiến thức cơ bản về biểu diễn ảnh số, biến đổi Fourier, hay những phương pháp phức tạp như phân đoạn ảnh, nhận diện đối tượng và các khái niệm về học sâu trong thị giác máy tính đều được cô giáo đã truyền đạt một cách chi tiết và dễ hiểu.

Điều em ấn tượng nhất là cách cô kết hợp lý thuyết và thực hành một cách chặt chẽ. Các bài thực hành trên máy tính không chỉ giúp em củng cố kiến thức mà còn rèn luyện khả năng lập trình và kỹ năng giải quyết vấn đề. Việc được trực tiếp triển khai các thuật toán, từ tiền xử lý ảnh, trích xuất đặc trưng đến huấn luyện mô hình học máy, đã giúp em hiểu rõ hơn về cách thức hoạt động của các hệ thống thị giác máy tính và những thách thức mà chúng phải đối mặt. Một trong những bài thực hành em nhớ nhất là việc phân loại các loại trái cây trong ảnh bằng CNN. Qua đó, em không chỉ áp dụng được lý thuyết vào thực tiễn mà còn học được cách cải thiện hiệu suất mô hình, điều chỉnh tham số và đánh giá kết quả.

Sự kiên nhẫn và nhiệt huyết của cô trong việc giải đáp thắc mắc, hướng dẫn và tạo ra môi trường học tập tích cực là yếu tố quan trọng giúp em hoàn thành tốt môn học. Cô không chỉ là người giảng dạy mà còn là người truyền cảm hứng, khích lệ và động viên em trong suốt hành trình học tập. Những lời khuyên và sự quan tâm của thầy/cô đã giúp em tự tin hơn trong việc tiếp cận và hiểu sâu các kiến thức mới.

Em tin rằng những kỹ năng và kiến thức em có được từ môn học này sẽ là hành trang quý giá cho em trong quá trình học tập và phát triển sự nghiệp sau này. Em xin một lần nữa gửi lời cảm ơn chân thành đến thầy/cô và hy vọng sẽ tiếp tục nhận được sự hướng dẫn quý báu trong các môn học tiếp theo.

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên công việc** | **Công việc chia đến nhỏ nhất** | **Thành viên thực hiện** |
| 1 | Viết báo cáo | Word chương 1 | Vũ Đình Thịnh |
| Word chương 2 | Nguyễn Quang Tân |
| Viết báo cáo chương 3 | Nguyễn Hoàng Phong |
| Viết, sửa kiểm tra lại và hoàn  thiện báo cáo chương 3 | Nguyễn Quang Tân |
| Viết,chỉnh sửa lại, frommat lại bài  trước khi in quyển | Nguyễn Hoàng Phong, Vũ Đình Thịnh |
| Viết kết luận sau khi đã hoàn  thành | Vũ Đình Thịnh |
| 2 | Code | Code giao diện | Vũ Đình Thịnh |
| Code phần phát hiện vùng biển | Nguyễn Hoàng Phong |
| Code Phát hiện chữ biển số xe | Nguyễn Quang Tân |
| 3 | Slide báo cáo | Thêm nội dung chi tiết vào slide | Nguyễn Quang Tân |
| Hoàn thiện và chỉnh sửa | Nguyễn Hoàng Phong |

# Chương 1 : Tổng quan về đề tài

**1.1 Tổng Quan Về Thị Giác Máy Tính**

    Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cửu, ửng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

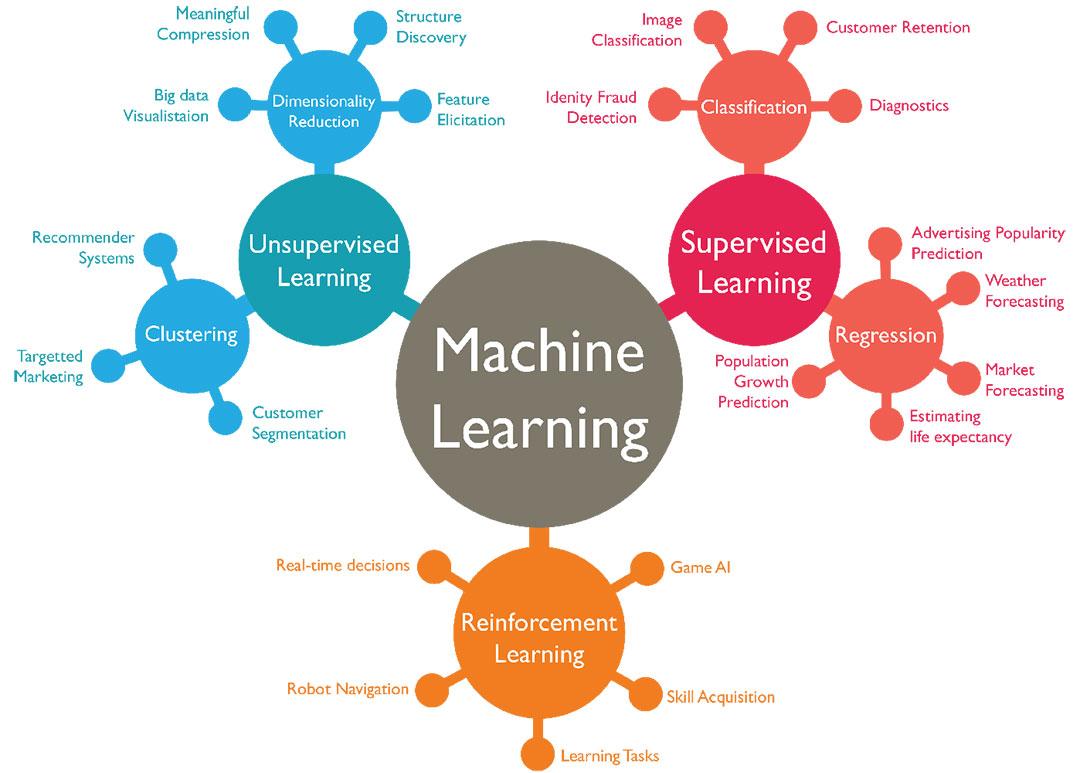
    Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thửc cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số là một môn học hết sửc cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn… Thử hai, các công cụ toán như Đại số tuyến tính, Sác xuất, thống kê. Một số kiến thử cần thiết như Trí tuệ nhân tao, Mạng nơ ron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

     Các phương pháp xử lý ảnh bắt đầu từ các ửng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mửc sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thử hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh sô thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thửc nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

     Để dễ tưởng tượng, xét các bước cần thiết trong xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu CCIR). Gần đây, với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. (Máy ảnh số hiện nay là một thí dụ gần gũi). Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh; có thể quét từ ảnh chụp bằng máy quét ảnh.

**1.2 Tổng Quan Về Học Máy**

Học máy (Machine Learning - ML) là một lĩnh vực con của trí tuệ nhân tạo (Artificial Intelligence - AI), tập trung vào việc phát triển các thuật toán và mô hình giúp máy tính có khả năng học từ dữ liệu để thực hiện các tác vụ mà không cần lập trình rõ ràng. Học máy đã trở thành một phần quan trọng của khoa học dữ liệu và được ứng dụng rộng rãi trong nhiều lĩnh vực như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, dự đoán kinh doanh, và xe tự hành.



*Hình 1.1 Hình ảnh về học máy*

**1.2.1 Học có giám sát (Supervised Learning)**

Khái niệm:

     Học có giám sát là một phương pháp học máy sử dụng một tập dữ liệu huấn luyện bao gồm các cặp đầu vào và đầu ra đã được gán nhãn. Mục tiêu là xây dựng một mô hình có thể học mối quan hệ giữa đầu vào và đầu ra để dự đoán chính xác nhãn cho các dữ liệu mới.

* Ví dụ:

Phân loại email: Phân loại email thành spam hoặc không spam dựa trên các đặc trưng của nội dung email.

Dự đoán giá nhà: Dự đoán giá nhà dựa trên các đặc trưng như diện tích, số phòng, và vị trí.

* Ưu điểm:

Hiệu suất cao: Khi có đủ dữ liệu huấn luyện, mô hình thường cho kết quả chính xác cao.

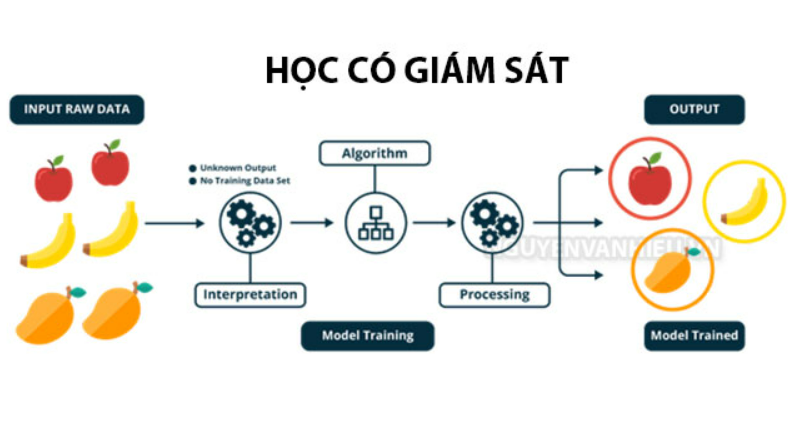
Dễ triển khai: Các thuật toán như hồi quy tuyến tính, cây quyết định, và mạng nơ-ron thường dễ triển khai và tối ưu hóa.

Dễ giải thích: Một số mô hình như hồi quy tuyến tính hoặc cây quyết định có thể dễ dàng giải thích.

* Nhược điểm:

Cần nhiều dữ liệu có nhãn: Việc thu thập và gán nhãn cho dữ liệu có thể tốn kém và mất thời gian.

     Có thể bị quá khớp (Overfitting): Nếu mô hình quá phức tạp, nó có thể học    quá kỹ các chi tiết của dữ liệu huấn luyện và không tổng quát hóa tốt trên dữ liệu mới.



*Hình 1.2 Ví dụ về học có giám sát*

### 1.2.2 Học không giám sát (Unsupervised Learning)

#### Khái niệm:

               Học không giám sát là một phương pháp học máy sử dụng một tập dữ liệu chỉ có đầu vào mà không có nhãn đầu ra. Mục tiêu là tìm kiếm các mẫu ẩn, cấu trúc hoặc nhóm trong dữ liệu.

* Ví dụ:

Phân cụm khách hàng: Nhóm khách hàng thành các nhóm dựa trên hành vi mua sắm của họ.

Giảm chiều dữ liệu: Sử dụng PCA (Phân tích thành phần chính) để giảm số chiều của dữ liệu nhằm tối ưu hóa xử lý.

* Ưu điểm

Không cần dữ liệu có nhãn: Dễ dàng thu thập và xử lý dữ liệu mà không cần gán nhãn

Tìm kiếm mẫu ẩn: Có khả năng khám phá các mẫu hoặc cấu trúc ẩn mà không được biết trước.

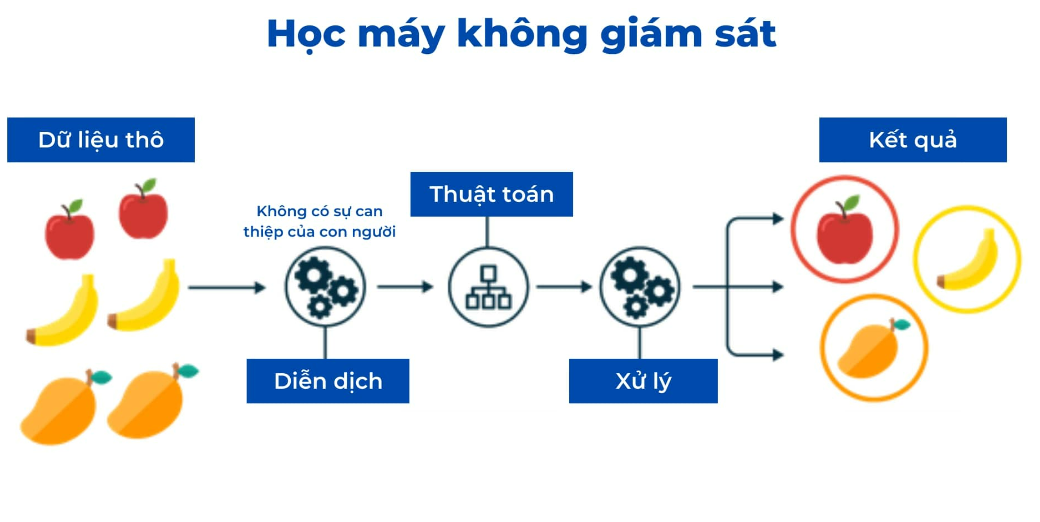
Hữu ích cho tiền xử lý dữ liệu: Có thể sử dụng để giảm nhiễu hoặc giảm chiều dữ liệu trước khi áp dụng các thuật toán khác.

* Nhược điểm:

Khó đánh giá chất lượng: Không có nhãn để đánh giá kết quả chính xác.

Kết quả khó giải thích: Các mẫu tìm được có thể không có ý nghĩa rõ ràng về mặt thực tiễn.

Dễ nhầm lẫn: Các thuật toán có thể nhầm lẫn khi dữ liệu không có cấu trúc rõ ràng

****

*Hình 1.3 Ví dụ về học không giám sát*

**1.2.3 Học bán giám sát (Semi-Supervised Learning)**

* Khái niệm:

Học bán giám sát là một phương pháp kết hợp giữa học có giám sát và học không giám sát. Nó sử dụng một lượng nhỏ dữ liệu có nhãn kết hợp với lượng lớn dữ liệu chưa có nhãn để huấn luyện mô hình. Phương pháp này tận dụng được sức mạnh của cả hai loại học.

* Ví dụ:

Nhận diện khuôn mặt: Một lượng nhỏ ảnh khuôn mặt được gán nhãn (khuôn mặt của ai), trong khi phần lớn ảnh không được gán nhãn.

Hệ thống khuyến nghị: Sử dụng một lượng nhỏ phản hồi người dùng có nhãn (thích hoặc không thích), kết hợp với lượng lớn dữ liệu chưa có nhãn về hành vi người dùng.

* Ưu điểm:

Tiết kiệm chi phí gán nhãn: Chỉ cần một lượng nhỏ dữ liệu có nhãn, giảm chi phí và thời gian gán nhãn.

Tận dụng tốt dữ liệu chưa có nhãn: Có thể cải thiện hiệu suất của mô hình bằng cách sử dụng thông tin từ dữ liệu chưa có nhãn.

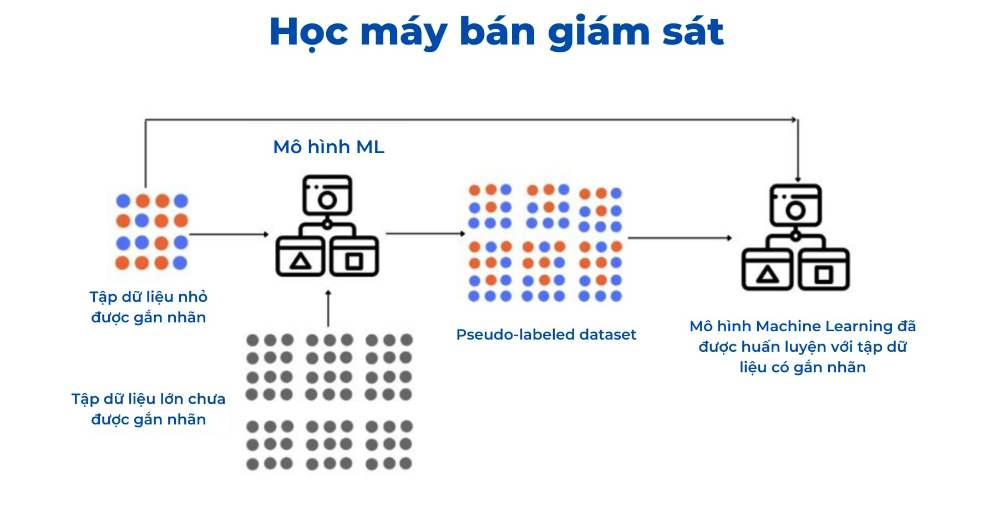
Giảm hiện tượng quá khớp: Việc sử dụng nhiều dữ liệu chưa có nhãn có thể giúp mô hình tổng quát hóa tốt hơn.

* Nhược điểm:

Rủi ro tích lũy lỗi: Nếu mô hình dự đoán sai nhãn cho dữ liệu chưa có nhãn, lỗi này có thể tích lũy và ảnh hưởng đến hiệu suất của mô hình.

Phức tạp hơn khi triển khai: Việc kết hợp và xử lý dữ liệu có nhãn và chưa có nhãn đòi hỏi kỹ thuật phức tạp hơn.

Yêu cầu dữ liệu chưa có nhãn phải đồng nhất: Nếu dữ liệu chưa có nhãn khác biệt quá nhiều so với dữ liệu có nhãn, hiệu suất có thể bị ảnh hưởng.

****

*Hình 1.4 Ví dụ về học bán giám sát*

## 1.3 Các phương pháp sử dụng

1.3.1 YOLOv5 (You Only Look Once version 5)

Là một thuật toán phát hiện đối tượng (object detection) sử dụng học sâu (deep learning), nổi bật với tốc độ nhanh và hiệu suất cao. Đây là phiên bản cải tiến của dòng YOLO (You Only Look Once), được phát triển bởi cộng đồng sau YOLOv4. YOLOv5 không phải là sản phẩm chính thức từ tác giả YOLO gốc (Joseph Redmon) mà được phát triển bởi công ty Ultralytics.

Thuật toán này có thể thực hiện nhận diện đối tượng và phân vùng (bounding box) cho nhiều loại đối tượng khác nhau trong ảnh, đồng thời cung cấp thông tin về vị trí và loại đối tượng.

- Ưu điểm của YOLOv5:

1. Hiệu suất cao

Tốc độ nhanh: YOLOv5 được tối ưu hóa tốt cho cả CPU và GPU, cho phép phát hiện đối tượng gần thời gian thực, đặc biệt quan trọng trong các ứng dụng yêu cầu xử lý nhanh (như giám sát an ninh, xe tự hành).

Hiệu quả bộ nhớ: Mô hình nhẹ hơn so với nhiều thuật toán phát hiện đối tượng khác như YOLOv4 hay Faster R-CNN. Điều này làm giảm yêu cầu về phần cứng.

2. Độ chính xác tốt

Cân bằng giữa tốc độ và độ chính xác: YOLOv5 duy trì độ chính xác cao (mAP - Mean Average Precision) đồng thời đạt tốc độ xử lý nhanh, phù hợp cho cả bài toán phát hiện đối tượng lớn và nhỏ.

Cải thiện phát hiện đối tượng nhỏ: YOLOv5 có kiến trúc phù hợp để xử lý tốt các đối tượng nhỏ hoặc chồng lấn nhau.

3. Dễ sử dụng và triển khai

Dễ dàng tích hợp: YOLOv5 được viết bằng PyTorch, dễ hiểu và dễ tùy chỉnh. Người dùng có thể nhanh chóng tinh chỉnh mô hình hoặc triển khai trong các ứng dụng thực tế.

Hỗ trợ tốt từ cộng đồng: Có tài liệu chi tiết, cộng đồng lớn và nhiều ví dụ mã nguồn mở giúp người dùng học và triển khai dễ dàng.

4. Các cải tiến so với phiên bản trước

Tích hợp các kỹ thuật mới: YOLOv5 sử dụng Mosaic Augmentation, CIoU Loss và các cải tiến trong huấn luyện để tăng hiệu quả và giảm overfitting.

Tự động điều chỉnh hyperparameter: Giúp tối ưu hóa mô hình cho các tập dữ liệu khác nhau.

Kiến trúc tối ưu hóa: Sử dụng cấu trúc CSPDarknet (Cross Stage Partial Network) để cải thiện khả năng trích xuất đặc trưng.

5. Khả năng mở rộng

Hỗ trợ nhiều kích thước mô hình khác nhau (YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x), cho phép người dùng chọn lựa giữa tốc độ và độ chính xác theo yêu cầu.

- Nhược điểm của YOLOv5:

1. Không phải mã chính thức từ tác giả YOLO

YOLOv5 không được tạo ra bởi tác giả ban đầu của dòng YOLO (Joseph Redmon). Đây là phiên bản do cộng đồng phát triển, dẫn đến tranh cãi về tính chính thức và thẩm quyền

2. Giới hạn đối với các ứng dụng phức tạp

Hiệu suất thấp trên đối tượng đặc biệt: Với các tập dữ liệu chứa các đối tượng khó nhận diện, YOLOv5 có thể không đạt hiệu quả bằng các mô hình phức tạp hơn như Faster R-CNN.

Khó xử lý đối tượng dày đặc: Trong các kịch bản mà nhiều đối tượng chồng chéo, YOLOv5 có thể gặp khó khăn trong việc phân tách chính xác.

3. Hạn chế trong mô hình hóa ngữ cảnh

Không xử lý tốt quan hệ giữa các đối tượng: YOLOv5 tập trung vào từng đối tượng riêng lẻ và có thể bỏ qua các thông tin ngữ cảnh giữa các đối tượng.

4. Yêu cầu tinh chỉnh thủ công

Cần điều chỉnh siêu tham số (Hyperparameter): Để đạt hiệu suất tối ưu, người dùng thường cần tinh chỉnh thủ công như learning rate, batch size, và anchor box, điều này có thể gây khó khăn cho người mới bắt đầu.

5. Giới hạn khả năng mở rộng

Mặc dù hỗ trợ nhiều cấu hình mô hình, YOLOv5 có thể không mở rộng tốt như các mô hình Transformer (như DETR - DEtection TRansformer) đối với dữ liệu có cấu trúc phức tạp hoặc kích thước lớn.

**1.4 Ngôn Ngữ và Công nghệ sử dụng**

****

*Hình 1.14 Ngôn ngữ PyThon*

**1.4.1 Ngôn Ngữ Python :**

Vai trò: Python là ngôn ngữ chính, mạnh mẽ và dễ sử dụng, với nhiều thư viện hỗ trợ cho xử lý ảnh và học máy.

Ứng dụng:

* Xử lý ảnh và phát hiện biển số xe (sử dụng OpenCV).
* Nhận diện ký tự quang học (OCR) với Tesseract.
* Phát triển mô hình học máy hoặc deep learning.

 Lý do chọn: Python có cú pháp đơn giản, cộng đồng lớn và thư viện phong phú.

1.4.2 Xử lý ảnh và nhận diện biển số

OpenCV :

* Thư viện xử lý ảnh mạnh mẽ, hỗ trợ phát hiện cạnh, contour và trích xuất vùng biển số từ ảnh.
* Các chức năng sử dụng:
* cv2.Canny() để phát hiện cạnh.
* cv2.findContours() để phát hiện vùng biển số.

Tesseract OCR

* Công cụ nhận diện ký tự quang học mã nguồn mở.
* Sử dụng để nhận diện và chuyển ký tự từ vùng biển số thành văn bản.

1.4.3 Học máy và Deep Learning

TensorFlow hoặc PyTorch :

* Framework mạnh mẽ cho học máy và deep learning.
* Sử dụng để huấn luyện các mô hình phát hiện và nhận diện biển số xe (ví dụ: YOLO, CNN).

Scikit-learn: Thư viện học máy dùng để phát triển các mô hình truyền thống như SVM hoặc Decision Trees (nếu không sử dụng deep learning)

Gradio: là một thư viện Python mạnh mẽ và dễ sử dụng, giúp xây dựng giao diện người dùng (UI) cho các mô hình học máy hoặc bất kỳ ứng dụng Python nào một cách nhanh chóng. Có thể sử dụng Gradio để tạo các ứng dụng web tương tác, cho phép người dùng tải lên tệp, nhập văn bản, hoặc tương tác với các tính năng khác.

**Chương 2 : Xây Dựng Hệ Thống Nhận Dạng**

## 2.1 Yêu cầu bài toán

Mô tả bài toán:

Bài toán đặt ra là xây dựng một hệ thống có khả năng tự động nhận diện biển số xe từ hình ảnh hoặc video đầu vào. Hệ thống phải thực hiện các tác vụ chính sau:

* Xác định vị trí biển số xe trong hình ảnh/video.
* Trích xuất và phân đoạn ký tự trên biển số.
* Nhận diện nội dung ký tự trên biển số (số và chữ).
* Hiển thị hoặc lưu trữ kết quả nhận diện.

2.1.1 Dữ liệu đầu vào:

Hình ảnh: Các bức ảnh chụp biển số xe, có thể đến từ các camera giám sát giao thông, camera bãi đỗ xe, hoặc từ các video clip.

Video: Các đoạn video có chứa xe đang di chuyển, cần phát hiện và nhận diện biển số trong từng khung hình.

Điều kiện đầu vào:

Biển số xe có thể xuất hiện ở các góc độ khác nhau, với độ phân giải thấp hoặc ảnh có nhiễu.

Ánh sáng không đồng đều hoặc có thể có các vật thể che khuất biển số.

2.1.2 Dữ liệu đầu ra:

Biển số nhận diện: Kết quả là biển số xe được trích xuất từ hình ảnh, bao gồm các chữ cái và số trên biển số xe.

Vị trí biển số: Các tọa độ của biển số trong hình ảnh (bounding box) hoặc video để dễ dàng hiển thị kết quả nhận diện.

Giao diện người dùng: Hiển thị biển số nhận diện được lên màn hình hoặc lưu kết quả vào tệp, cơ sở dữ liệu.

2.1.3. Các bước yêu cầu trong hệ thống:

Bước 1: Tiền xử lý dữ liệu

Xử lý ảnh:

* Chuyển đổi ảnh sang ảnh xám (grayscale) hoặc ảnh nhị phân (binary) để giảm độ phức tạp.
* Cân bằng ánh sáng và tăng cường độ tương phản nếu cần thiết.
* Loại bỏ nhiễu bằng các bộ lọc (Gaussian, Median).

Chuẩn bị dữ liệu đầu vào:

* Chuyển đổi và điều chỉnh kích thước hình ảnh cho phù hợp với các mô hình nhận diện.

Bước 2: Phát hiện biển số xe

Phát hiện đối tượng:

* Sử dụng các phương pháp như Haar Cascade Classifier, YOLO, hoặc HOG + SVM để phát hiện và xác định vùng biển số xe trong ảnh.
* Tìm kiếm các đặc điểm đặc trưng của biển số (hình chữ nhật, tỷ lệ khung hình phù hợp, màu sắc nổi bật).

Cắt biển số xe:

* Trích xuất vùng chứa biển số trong ảnh sau khi phát hiện đối tượng.

Bước 3: Phân đoạn ký tự trên biển số

Phân đoạn ký tự:

* Dùng các kỹ thuật phân đoạn (image segmentation) để tách các ký tự trên biển số, đảm bảo không bị nhầm lẫn giữa các ký tự.
* Cắt từng ký tự riêng biệt trong biển số để chuẩn bị cho bước nhận diện.

Bước 4: Nhận diện ký tự

* Sử dụng OCR (Optical Character Recognition):
* Tích hợp Tesseract OCR để nhận diện các ký tự chữ và số trên biển số xe.
* Có thể sử dụng CNN (Convolutional Neural Networks) nếu cần cải thiện độ chính xác.

Bước 5: Hiển thị và lưu trữ kết quả

* Hiển thị kết quả nhận diện:
* Hiển thị biển số nhận diện trên giao diện người dùng (GUI) hoặc hiển thị trực tiếp trên video.
* Vẽ hộp chứa biển số (bounding box) và hiển thị thông tin biển số lên hình ảnh/video.

Lưu trữ kết quả:

* Lưu kết quả nhận diện vào cơ sở dữ liệu hoặc tệp văn bản để phục vụ cho các ứng dụng giám sát hoặc quản lý bãi đỗ xe.

4. Yêu cầu về công nghệ và công cụ:

Ngôn ngữ lập trình:

* Python: Lựa chọn phổ biến cho việc phát triển hệ thống xử lý ảnh và nhận diện đối tượng.

Thư viện sử dụng:

* OpenCV: Dùng để tiền xử lý ảnh, phát hiện đối tượng và phân đoạn hình ảnh.
* Tesseract OCR: Để nhận diện ký tự chữ và số trên biển số.
* TensorFlow/PyTorch (nếu cần): Để triển khai mô hình học sâu cho việc nhận diện ký tự chính xác hơn.
* NumPy, Matplotlib: Hỗ trợ xử lý và hiển thị dữ liệu hình ảnh.

5. Đánh giá hệ thống:

* Độ chính xác (Accuracy): Đo lường tỷ lệ đúng/ sai của kết quả nhận diện biển số.
* Tốc độ xử lý: Đánh giá thời gian xử lý mỗi hình ảnh/video để đảm bảo phù hợp với yêu cầu thời gian thực.
* Khả năng chịu lỗi: Kiểm tra khả năng hoạt động trong các điều kiện ảnh phức tạp, như ảnh có nhiễu, ánh sáng yếu, hoặc khi biển số xe bị che khuất một phần.

**2.2 Xây dựng hệ thống nhận diện biển số xe**

**2.2.1 Khái niệm biển số xe**

Tại Việt Nam, biển kiểm soát xe cơ giới (thường được gọi tắt là biển số xe) là một tấm biển đặc biệt gắn trên mỗi phương tiện cơ giới. Đây là minh chứng pháp lý do cơ quan công an cấp (hoặc Bộ Quốc phòng đối với xe quân sự), thường được cấp khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được chế tác từ hợp kim nhôm-sắt, mang thiết kế hình chữ nhật hoặc gần vuông. Trên đó khắc in các ký tự gồm số và chữ, giúp phân định vùng, địa phương quản lý cũng như thông tin về chủ sở hữu. Đặc biệt, trên biển số còn dập nổi hình Quốc huy Việt Nam, tăng tính nhận diện và bảo mật.

**Quy định về chữ và ký hiệu**

* Biển xe dân sự không sử dụng các chữ cái I, J, O, Q, W, trong khi chữ R chỉ dành riêng cho xe rơ-moóc hoặc sơ-mi rơ-moóc.
* Khi tra cứu, biển số cung cấp thông tin chi tiết về thời gian mua xe, danh tính người sở hữu hoặc đơn vị sử dụng, hỗ trợ đắc lực cho công tác an ninh và quản lý phương tiện.

**Tiêu chuẩn kích thước biển số**

Ở mỗi quốc gia, biển số xe thường tuân theo quy chuẩn riêng về kích thước. Tại Việt Nam, tỉ lệ kích thước của các biển số được giữ gần như đồng nhất. Biển số xe có hai loại chính với kích thước cụ thể:

* Loại biển dài: Chiều cao 110 mm, chiều dài 470 mm.
* Loại biển ngắn: Chiều cao 200 mm, chiều dài 280 mm.
* Dựa trên đó, tỉ lệ chiều cao/rộng được quy định như sau:
* Biển một hàng: 3.5≤cao/rộng≤6.53.5 \leq \text{cao/rộng} \leq 6.53.5≤cao/rộng≤6.5
* Biển hai hàng: 0.8≤cao/rộng≤1.50.8 \leq \text{cao/rộng} \leq 1.50.8≤cao/rộng≤1.5

**Quy cách chữ và số**

* Số lượng ký tự: từ 7 đến 9.
* Chiều cao của ký tự (chữ và số): 80 mm.
* Chiều rộng của ký tự: 40 mm.

**Ý nghĩa ứng dụng**

Với các tiêu chuẩn này, chúng ta có thể thiết lập các thông số, công cụ lọc hoặc điều chỉnh nhằm nhận diện và quản lý phương tiện phù hợp với yêu cầu. Việc đồng bộ hóa kích thước và thông tin trên biển số xe không chỉ đảm bảo tính thẩm mỹ mà còn hỗ trợ hiệu quả cho hệ thống an ninh, quản lý giao thông và bảo vệ quyền lợi của người sở hữu.

## 2.2.2 Xử lý ảnh và Open CV

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,....

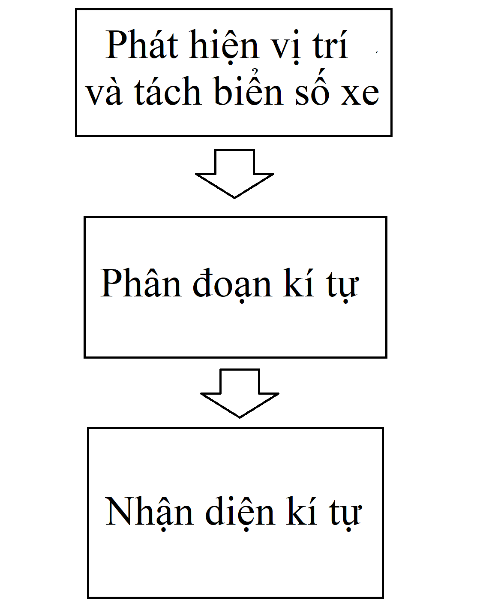
OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Opencv có rất nhiều ứng dụng như:

* Nhận dạng ảnh
* Xử lý hình ảnh
* Phục hồi hình ảnh/video
* Thực tế ảo
* Các ứng dụng khác

## Hướng giải quyết

Hiện nay trên thế giới đã có rất nhiều cách tiếp cận khác nhau với việc nhận dạng biển số xe, tuy nhiên trong phạm vi đồ án này em sẽ giải quyết vấn đề theo 3 bước chính:

* 1. Phát hiện vị trí và tách biển số xe từ một hình ảnh có sẵn từ đầu vào là camera
  2. Phân đoạn các kí tự có trong biển số xe
  3. Nhận diện các kí tự đó rồi đưa về mã ASCII



Hình 2.3 - 1 Các bước chính trong nhận dạng biển số xe

# 2.2.3 Phát hiện vị trí và tách biển số

Để nhận diện được biển số xe đầu tiên chúng ta phải xác định được vị trí của biển số xe, trong tác vụ này mô hình YOLO là lựa chọn hàng đầu.

1. **Lý thuyết mô hình YOLO**

Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers (Conv) và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers (FC) sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh.

Điểm mạnh của mô hình YOLO

* **Hiệu suất cao**: YOLO tối ưu hóa cả về tốc độ lẫn độ chính xác, phù hợp với các ứng dụng thời gian thực.
* **Cấu trúc nhẹ nhàng**: Mô hình được thiết kế nhỏ gọn, cho phép triển khai trên các thiết bị có tài nguyên hạn chế như điện thoại di động, IoT, hoặc edge devices.

1. **Huấn luyện mô hình** 
   1. **Dữ liệu huấn luyện**

**A collage of images of motorcycles

Description automatically generated**

Hình ảnh : Bộ dữ liệu

Về dữ liệu huấn luyện nhóm sử dụng bộ dữ liệu từ MIAI , một trang nổi tiếng về AI gồm 1750 tệp hình ảnh đã được gán nhãn vị trí.

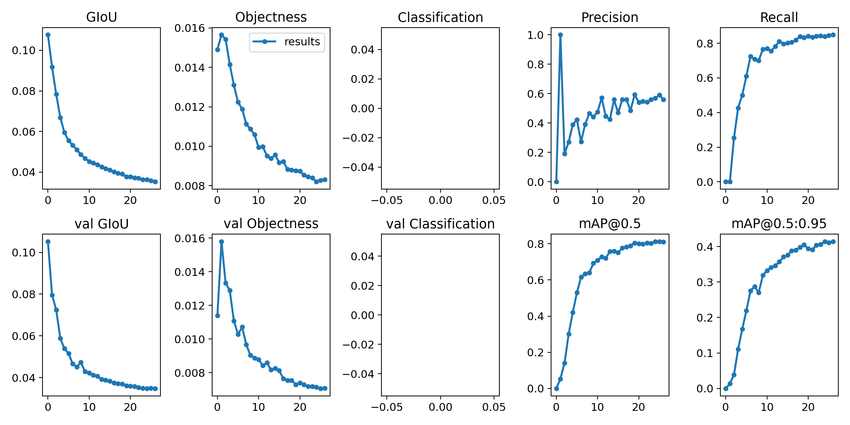
* 1. **Trainning mô hình**

Để huấn luyện mô hình nhóm sử dụng GPU của Colab để huấn luyện với 800 vòng sau khi huấn luyện thành công sai số trên tâp đánh giá là 0.62 và đã không thể giảm được nữa, nhóm dừng quá trình huấn luyện.

A graph of a graph showing the loss of ephs

Description automatically generated with medium confidence

Kết quả mô hình huấn luyện

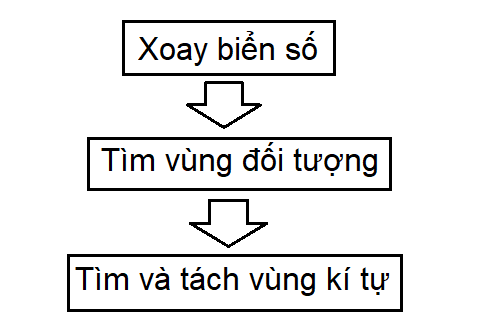


Nhìn vào kết quả đánh giá ta có thể thấy mô hình đạt giá trị mAP cao 0.8/1 và các sai số GioU đã hội tụ không thể giảm được nữa , mô hình hoàn toàn có thể chạy tốt trong thực tế

# 2.2.4 Phân đoạn ký tự

## Hướng giải quyết

Ở giai đoạn này có những bước chính sau: Xoay biển số để tăng khả năng nhận diện, Tìm tất cả các vùng kín cho là kí tự và lọc ra những kí tự đúng. Tách hình ảnh nhưng kí tự đó ra và đưa vào bộ nhận diện



Hình 4.1 - 1 Các bước chính trong phân đoạn kí tự

## Xoay biển số

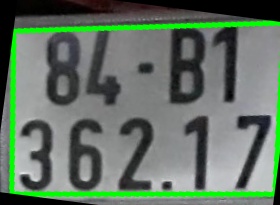
Khi chụp ảnh đầu vào, không phải lúc nào biển số cũng ở chính diện, có thể bị méo sang trái, sang phải, nghiêng góc dẫn đến nếu cứ sử dụng ảnh biển số đã cắt mà không điều chỉnh góc độ dẫn đến ảnh kí tự được cắt ra đưa vào bộ nhận diện rất dễ bị sai. Ví dụ giữa số 1 và số 7, số 2 và chữ Z, chữ B và số 8,...



Hình 4.2 - 1 Ảnh biển số chưa xoay

Phương pháp xoay ảnh em sử dụng ở đây là:

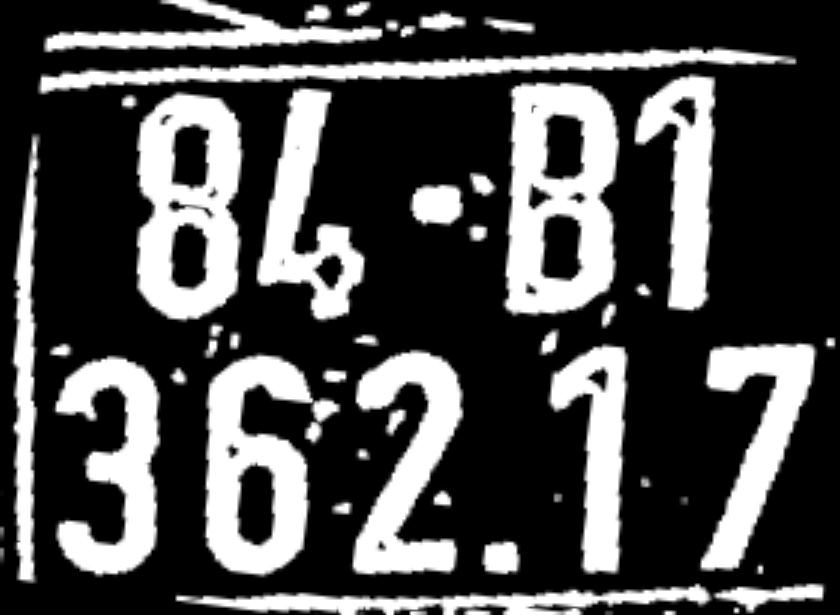
1. Lọc ra tọa độ 2 đỉnh A,B nằm dưới cùng của biển số
2. Từ 2 đỉnh có tọa độ lần lượt là A(x1, y1) và B(x2,y2) ta có thể tính được cạnh đối và cạnh kề của tam giác ABC
3. Ta tính được góc quay
4. Xoay ảnh theo góc quay đã tính. Nếu ngược lại điểm A nằm cao hơn điểm B ta cho góc quay âm



Hình 4.2 - 2 Ảnh biển số đã xoay

## Tìm vùng đối tượng

Từ ảnh nhị phân, ta lại tìm contour cho các kí tự (phần màu trắng). Sau đó vẽ những hình chữ nhật bao quanh các kí tự đó. Tuy nhiên việc tìm contour này cũng bị nhiễu dẫn đến việc máy xử lý sai mà tìm ra những hình ảnh không phải kí tự. Ta sẽ áp dụng các đặc điểm về tỉ lệ chiều cao/rộng của kí tự, diện tích của kí tự so với biển số



Hình 4.3 - 1 Ảnh nhị phân



Hình 4.3 - 2 Tìm vùng đối tượng

Trong ảnh 4.3 - 2 những đường màu vàng là đường contour và nếu so sánh với ảnh nhị phân 4.3 -1 thì có rất nhiều đường nhiễu như đường viền biển số, dấu gạch, dấu chấm... Sau khi đã áp dụng các điều kiện thì sẽ vẽ ra những hình chữ nhật màu xanh bao quanh các kí tự.

## 2.2.5 Tìm và tách kí tự

Sau khi đã nhận dạng từng kí tự bằng hình chữ nhật và cũng đã có tọa độ vị trí 4 đỉnh của hình đó, ta lúc này có thể cắt hình ảnh kí tự đó ra phục vụ cho giai đoạn sau “Nhận diện kí tự”. Lưu ý ở đây ta cắt ảnh nhị phân chứ không cắt từ ảnh gốc.



Hình 4.4 - 1 Ảnh kí tự sau khi cắt

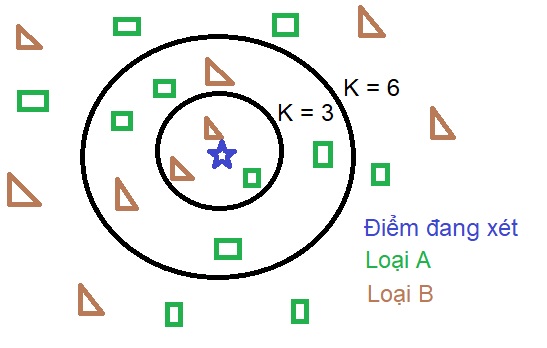
**2.2.5 Nhận diện ký tự**

### Thuật toán KNN (K - Nearest Neighbor)

KNN là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning, có thể sử dụng cho cả bài toán phân loại và hồi quy. Về ý tưởng là gán kết quả với dữ liệu training gần giống với mẫu nhất. Ví dụ như khi đi câu cá, ta không biết cá câu lên là cá rô hay cá chép, nhưng khi so sánh những đặc điểm về mắt, mang, vây,... từ những con cá rô, cá chép đã thấy thì cuối cùng có thể quyết định xem con cá mình câu được thuộc nhóm cá nào.

KNN hoạt động theo quy trình gồm 4 bước chính:

1. Xác định tham số K (số láng giềng gần nhất).
2. Tính khoảng cách từ điểm đang xét đến tất cả các điểm trong tập dữ liệu cho trước
3. Sắp xếp các khoảng cách đó theo thứ tự tăng dần
4. Xét trong tập K điểm gần nhất với điểm đang xét, nếu số lượng điểm của loại nào cao hơn thì coi như điểm đang xét thuộc loại đó



Hình 5.1 - 3 Ví dụ về KNN

Việc điểm đang xét thuộc loại nào còn phụ thuộc vào hệ số K hay trọng số khoảng cách... mà người dùng đặt sao cho phù hợp với bài toán đang xét. Chẳng hạn ở hình trên nếu ta xét K = 3 thì điểm đang xét sẽ thuộc loại B, ngược lại nếu K = 6 thì nó thuộc loại A. Ngoài ra người ta có thể để trọng số cao hơn cho những điểm gần hơn hay ít khi sử dụng K = 1 để đảm bảo kết quả đầu ra được tối ưu.

Thông thường việc tính khoảng cách đến các điểm sẽ theo công thức Euclid:

Tuy nhiên phương pháp này có một số ưu và nhược điểm như sau:

* Ưu điểm:
  + Dễ sử dụng và cài đặt
  + Độ phức tạp tính toán nhỏ
  + Việc dự đoán kết quả rất đơn giản
* Nhược điểm:
  + Với K nhỏ, khi gặp nhiễu dễ đưa ra kết quả không chính xác
  + Cần nhiều thời gian lưu trainning set và nếu test tăng lên sẽ tốn rất nhiều thời gian

## Hướng giải quyết

Ở giai đoạn cuối này được thực hiện theo những bước sau:

1. Tạo tập dữ liệu để huấn luyện
2. Huấn luyện mô hình KNN
3. Đưa hình ảnh từ bước “Phân đoạn kí tự” vào mô hình KNN đã tạo để đưa ra kết quả
4. In ra kết quả biển số

Bước 1 và 2 ta sẽ tạo ra mô hình KNN riêng biệt với code chính. Để khi cần nhận diện kí tự ta không cần phải làm lại các bước từ đầu. Đầu tiên em tạo tập dữ liệu (tập hình ảnh của các chữ số và kí tự) để train từ phần mềm paint. Trong phần mềm Paint ta viết các chữ số và kí tự (trừ kí tự O, I, J) với phông chữ “Biển số xe Việt Nam”, có thể xoay các kí tự này lần lượt với các góc . Kết quả có dạng như sau:



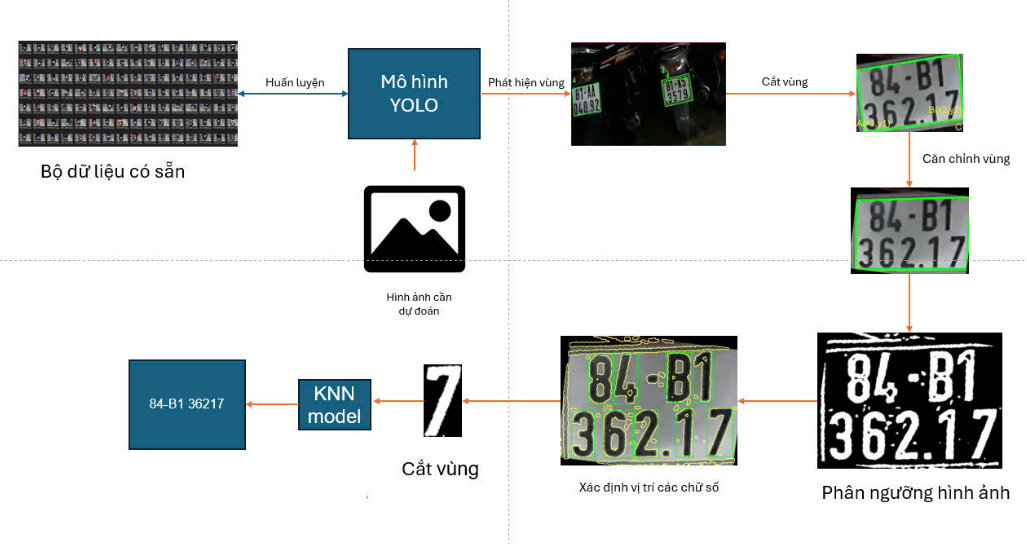
Hình 5.2 - 1 Tập dữ liệu huấn luyện

Tiếp theo ta lấy ngưỡng, vẽ contour và cắt từng kí tự. Vì mỗi kí tự có kích thước khác nhau xử lý phức tạp nên cần chuẩn hóa hình ảnh lại với kích thước cao: rộng là 30:20 pixels. Thay vì mỗi kí tự đưa vào mô hình để máy nhận diện thì những kí tự này sẽ được ta gắn nhãn bằng những phím bấm trên máy tính. Sau khi gắn nhãn hết các kí tự ta sẽ lưu hai file .txt là classifications.txt và flattened\_images.txt. File classifications.txt có nhiệm vụ lưu các mã ASCII của các kí tự đó và file flattened\_images.txt sẽ lưu giá trị các điểm ảnh có trong hình ảnh kí tự (hình 20x30 pixel có tổng cộng 600 điểm ảnh có giá trị 0 hoặc 255)

Bước 3 và 4. Ta thực hiện đưa ảnh đang xét vào và tính khoảng cách đến tất cả các điểm trong mẫu, kết quả sẽ là mã ASCII đại điện cho hình ảnh đó. Cuối cùng ta in biển số xe ra hình. Tuy nhiên ở Việt Nam có hai loại biển số là biển một hàng và biển hai hàng. Về ý tưởng chung để phân biệt hai hàng này ta dựa vào vị trí của hình ảnh kí tự, nếu vị trí nằm thấp 1/3 chiều cao của biển số thì kí tự sẽ được xếp vào hàng một. Ngược lại sẽ được xếp vào hàng hai.

|  |  |
| --- | --- |
| Hình 5.2 - 2 Biển số trước khi nhận diện | Hình 5.2 - 3 Biển số sau khi nhận diện |

* Hình ảnh minh họa hệ thống nhận dạng :



**CHƯƠNG 3 : KẾT QUẢ THỰC NGHIỆM**

**3.1 Dữ liệu huấn luyện**

**a. Dữ liệu huấn luyện mô hình nhận vị trí**

Để huấn luyện mô hình YOLOv5 nhận diện vị trí nhóm sử dụng dữ liệu từ MÌ AI – Nhà cung cấp dữ liệu miễn phí phục vụ cho mục đích nghiên cứu.

Dữ liệu gồm 1750 tệp hình ảnh đã được gán nhãn vị trí.

**A collage of images of motorcycles

Description automatically generated**

Hình: Bộ dữ liệu nhận diện biển số xe

**b. Dữ liệu huấn luyện mô hình nhận diện ký tự**

Nhóm sử dụng bộ dữ liệu từ Github của tác giả winter2897 với dữ liệu là các hình biển số xe đã được cắt chính xác và gán nhãn sẵn phục vụ cho mục đích huấn luyện sau này.

**A screenshot of a computer

Description automatically generated** **Kết quả huấn luyện mô hình**

Kết quả thực nghiệm mô hình phát hiện vị trí biển số xe được miêu tả ở bảng dưới đây:

|  |  |  |  |
| --- | --- | --- | --- |
| Loại biển | Tổng biển số | Số biển  tìm thấy | Tỉ lệ tìm thấy biển số xe (%) |
| Biển 1 hàng | 370 | 182 | 49,2% |
| Biển 2 hàng | 2349 | 924 | 39,3% |

Bảng 1.1 Tỉ lệ tìm thấy biển số xe trong hình

Ở biển số 1 hàng có tỉ lệ nhận diện cao hơn vì tập mẫu còn ít và mỗi frame ảnh chỉ có 1 biển, dẫn đến biển số thường nằm ở vị trí trung tâm nên dễ nhận diện hơn. Ngược lại, với biển 2 hàng có tập mẫu lớn, và số biển số trong 1 frame nhiều hơn. Chưa kể, biển số ở đây được tính là vùng hình bình hành có tối thiểu 7 kí tự mới được tính là biển số, dẫn đến có nhiều biển số đã cắt đúng những vẫn không được tính vào. Tuy nhiên cách làm này lại giúp loại bỏ đáng kể những chi tiết, biển số sai lệch từ môi trường ngoài.

|  |  |
| --- | --- |
| Hình 3.3.2 Không tìm thấy kí tự | Hình 3.3.1 Ảnh gốc |
| Hình 3.3. 3 Tìm thấy 5 kí tự |
| Hình 3.3. 4 Tìm thấy 9 kí tự |

Ở hình 3.3 .1 ta thấy chương trình xử lý tìm ra 3 vùng được cho là biển số, tuy nhiên số kí tự tìm ra ở hình 3.3.2 và 3.3.3 không đủ nên sẽ bị loại dẫn đến trong ảnh chỉ còn lại duy nhất biển số hình 3.3. 4.

Khi ta quay theo nhiều góc độ, nhiều vị trí dẫn đến khi tính toán diện tích, tỉ lệ cao/rộng của biển số không còn thỏa điều kiện đặt ra nên đã bị loại. Biển số có thể bị ảnh hưởng bởi những chi tiết ngoài nên khi xấp xỉ contour không ra hình tứ giác, dẫn đến cũng gây mất biển số. Lỗi này đặc biệt xảy ra ở những xe ô tô vì ô tô thường có nền xung quanh biển số là những vật liệu phản chiếu ánh sáng mạnh, gây ảnh hưởng lớn đến quá trình xác định vùng biển số.

|  |  |
| --- | --- |
| Hình 3.3. 5 Lấy ngưỡng cao/rộng 1.5 | Hình 3.3. 6 Lấy ngưỡng cao/rộng 1.4 |

Dưới đây ta xét khả năng khoanh vùng và nhận diện kí tự tương ứng với giai đoạn “Phân đoạn kí tự” và “Nhận diện kí tự” đã đặt ra ở đầu bài toán.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Loại biển | Số biển tìm thấy | Không sai | Sai 1 kí tự | Sai 2 kí tự | Sai 3 kí tự  trở lên |
| Biển 1 hàng | 182 | 61 | 88 | 19 | 14 |
| Tỉ lệ (%) | | 33,5 | 48,4 | 10,4 | 7,7 |

Bảng 1.2 Tỉ lệ nhận diện sai kí tự ở biển 1 hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Loại biển | Số biển tìm thấy | Không sai | Sai 1 kí tự | Sai 2 kí tự | Sai 3 kí tự  trở lên |
| Biển 2 hàng | 924 | 286 | 273 | 175 | 190 |
| Tỉ lệ nhận diện (%) | | 31 | 29,5 | 18,9 | 20,6 |

Bảng 1.3 Tỉ lệ nhận diện sai kí tự ở biển 2 hàng

Nhìn chung mô hình nhận diện KNN cũng khá tốt, có những kí tự dù bị mờ, bị nghiêng vẫn nhận diện đúng. Điều này một phần nhờ vào chương trình đã xoay biển số lại cho để tăng khả năng nhận diện, cho dù nghiêng thì kí tự cũng chỉ nghiêng từ đến .Tuy nhiên vẫn còn nhầm lẫn nhiều giữa các kí tự như số 1 với số 7. Chữ G, chữ D, số 6 với số 0. Chữ B với số 8...



Hình 3.4. 1Ảnh gốc nhận diện 3 biển số

|  |  |  |
| --- | --- | --- |
| Hình 3.4 .2Biển số 1 | Hình 3.4.3 Biển số 2 | Hình 3.4. 4 Biển số 3 |

Theo như hình gốc 3.5 - 1 ta nhận diện đúng hết biển số 2 và 3. Còn biển số 1 bị sai giữa chữ B và số 8



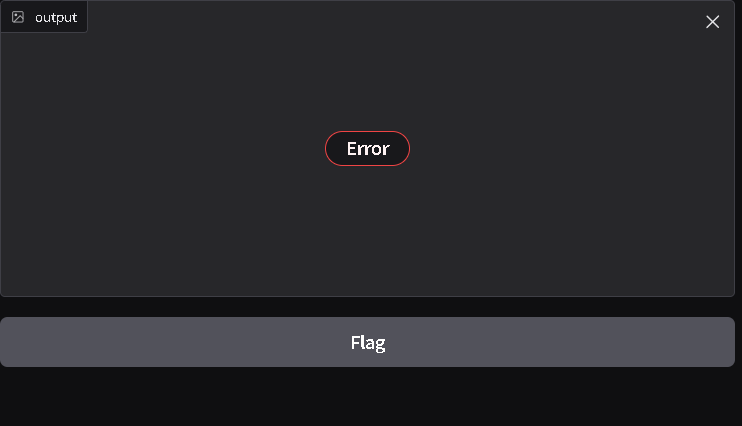
Hình 3.4 .5 Không khoanh được vùng kí tự

Ở ảnh 3.5 - 4 chỗ số 0 bị dính con ốc dẫn đến không thể khoanh vùng được, Tuy nhiên ở chữ F mặc dù vẫn dính con ốc trong ảnh cắt ra để nhận diện, mô hình KNN vẫn cho ra đáp án đúng.

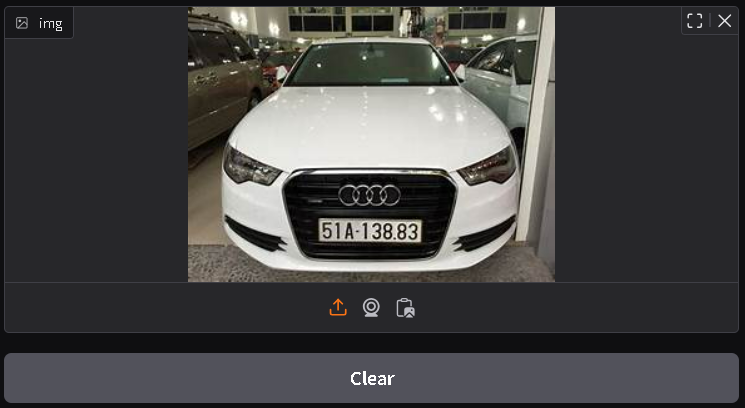
**3.2 Kết quả thực nghiệm**

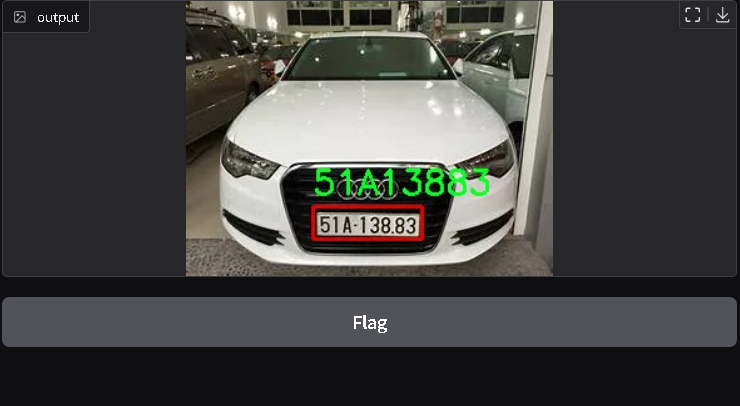
* **Giao diện Input và Output của Hệ thống :**

****

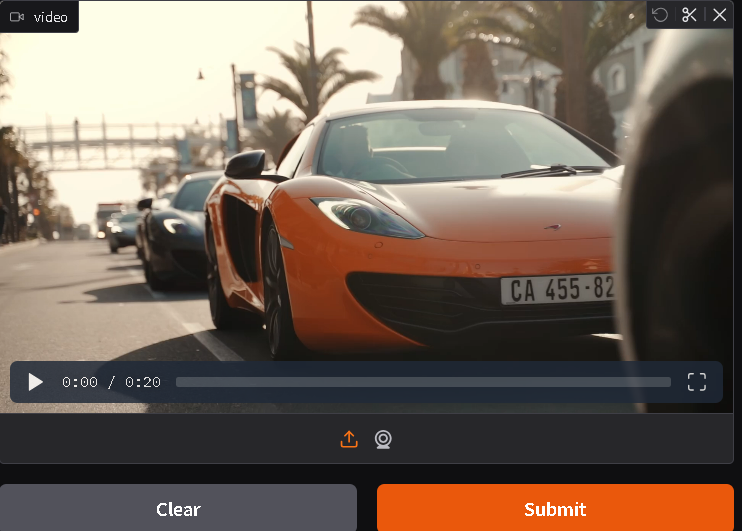


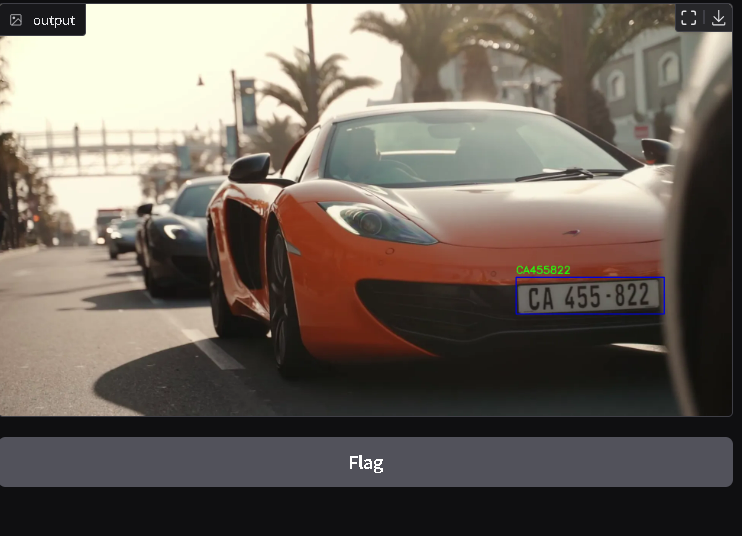
* Ảnh được nhận dạng :





* **Nhận dạng với video :**

****

****

* **Nhận dạng với webcam :**





# Kết luận

**Kết quả đã đạt được:**

Trong quá trình xây dựng hệ thống nhận diện biển số xe tự động, các kết quả sau đây đã được hoàn thành:

* Tiền xử lý ảnh và phát hiện biển số xe:
* Hệ thống đã áp dụng các phương pháp tiền xử lý ảnh như chuyển đổi sang ảnh xám, cân bằng ánh sáng và loại bỏ nhiễu, giúp cải thiện chất lượng ảnh đầu vào.
* Sử dụng các thuật toán phát hiện đối tượng như YOLO và Haar Cascade để phát hiện và xác định vùng biển số trong hình ảnh với độ chính xác cao.

Tách và nhận diện ký tự:

* Các phương pháp phân đoạn ký tự đã được áp dụng thành công, giúp tách các ký tự trên biển số một cách rõ ràng và dễ dàng cho quá trình nhận diện.
* Tích hợp Tesseract OCR hoặc mô hình học sâu để nhận diện ký tự trên biển số, đạt được kết quả tương đối chính xác trong các trường hợp biển số rõ ràng và có độ phân giải tốt.
* Tốc độ xử lý và độ chính xác:
* Hệ thống có thể xử lý hình ảnh trong thời gian thực (real-time) với tốc độ ổn định, đáp ứng yêu cầu của các ứng dụng giám sát giao thông và quản lý bãi đỗ xe.

**Chưa đạt được:**

Mặc dù đạt được những kết quả quan trọng, nhưng hệ thống vẫn chưa hoàn toàn hoàn thiện và còn một số hạn chế:

* Khả năng nhận diện biển số trong điều kiện phức tạp:
* Hệ thống gặp khó khăn khi nhận diện biển số trong điều kiện ánh sáng yếu, biển số bị che khuất, hoặc khi biển số có góc nghiêng lớn. Tốc độ xử lý và độ chính xác có thể giảm trong những tình huống này.
* Độ chính xác khi biển số bị nhiễu hoặc chất lượng ảnh kém:
* Khi biển số bị nhiễu, mờ hoặc có độ phân giải thấp, việc nhận diện ký tự gặp khó khăn. Các thuật toán nhận diện hiện tại vẫn chưa hoàn toàn chính xác trong các tình huống này.
* Khả năng nhận diện trên nhiều kiểu biển số khác nhau:
* Hệ thống chưa hoàn toàn linh hoạt khi áp dụng vào nhiều loại biển số khác nhau, đặc biệt là những biển số có font chữ hoặc màu sắc không phổ biến.

**Hướng phát triển:**

Để nâng cao hiệu quả và khả năng ứng dụng của hệ thống nhận diện biển số xe tự động, một số hướng phát triển sau có thể được thực hiện:

* Cải tiến khả năng nhận diện trong điều kiện khó khăn:
* Nâng cao độ chính xác trong việc xử lý ảnh có độ phân giải thấp thông qua các mô hình học sâu tiên tiến và tăng cường dữ liệu.
* Tăng cường khả năng nhận diện với nhiều kiểu biển số:
* Phát triển và huấn luyện các mô hình nhận diện biển số đa dạng, có khả năng nhận diện biển số từ nhiều quốc gia, loại biển số, và font chữ khác nhau.
* Xây dựng một bộ dữ liệu đa dạng hơn, bao gồm nhiều kiểu biển số và điều kiện môi trường khác nhau để huấn luyện mô hình học sâu.
* Tối ưu hóa tốc độ và hiệu suất:
* Tối ưu hóa thuật toán và mô hình học sâu để cải thiện tốc độ xử lý, đảm bảo hệ thống hoạt động hiệu quả trong môi trường thực tế, đặc biệt khi xử lý video hoặc nhiều khung hình trong thời gian thực.

Tích hợp với các hệ thống khác:

* Xây dựng các ứng dụng tích hợp với hệ thống giám sát giao thông, hệ thống thu phí tự động, hoặc các hệ thống quản lý bãi đỗ xe thông minh, giúp tối ưu hóa quá trình vận hành và nâng cao hiệu quả công việc.

Cải tiến giao diện người dùng:

* Phát triển giao diện người dùng trực quan và dễ sử dụng, hỗ trợ hiển thị và lưu trữ kết quả nhận diện biển số một cách thuận tiện và nhanh chóng.

# Danh mục tham khảo

1. Trí tuệ nhân tạo và học sâu trong thị giác máy tính, Ian Goodfellow, Yoshua Bengio, và Aaron Courville - "Deep Learning" (2016)

2. Backpropagation theory, Rumelhart, Hinton, and Williams (1986)

3. The History of Computer Vision, Tsai and Lee (2008)

4. Object Detection & Recognition: Theories and Implementations, Sinno Jialin Pan (2015)

5. Deep Learning for Computer Vision, Adrian Rosebrock (2020)

6. Machine Learning, Tom M. Mitchell (1997)

7.<https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P>

8. <https://viblo.asia/p/nhan-dien-bien-so-xe-viet-nam-Do754P9L5M6>

9. https://www.carmudi.vn/blog-xe-hoi/bien-so-xe-cac-tinh/