**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc529839839)

[Chương 1 QUÁ TRÌNH THU NHẬN ẢNH VÀ VIDEO 5](#_Toc529839840)

[1.1 Tổng quan về xử lý ảnh 5](#_Toc529839841)

[1.2 Các quá trình xử lý ảnh 6](#_Toc529839842)

[1.3 Thiết bị thu nhận ảnh 7](#_Toc529839843)

[1.4 Lấy mẫu và lượng tử hóa 7](#_Toc529839844)

[1.4.1 Lấy mẫu 8](#_Toc529839845)

[1.4.2 Lượng tử hóa 8](#_Toc529839846)

[1.5 Biểu diễn ảnh 9](#_Toc529839847)

[1.6 Một số định dạng ảnh phổ biến 11](#_Toc529839848)

[1.6.1 File ảnh IMG 11](#_Toc529839849)

[1.6.2 File ảnh BMP 12](#_Toc529839850)

[1.6.3 File ảnh PNG 17](#_Toc529839851)

[1.6.4 File ảnh JPEG 18](#_Toc529839852)

[1.7 Kết thúc chương 1 19](#_Toc529839853)

[Chương 2 BÀI TOÁN PHÂN LOẠI ĐỐI TƯỢNG TRONG ẢNH 20](#_Toc529839854)

[2.1 Bài toán phân loại đối tượng trong ảnh 20](#_Toc529839855)

[2.2 Thuật toán “Viola and Jones” 21](#_Toc529839856)

[2.2.1 Đặc trưng Haar 21](#_Toc529839857)

[2.2.1 Tính giá trị đặc trưng Haar 22](#_Toc529839858)

[2.2.2 Thuật toán Adaboost 25](#_Toc529839859)

[2.2.3 Khai triển thuật toán Adaboost 28](#_Toc529839860)

[2.3 Kết thúc chương 2 30](#_Toc529839861)

[Chương 3 KỸ THUẬT TRACKING 31](#_Toc529839862)

[3.1 Khái niệm Tracking 31](#_Toc529839863)

[3.2 Kỹ thuật Tracking 31](#_Toc529839864)

[3.2.1 Quy trình chung 31](#_Toc529839865)

[3.2.2 Thuật toán centroid based tracking 32](#_Toc529839866)

[3.3 Kết luận chương 3. 34](#_Toc529839867)

[Chương 4 XÂY DỰNG PHẦN MỀM PHÁT HIỆN VÀ PHÂN LOẠI PHƯƠNG TIỆN GIAO THÔNG 35](#_Toc529839868)

[4.1 Bài toán 35](#_Toc529839869)

[4.1.1 Yêu cầu 1 35](#_Toc529839870)

[4.1.2 Yêu cầu 2 35](#_Toc529839871)

[4.2 Phân tích yêu cầu và lựa chọn công cụ 35](#_Toc529839872)

[4.2.1 Phân tích yêu cầu 35](#_Toc529839873)

[4.2.2 Sơ lược về OpenCV 36](#_Toc529839874)

[4.3 Huấn luyện bộ phát hiện 37](#_Toc529839875)

[4.3.1 Thu thập dữ liệu 37](#_Toc529839876)

[4.3.2 Tạo file miêu tả cho mẫu dữ liệu khẳng định và phủ định 39](#_Toc529839877)

[4.3.3 Tạo file có định dạng “OpenCV data vector” từ tệp dữ liệu khẳng định 40](#_Toc529839878)

[4.3.4 Huấn luyện bộ nhận dạng 41](#_Toc529839879)

[4.3.5. Giải thích ý nghĩa của file cascade. 42](#_Toc529839880)

[4.4 Thiết kế giao diện 44](#_Toc529839881)

[4.5 Thiết kế các lớp chính 45](#_Toc529839882)

[4.6 Kiểm thử 45](#_Toc529839883)

[4.6.1 Tính năng phân loại phương tiện giao thông trong ảnh 45](#_Toc529839884)

[4.6.2 Tính năng phân loại và đếm xe máy trong video 49](#_Toc529839885)

[KẾT LUẬN 51](#_Toc529839886)

**DANH MỤC BẢNG**

[*Bảng 1‑1 Giải thích BitmapHeader (54 bytes)* 16](#_Toc529839888)

[*Bảng 4‑1 Kết quả thử nghiệm đếm xe máy* 49](#_Toc529839889)

**DANH MỤC HÌNH VẼ**

[Hình 1.1 Các giai đoạn chính trong xử lý ảnh 6](#_Toc529840950)

[Hình 1.2.a Hình 1.2.b 9](#_Toc529840951)

Hình 1.3.a Quan hệ 8 lân cận….……………………………………………….11

Hình 1.3.b Quan hệ 4 lân cận…………….……..……………………………...11

[Hình 2.1 Đặc trưng cạnh 21](#_Toc529840952)

[Hình 2.2 Đặc trưng đường 22](#_Toc529840953)

[Hình 2.3 Đặc trưng chữ nhật 22](#_Toc529840954)

[Hình 2.4 Minh họa tính đặc trưng Haarlike với đặc trưng 1 x 2 23](#_Toc529840955)

[Hình 2.5 Ví dụ tính Intergral Image 24](#_Toc529840956)

[Hình 2.6 Tính Intergral Image của vùng ảnh D 24](#_Toc529840957)

[Hình 2.7 Mô hình phân tầng kết hợp các bộ phân loại yếu để xác định đối tượng 26](#_Toc529840958)

[Hình 2.8 Kết hợp các bộ phân loại yếu thành bộ phân loại mạnh 27](#_Toc529840959)

[Hình 3.1 Khởi tạo định danh cho các đối tượng cần tracking 32](#_Toc529840960)

[Hình 3.2 Tính giá trị Euclidean của các cặp hình cũ mới 33](#_Toc529840961)

[Hình 4.1 Minh họa dữ liệu mẫu 38](#_Toc529840962)

[Hình 4.2 Ví dụ về file info.txt 39](#_Toc529840963)

[Hình 4.3 Minh họa phần đầu của file cascade. 42](#_Toc529840964)

[Hình 4.4.a Cấu trúc trong file cascade 43](#_Toc529840965)

[Hình 4.4.b Cấu trúc trong file cascade. 43](#_Toc529840966)

[Hình 4.5 Giao diện chương trình 44](#_Toc529840967)

[Hình 4.6 Các lớp chính trong chương trình 45](#_Toc529840968)

[Hình 4.7.a Kết quả phân loại 46](#_Toc529840969)

[Hình 4.7.b Kết quả phân loại 46](#_Toc529840970)

[Hình 4.8.a Các vùng nhận dạng bị trùng lặp 47](#_Toc529840971)

[Hình 4.8.b Các vùng nhận dạng bị trùng lặp 47](#_Toc529840972)

[Hình 4.9.a Loại bỏ trùng lặp 48](#_Toc529840973)

[Hình 4.9.b Loại bỏ trùng lặp 48](#_Toc529840974)

[Hình 4.10 Kết quả đếm xe máy trong video 1 49](#_Toc529840975)

[Hình 4.11 Kết quả đếm xe máy trong video 2 50](#_Toc529840976)

# MỞ ĐẦU

**1.Tính cấp thiết của khóa luận**

Ngày nay với sự phát triển như vũ bảo của công nghệ thông tin. Nó đã đem lại những ứng dụng to lớn trong nhiều lĩnh vực khác nhau. Công nghệ thông tin đã trở thành ngành công nghiệp mũi nhọn của nhiều nước trên thế giới. Sự tồn tại và phát triển của một doanh nghiệp, cơ quan, tổ chức nhà nước...Không thể thiếu sự trợ giúp của máy tính.

Trong việc quản lý, thu nhận và xử lý thông tin với khối lượng ngày càng lớn, nhiều lúc với những phần mềm thủ công không đem lại hiệu quả mong muốn, tốn nhiều công sức và thời gian.

Cùng với đó, CNTT rất hữu ích trong hầu hết các giai đoạn của quá trình thu thập, quản lý nội dung cũng như khai thác và xử lý dữ liệu dưới dạng hình ảnh.

Việc nhận dạng và phát hiện các đối tượng trong ảnh trở nên cấp thiết và có nhiều ứng dụng trong thực tiễn như : tích hợp trong các hệ thống giám sát, cảnh báova chạm giao thông, các hệ thống nhận diện, các hệ thống phân tích ảnh y tế,…

Ngay từ những năm 1960, cụ thể là năm 1964, 1965. Bledsoe, cùng với Helen Chan và Charles Bisson là các nhà nghiên cứu đã tiên phong trong việc nghiêu cứu các ký thuật nhận dạng khuôn mặt con người.

Ngày nay, bài toán nhận dạng, phân loại các đối tượng vật thế đã trở nên cơ bản và có mặt trong một hoặc nhiều các hệ thống lớn. Những thập niên gần đây nhiều nhà nghiên cứu đã phát triển mạnh mẽ các bài toán liên quan đến việc xử lý, nhận dạng và phân loại hình ảnh.

Cùng với sự phát triển của Khoa học Công nghệ, nhu cầu tìm hiểu tri thức của sinh viên ngày càng cao, đặc biệt là sinh viên ngành Công nghệ thông tin. Sinh viên, những người đang ngồi trên ghế nhà trường, luôn mong muốn nhận được những tư vấn, chia sẻ kinh nghiệm, giải đáp thắc mắc và cập nhật thông tin một cách đáng tin cậy. Do đó, em lựa chọn đề tài “***Xây dựng phần mềm phát hiện và phân loại phương tiện giao thông trong ảnh***” làm khóa luận tốt nghiệp của mình.

**2. Cơ sở khoa học và tính thực tiễn của đề tài**

- **Cơ sở khoa học:**

Trên thế giới hiện có rất nhiều các framework, library cũng như các phần mềm hỗ trợ cho mục đích xử lý ảnh số. Có thể kể đến các thư viện hay framework như OpenCV, YOLO ( You only look once) là các thư viện miễn phí và rất linh hoạt hỗ trợ xử lý ảnh có cộng đồng hỗ trợ và phát triển mạnh mẽ. Có thể nói sự phát triển của lĩnh vực thị giác máy là rất mạnh mẽ. Cơ sở của việc phát triển mạnh mẽ đó gắn liền với sự phát triển của khoa học máy tính, toán học. Thị giác máy còn được coi như một nhánh nhỏ của trí tuệ nhân tạo. Sự phát triển của các phương pháp học máy hiện đại có thể tạo ra các thuật toán nhận dạng, phân loại mạnh mẽ và hiệu quả hơn so với những thuật toán cổ điển. Việc nghiên cứu, nắm bắt các nguyên lý cơ bản của các thuật toán máy học nói chung và nhận dạng, phân loại nói riêng đã trở thành hành trang cần thiết cho mọi người làm việc trong lĩnh vực Công nghệ thông tin trong thời đại mới.

- **Tính thực tiễn**:

Không chỉ có ứng dụng trong các ngành nghiên cứu khoa học máy tính nói chung mà thị giác máy còn được ứng dụng trong nhiều lĩnh vực khác nhau. Ví dụ như các phần mềm để biên tập và xử lí hình ảnh như Photoshop, Lightroom,… Hoặc hỗ trợ trong việc giúp các bác sĩ chuẩn đoán tổn thương, dị vật trong cơ thể bệnh nhân. Bên cạnh đó là việc thu thập, phân tích hình ảnh để lập bản đồ số, khảo sát địa hình. Hoặc trong nghiệp vụ giám sát, an ninh. Ứng dụng của xử lý ảnh số là rất tiềm năng và ảnh hưởng đến nhiều lĩnh vực khác nhau.

Hiện nay, trong quá trình công nghiệp hóa, hiện đại hóa đất nước. Việc xây dựng, đảm bảo cơ sở hạ tầng, nhất là cơ sở hạ tầng giao thông là rất quan trọng. Song song và có tầm quan trọng không kém đó là vấn đề quản lý, giữ gìn trật tự an toàn giao thông. Việc giám sát, phát hiện các sai phạm giao thông trong điều kiện đô thị đang phát triển là rất quan trọng.

Việc thu thập các dữ liệu về giao thông không chỉ có ý nghĩa trong việc quản lý trật tự đô thị mà nó còn có ý nghĩa trong việc phân tích dữ liệu để phục vụ các mục đích khác nhau như xây dựng, cải tiến cơ sở hạ tầng giao thông.

Với lượng dữ liệu lớn như vậy. Giải pháp sử dụng phần mềm để phát hiện và phân loại phương tiện giao thông đem lại giá trị thiết thực trong việc xác định một tham số quan trọng trong các bài toán về đô thị đó là số lượng phương tiện giao thông.

**3. Mục tiêu đề tài**

**- Mục tiêu chi tiết:**

Đề tài “Xây dựng phần mềm phát hiện và phân loại phương tiện giao thông trong ảnh” nhằm mục đích phát hiện các đối tượng tham gia giao thông trong ảnh.

Sau đó xếp các đối tượng đó vào các lớp đối tượng tham gia giao thông.

**- Mục tiêu cụ thể:**

Ảnh cần được phân tích được đưa vào phần mềm.Các đối tượng giao thông có trong ảnh sẽ được đóng khung.Các đối tượng giao thông có trong ảnh sẽ được phân loại: Đối tượng xe ô tô. Đối tượng xe 2 bánh.

**4. Nội dung khóa luận**

Khóa luận này bao gồm 4 chương như sau:

- Chương 1: Quá trình thu nhận ảnh và video.

- Chương 2: Bài toán phân loại đối tượng trong ảnh.

- Chương 3: Kỹ thuật Tracking.

- Chương 4: Xây dựng phần mềm phát hiện và phân loại phương tiện giao thông trong ảnh.

Em xin gửi lời cảm ơn chân thành tới toàn thể các thầy cô giáo trong khoa Công nghệ thông tin - Học viện Kỹ thuật Quân sự, những người đã tạo điều kiện, truyền tâm huyết và vốn kiến thức của mình cho em trong suốt những năm học tại mái trường này. Thầy cô đã tạo cho chúng em môi trường và điều kiện thuận lợi nhất để học tập tốt, trang bị cho chúng em những kiến thức quý báu giúp chúng em có thể vững bước trên con đường tương lai.

Em xin chân thành cảm ơn thầy giáo TS. Nguyễn Văn Giang, người đã luôn tận tình dẫn dắt, định hướng và hướng dẫn em hoàn thành tốt đề tài này.

Do kinh nghiệm và kiến thức còn hạn chế, trong quá trình xây dựng hệ thống vẫn còn nhiều thiếu sót. Em kính mong nhận được ý kiến đóng góp của các thầy cô và các bạn để đề tài và kiến thức bản thân được hoàn thiện hơn.

Em xin chân thành cảm ơn.

Sinh viên thực hiện

Đặng Vũ Hiệp

# Chương 1 QUÁ TRÌNH THU NHẬN ẢNH VÀ VIDEO

**1.1 Tổng quan về xử lý ảnh**

Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh. Quá trình này có thể nói là quá trình nhận dạng và phân loại đối tượng trong ảnh. Tùy theo tiêu chí đề ra, ta có thể tìm ra đối tượng khớp hoặc gần khớp với mẫu nhất để thỏa mãn yêu cầu nhận dạng. Hoặc ta có thể tìm các đối tượng, vùng có cùng một đặc trưng nhận dạng. Các đặc trưng này có thể không có một mẫu cụ thể nào để so khớp. Nhưng chúng có chung nhiều đặc điểm và có thể được xếp chung vào một lớp.Hoặc cũng có thể là quá trình biến đổi ảnh ban đầu theo các tiêu chí về máu sắc nhằm thỏa mãn nhu cầu về mặt nghệ thuật của con người.

Cũng như xử lý dữ liệu bằng đồ hoạ, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

* Biến đổi ảnh làm tăng chất lượng ảnh.
* Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,…

Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

* 1. **Các quá trình xử lý ảnh**

****

Hình 1.1 Các giai đoạn chính trong xử lý ảnh

Thu nhận ảnh : Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình xử lý ảnh. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner,v.v… và sau đó các tín hiệu này sẽ được số hóa. Việc lựa chọn các thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tượng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

Số hóa : Ở bước này, các tín hiệu điện tử tương tự được lọc và biến đổi thànhcác tín hiệu số trước khi chuyển qua giai đoạn xử lý, phân tích.

Phân tích ảnh: gồm nhiều công đoạn khác nhau. Trước hết là nâng cao chất lượng ảnh. Do nhiều nguyên nhân khác nhau mà ảnh có thể bị suy biến. Do vậy cần phải tăng cường và khôi phục hình ảnh để làm nổi bật một số đặc tính chính của ảnh. Giai đoạn tiếp theo là phát hiện các đặc tính biên, phân vùng ảnh,…

Nén, lưu trữ, truyền tải, nhận dạng : Giai đoạn cuối cùng trong quá trình, tùy theo mục đích của ứng dụng mà ta có thể lựa chọn các cách xử lý cho ảnh.

Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước phân tích ảnh. Một cách tổng quát thì những chức năng xử lý bao gồm cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay, đối tượng giao thông, …

* 1. **Thiết bị thu nhận ảnh**

Các thiết bị thu nhận ảnh thường bao gồm máy camera được tích hợp chuyển đổi tương tự - số ( Analog to Digital ). Hoặc máy quét (scanner) chuyên dụng.Ngày nay, với sự phát triển bùng nổ của phần cứng, ta có thể dễ dàng bắt gặp các thiết bị thu nhận hình ảnh này ở các thiết bị di động ( smartphone, laptop, webcam,… ) hoặc các máy chụp ảnh chuyên dụng của các hãng thiết bị chụp ảnh. Và hầu hết các camera hiện đại đều có sẵn cả hai tính năng chụp ảnh và quay video.

Tùy vào cấu hình của thiết bị, nhưng phần lớn các thiết bị này có thể cho ảnh trắng – đen với mật độ 400 – 1600 dpi (dot per inch) hoặc ảnh màu. Với ảnh trắng – đen mức màu là 0 hoặc 1. Với ảnh màu, thường mỗi điểm ảnh được lưu trữ trong 3 bytes do đó ta có 28x3 màu ( khoảng 16,7 triệu màu).

* 1. **Lấy mẫu và lượng tử hóa**

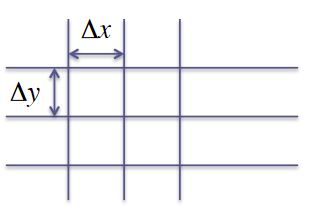
Ảnh trong thực tế là một ảnh liên tục cả về không gian và giá trị độ sáng.

Thời kí đầu để có thể lưu lại các bức hình, người ta dùng các tấm phim để hứng ảnh sau khi đi qua hệ thống thấu kính. Trên các tấm phim này có các chất hóa học phản ứng với ánh sáng từ đó lưu lại ảnh. Đó có thể coi là ảnh tương tự.

Để có thể xử lý ảnh bằng máy tính thì cần thiết phải tiến hành số hóa ảnh tương tự. Cũng chính là số hóa tín hiệu ánh sáng liên tục. Quá trình số hóa là quá trình biến đổi các tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng tử hóa các thành phần giá trị mà về nguyên tắc bằng mắt thường không thể phân biệt được thành các điểm liền kề nhau. Các điểm như vậy được gọi là các pixel (Picture Element) hay các phần tử ảnh hoặc điểm ảnh.

### 1.4.1 Lấy mẫu

Quá trình lấy mẫu có thể hiểu là số hóa tọa độ ảnh. Tức là với một ảnh liên tục. Ta sẽ chia ảnh ra làm nhiều phần khác nhau theo tọa độ x-y. Mỗi một điểm như vậy được qui ước có kích thước ∆x và ∆y tương ứng theo hệ tọa độ tham chiếu.



Hình 1.2 Lấy mẫu ảnh

Gọi F(x,y) là khai triển Fourier của hàm f(x,y) là hàm tọa độ của ánh sáng.

x0 và y0  là các giá trị thỏa mãn F(x,y) = 0 với |x| > x0 và |y| > y0.

x0 và y0  được gọi là dải giới hạn.

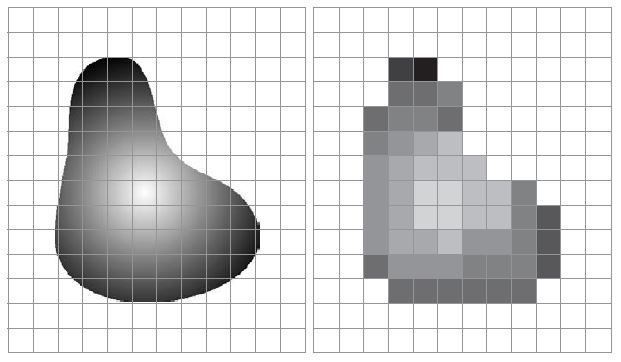
Việc lấy mẫu phải tuân thủ theo tỷ số Nyquist để ảnh có thể khôi phục:

> 2 ξx0 và > 2 ξy0

### 1.4.2 Lượng tử hóa

Quá trình lượng tử hóa có thể hiểu là thực hiện một ánh xạ từ biến u ( biểu diễn giá trị độ sáng liên tục) sang biến u’ với các giá trị thuộc tập hợp hữu hạn.

Kết quả của 2 quá trình này sẽ cho ta một ma trận với các phân tử là các điểm ảnh mang độ sáng khác nhau. Hình 1.2.a và 1.2.b sẽ cho ta cái nhìn khái quát về kết quả của 2 quá trình này.



Hình 1.2.a Hình 1.2.b

Ảnh tương tự Ảnh sau quá trình lấy mẫu và lượng tử hóa

## 1.5 Biểu diễn ảnh

Sau quá trình lấy mẫu và lượng tử hóa, mỗi ảnh là tập hợp các điểm ảnh.Khi được số hóa nó thường được biểu diễn bởi mảng 2 chiều I(n,p): n là dòng và p là cột.

Về mặt toán học có thể xem ảnh là một hàm hai biến f(x,y) với x, y là các biến tọa độ. Giá trị số ở điểm (x,y) tương ứng với giá trị xám hoặc độ sáng của ảnh (x là các cột còn y là các hàng). Giá trị của hàm ảnh f(x,y) được hạn chế trong phạm vi của các số nguyên dương.

0 ≤ f(x,y) ≤ fmax.

Với ảnh đen trắng mức xám của ảnh có thể được biểu diễn bởi một số như sau:

*f = k*

Trong đó SBW() là đặc tính phổ của cảm biến được sử dụng và k là hệ số tỷ lệ xích. Vì sự cảm nhận độ sáng có tầm quan trọng hàng đầu đối với ảnh đen trắng nên SBW()được chọn giống như là hiệu suất sáng tương đối. Vì *f* biểu diễn công suất trên đơn vị diện tích, nên nó bao giờ cũng không âm và hữu hạn.

*0≤ f ≤* fmax.

Trong đó fmax là giá trị lớn nhất mà *f* đạt được. Trong xử lý ảnh, *f* được chia thang sao cho nó nằm trong một phạm vi thuận lợi nào đó.

Thông thường đối với ảnh xám, giá trị fmax là 255 ( 28=256) bởi vì mỗi phần tử

ảnh được mã hóa bởi một byte. Khi quan tâm đến ảnh màu ta có thể mô tả màu qua ba hàm số: thành phần màu đỏ qua *R(x,y)*, thành phần màu lục qua *G(x,y)* và thành phần màu lam qua *B(x,y)*. Bộ ba giá trị *R*, *G*, và *B* nhận được từ:

R *= k*

G *= k*

B *= k*

Ở đó *SR( ),SG( ) và SB( )* theo thứ tự là những đặc tính phổ của các cảm biến (bộ lọc) đỏ, lục và lam. *R*, *G*, *B* cũng không âm và hữu hạn.

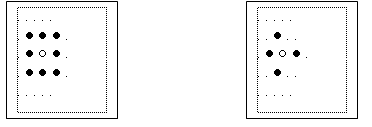
Ảnh có thể được biểu diễn theo một trong hai mô hình: mô hình Vector hoặc mô hình Raster.

- Mô hình Vector: Ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng hiển thị và in ấn, các ảnh biểu diễn theo mô hình vector còn có ưu điểm cho phép dễ dàng lựa chọn, sao chép, di chuyển, tìm kiếm…Theo những yêu cầu này thì kỹ thuật biểu diễn vector tỏ ra ưu việt hơn. Trong mô hình này, người ta sử dụng hướng vector của các điểm ảnh lân cận để mã hóa và tái tạo lại hình ảnh ban đầu. Các ảnh vector được thu nhận trực tiếp từ các thiết bị số hóa như Digitalize hoặc được chuyển đổi từ các ảnh Raster thông qua các chương trình vector hóa.

- Mô hình Raster: là mô hình biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh. Tùy theo nhu cầu thực tế mà mỗi điểm ảnh có thể được biểu diễn bởi một hay nhiều bit. Mô hình Raster thuận lợi cho việc thu nhận, hiển thị và in ấn. Các ảnh được sử dụng trong phạm vi của đề tài này cũng là các ảnh được biểu diễn theo mô hình Raster.

Khi xử lý các ảnh Raster chúng ta có thể quan tâm đến mối quan hệ trong vùng

lân cận của các điểm ảnh. Các điểm ảnh có thể xếp hàng trên một lưới (raster) hình vuông, lưới hình lục giác hoặc theo một cách hoàn toàn ngẫu nhiên với nhau.

****

Hình 1.3.a Quan hệ 8 lân cận Hình 1.3.b Quan hệ 4 lân cận

Cách sắp xếp theo hình vuông là được quan tâm đến nhiều nhất và có hai loại: điểm 4 lân cận (4 liền kề) hoặc 8 lân cận (8 liền kề).

Với điểm 4 lân cận, một điểm ảnh I(i, j) sẽ có điểm kế cận theo 2 hướng i và j; trong khi đó với điểm 8 lân cận, điểm ảnh I(i, j) sẽ có 4 điểm kế cận theo 2 hướng i, j và 4 điểm kế cận theo hướng chéo 450 (Xem hình 1.2).

Video thực tế là một chuỗi các ảnh, quan hệ thời gian giữa

các khung hình biểu diễn ảnh động. Nên quá trình thu nhận video là tương tự so với thu nhận ảnh. Chỉ có cách lưu trữ là khác nhau. Ta có thể tách video để được một hình ảnh riêng biệt. Hình ảnh này gọi là khung hình.

## 1.6 Một số định dạng ảnh phổ biến

### 1.6.1 File ảnh IMG

Ảnh IMG là ảnh đen trắng, phần đầu file IMG có 16 bytes chứa các thông tin cần thiết:

+ 6 bytes đầu dùng để đánh dấu nhận dạng file IMG. Giá trị của 6 bytes đầu này viết dưới dạng hexa: 0x0001 0x0008 0x0001.

+ 2 bytes chứa độ dài các mẫu tin. Đó là độ dài của một dãy các bytes lặp lại một số lần nào đó, số lần lặp này sẽ được lưu trong một file đếm. Nhiều dãy giống nhau được lưu trong một bytes. Đó chính là cách lưu trữ nén

+ 4 bytes tiếp theo mô tả kích cỡ của pixel

+ 2 bytes tiếp mô tả số pixel trên một dòng

+ 2 bytes cuối cho biết số dòng trong ảnh

Các dòng giống nhau được nén thành một pack. Có 4 loại pack sau:

+ Loại 1: Gói các dòng giống nhau.Quy cách gói tin này 0x00 0x00 0xFF Count. 3 bytes đầu cho biết số các dãy giống nhau ,bytes cuối cho biết số các dòng giống nhau.

+ Loại 2: Gói các dãy giống nhau. Quy cách gói này 0x00 Count. Bytes thứ hai cho số các dãy giống nhau được nén trong gói. Độ dài cử dãy được ghi đầu file.

+ Loại 3:Dãy các pixel không giống nhau, không lặp lại và không nén được. Quy cách như sau: 0x80 Count. Bytes thứ hai cho biết độ dài dãy các pixel không giống nhau, không nén được.

+ Loại 4:Dãy các pixel giống nhau. Tuỳ theo các bit cao của bytes đầu được bật hay tắt, nếu bit cao được bật(giá trị 1) thì đây là gói nén các bytes chỉ gồm bit 0, số các bytes được nén được tính bởi 7 bit thấp còn lại. Nếu bit cao tắt (giá tri 0) thì đay là gói nén các bytes toàn bit 1. Số các bytes được nén được tính bởi 7 bit thấp còn lại.

Các kiểu file IMG phong phú như vậy là do ảnh IMG là ảnh đen trắng nên chỉ cần 1 bit cho một pixel thay vì 4 hoặc 8 bit như đã nói ở trên toàn bộ ảnh chỉ có điểm sáng hoặc tối tương ứng với 1 hoặc 0. Tỷ lệ nén của file này là khá cao.

### 1.6.2 File ảnh BMP

*1.6.2.1 Khái niệm về ảnh đen trắng, ảnh màu, ảnh cấp xám*

+ Ảnh đen trắng.

Đó là những bức ảnh mà mỗi điểm ảnh chỉ là những điểm đen hoặc trắng, được quy định bằng một bit. Nếu bit mang giá trị là 0 thì điểm ảnh là điểm đen, còn nếu mang giá trị là 1 thì điểm ảnh là điểm trắng. Do đó để biểu diễn một điểm ảnh đen trắng ta có thể dùng một ma trận nhị phân, là ma trận mà mỗi phần tử chỉ nhận một trong hai giá trị là 0 hoặc 1.

+ Ảnh màu

Quá trình giấu tin vào ảnh màu cũng tương tự như với ảnh đen trắng nhưng trước hết ta phải chọn từ mỗi điểm ảnh ra bit có trọng số thấp nhất (LSB) để tạo thành một ảnh nhị phân gọi là ảnh thứ cấp. Sử dụng ảnh thứ cấp này như ảnh môi trường để giấu tin, sau khi biến đổi ảnh thứ cấp ta trả nó lại ảnh ban đầu để thu được ảnh kết quả.

+Ảnh đa cấp xám

Đối với ảnh đa cấp xám bảng màu của nó đã có sẵn, tức là những cặp màu trong bảng màu có chỉ số chênh lệch càng ít thì càng giống nhau. Vì vậy đối với ảnh đa cấp xám bit LSB của mỗi điểm ảnh là bit cuối cùng của mỗi điểm ảnh.

Quá trình tách bit LSB của ảnh đa cấp xám và thay đổi các bit này bằng thuật

toán giấu tin trong ảnh đen trắng sẽ làm chỉ số của điểm màu bị thay đổi tăng hoặc giảm 1 đơn vị, do đó điểm ảnh mới sẽ có độ sáng tối của ô màu liền trước hoặc liền sau ô màu của điểm ảnh cũ. Bằng mắt thường rất khó có thể nhận thấy sự thay đổi về độ sáng tối này.

+ Ảnh nhỏ hơn hoặc bằng 8 bits màu

Những ảnh thuộc loại này gồm có 16 màu (4 bit màu) và ảnh 256 màu (8 bits màu). Khác với ảnh màu, ảnh xám với số bit nhỏ hơn hoặc bằng 8 bit không phải luôn luôn được sắp xếp màu bảng màu.

Những màu ở liền kề nhau trong bảng màu có thể rất khác nhau chẳng hạn như

màu đen với màu trắng vẫn có thể được xếp cạnh nhau.

Vì vậy việc xác định bit LSB của ảnh loại này rất khó. Nếu ta chỉ làm như đối

với ảnh xám, tức là vẫn lấy bit cuối cùng của mỗi điểm ảnh để tạo thành ảnh thứ cấp thì mỗi thay đổi 0 sang 1 hoặc 1 sang 0 trên ảnh thứ cấp có thể làm cho ảnh màu của điểm ảnh cũ và mới tương đương ứng thay đổi rất nhiều dù chỉ số màu của chúng cũng tăng hoặc giảm 1 mà thôi.

+ Ảnh hightcolor (16 bits màu)

Ảnh 16 bit màu thực tế chỉ sử dụng 15 bit cho mỗi điểm ảnh trong đó 5 bit biểu diễn cường độ tương đối của màu đỏ, 5 bit biểu diễn cường độ tương đối của màu xanh lam, 5 bit biểu diễn cường độ tương đối của màu xanh lơ. Còn lại một bit không dùng đến là bit cao nhất của byte thứ hai trong mỗi cặp thứ hai byte biểu diễn một điểm ảnh, đó chính là bit LSB của ảnh 16 bit màu. Việc thay đổi giá trị của những bit này sẽ không hề ảnh hưởng tới màu sắc của từng điểm ảnh trong môi trường.

+ Ảnh true color (24 bit màu)

Ảnh true color sử dụng 3 byte cho mỗi điểm ảnh, mỗi byte biểu diễn một thành phần trong cấu trúc RGB. Trong mỗi byte các bit cuối cùng của mỗi byte trong phần dữ liệu ảnh là các bit LSB của ảnh true color.

Để tăng lượng thông tin giấu được vào ảnh môi trường, từ mỗi byte của ảnh true

color ra sẽ lấy nhiều hơn một bit để tạo thành ảnh thứ cấp. Thông thường cũng chỉ nên lấy nhiều nhất 4 bit cuối cùng của mỗi byte để ảnh kết quả không bị nhiễu đáng kể, khi đó lượng thông tin tối đa có thể giấu trong ảnh cũng tăng lên gấp bốn lần so với lượng thông tin tối đa giấu được trong ảnh đó nếu chỉ lấy 1 bit cuối cùng ở từng byte.

*1.6.2.2. Cấu trúc ảnh BMP*

Để thực hiện việc giấu tin trong ảnh, trước hết ta phải nghiên cứu cấu trúc của ảnh và có khả năng xử lý được ảnh tức là phải số hoá ảnh. Quá trình số hoá các dạng ảnh khác nhau và không như nhau. Có nhiều loại ảnh đã được chuẩn hoá như: JPEG, PCX, BMP…

Sau đây là cấu trúc ảnh \*.bmp. Mỗi file ảnh BMP gồm 3 phần:

- BitmapHeader (54 byte)

- Palette màu (bảng màu)

- BitmapData (thông tin ảnh)

Cấu trúc cụ thể của ảnh:

- Palette màu (bảng màu): bảng màu của ảnh, chỉ những ảnh lớn hơn hoặc bằng 8bits màu mới có Palette màu.

- BitmapData (thông tin ảnh): phần này nằm ngay sau phần palette màu của ảnh

BMP. Đây là phần chứa giá trị màu của điểm ảnh trong ảnh BMP, các dòng ảnh được lưu từ dưới lên trên, các điểm ảnh được lưu từ trái sang phải. Giá trị của mỗi điểm ảnh là một chỉ số trỏ tới phần tử màu tương ứng của palette màu.

- Thành phần BitCount của cấu trúc BitmapHeader cho biết số bit dành cho mỗi

điểm ảnh và số lượng màu lớn nhất của ảnh. BitCount có thể nhận các giá trị sau:

* 1: Bitmap là ảnh đen trắng, mỗi bit biểu diễn 1 điểm ảnh. Nếu bit mang giá trị 0.
* thì điểm ảnh là đen, bit mang giá trị 1 điểm ảnh là điểm trắng.
* 4: Bitmap là ảnh 16 màu, mỗi điểm ảnh được biểu diễn bởi 4 bit.
* 8: Bitmap là ảnh 256 màu, mỗi điểm ảnh biểu diễn bởi 1 byte.
* 16: Bitmap là ảnh highcolor, mỗi dãy 2 byte liên tiếp trong bitmap biểu diễn.
* cường độ tương đối của màu đỏ, xanh lá cây, xanh lơ của một điểm ảnh.
* 24: Bitmap là ảnh true color (224 màu), mỗi dãy 3 byte liên tiếp trong bitmap biểu diễn cường độ tương đối của màu đỏ, xanh lá cây, xanh lơ (RGB) của một điểm ảnh.

- Thành phần ColorUsed của cấu trúc BitmapHeader xác định số lượng màu của

palette màu thực sự được sử dụng để hiển thị bitmap. Nếu thành phần này được đặt là 0, bitmap sử dụng số màu lớn nhất tương ứng với giá trị của BitCount.

*Bảng 1‑1 Giải thích BitmapHeader (54 bytes)*

|  |  |  |  |
| --- | --- | --- | --- |
| Byte | Đặt tên | Ý nghĩa | Giá trị |
| 1-2 | ID | Nhận dạng file | “BMP” hoặc 19778 |
| 3-6 | File\_Size | Kích thước file | Long trong C |
| 7-10 | Reserved | Dành riêng | 0 |
| 11-14 | Offset | Byte bắt đầu vùng dữ liệu | Offset của byte bắt đầu vùng dữ liệu |
| 15-18 | Isize | Số byte dành cho vùng thông tin | 40 |
| 19-22 | Width | Chiều rộng của ảnh | Tính bằng pixel |
| 23-26 | Height | Chiều cao của ảnh | Tính bằng pixel |
| 27-28 | Planes | Số planes màu | 1 |
| 29-30 | bitCount | Số bits cho một pixel | 1,4,6,16,24 |
| 31-34 | Compression | Kiểu nén dữ liệu | 0: Không nén  1: Nén runlength 8 bits/pixel  2: Nén runlength 4 bits/pixel |
| 35-38 | ImageSize | Kích thước ảnh | Tính bằng byte |
| 39-42 | XpelsPerMeter | Độ phân giải ngang | Tính băng pixel/meter |
| 43-46 | YpelsPerMeter | Độ phân giải dọc | Tính băng pixel/meter |
| 47-50 | ColorsUsed | Số màu sử dụng trong ảnh |  |
| 51-54 | ColorsImportant | Số màu sử dụng khi hiện ảnh |  |

### 1.6.3 File ảnh PNG

Là một dạng hình ảnh sử dụng phương pháp nén dữ liệu mới – không làm mất đi dữ liệu gốc. PNG được tạo ra nhằm cải thiện và thay thế định dạng ảnh GIF với một định dạng hình ảnh không đòi hỏi phải có giấy phép sáng chế sử dụng. PNG được hỗ trợ bởi thư viện tham chiếu libpng, một thư viện nền độc lập bao gồm các hàm được viết bằng C để quản lý các hình ảnh PNG.

Những tập tin PNG thường có phần mở rộng là PNG và đã được gán kiểu chuẩn

MIME là image/png.

Một tập tin PNG bao gồm 8 – byte kí hiệu (89 50 4E 47 0D 0A 1A) được viết trong hệ thống có cơ số 16, chứa các chữ “PNG” và 2 dấu xuống dòng, ở giữa là xếp theo số lượng của các thành phần, mỗi thành phần đều chứa thông tin về hình ảnh.

Cấu trúc dựa trên các thành phần được thiết kế cho phép định dạng PNG có thể tương thích với các phiên bản cũ khi sử dụng. Các “thành phần” trong tập tin. PNG là cấu trúc như một chuỗi các thành phần, mỗi thành phần chứa kích thước, kiểu, dữ liệu, và mã sửa lỗi CRC ngay trong nó.

Chuỗi được gán tên bằng 4 chữ cái phân biệt chữ hoa chữ thường. Sự phân biệt này giúp bộ giải mã phát hiện bản chất của chuỗi khi nó không nhận dạng được. Với chữ cái đầu, viết hoa thể hiện chuỗi này là thiết yếu. Chuỗi thiết yếu chứa thông tin cần thiết để đọc được tệp và nếu bộ giải mã không nhận dạng được chuỗi thiết yếu,việc đọc tệp phải được hủy.

Về cơ bản, định dạng PNG đem lại cho ta những ưu thế vượt trội hơn so với cácđịnh dạng phổ thông khác hiện nay như JPG, GIF, BMP…Những ưu thế tỏ rõ sứcmạnh hơn khi được sử dụng trong môi trường đồ họa web.

- Giảm thiểu dung lượng: Trong tất cả các định dạng ảnh phổ thông hiện nay thì hình ảnh PNG có thể coi là dung lượng nhỏ nhất. Điều này rất quan trọng khi sử dụng PNG trong môi trường web.

- Độ sâu của màu: Ảnh PNG hỗ trợ đến true color 48bit màu. Trong khi đó ảnh gif chỉ ở mức 256 màu.

### 1.6.4 File ảnh JPEG

JPEG (viết tắt cho Joint Photographic Experts Group) là một trong những phương pháp nén ảnh hiệu quả, có tỷ lệ nén ảnh tới vài chục lần. Tuy nhiên ảnh sau khi giải nén sẽ khác với ảnh ban đầu. Chất lượng ảnh bị suy giảm sau khi giải nén. Sự suy giảm này tăng dần theo hệ số nén. Tuy nhiên sự mất mát thông tin này là có thể chấp nhận được và việc loại bỏ những thông tin không cần thiết được dựa trên những nghiên cứu về hệ nhãn thị của mắt người.

Phần mở rộng của các file JPEG thường có dạng.jpeg,.jfif,.jpg,.JPG, hay.JPE; dạng .jpg là dạng được dùng phổ biến nhất. Hiện nay dạng nén ảnh JPEG rất được phổ biến trong Điện thoại di động cũng như những trang thiết bị lưu giữ có dung lượng nhỏ.Một nhóm các nhà nghiên cứu đã phát minh ra định dạng này để hiển thị các hình ảnh đầy đủ màu hơn (full-colour) cho định dạng di động mà kích thước file nhỏ hơn. Giống như GIF, JPEG cũng được sử dụng rất nhiều trên web. Lợi ích chính của chúng là có thể hiển thị các hình ảnh với các màu chính xác (true-colour) chúng có thể lên đến 16 triệu màu. Điều đó cho phép chúng được sử dụng tốt nhất cho các hình ảnh chụp và hình ảnh minh hoạ với lượng màu lớn.

Nhược điểm lớn nhất là chất lượng ảnh đã bị nén mất đi (lossy), một số đường bao giữa các khối màu sẽ xuất hiện điểm mờ, và các vùng sẽ mất đi sự rõ nét. Và giống như dạng mp3, JPEG sẽ không thể phục hồi giống như hình ảnh ban đầu dù dung lượng được tăng lên giống dung lượng ảnh thật.

Các ảnh JPEG không thể làm trong suốt hoặc chuyển động.

## 1.7 Kết thúc chương 1

Nội dung chương 1đã trình bày khái quát các bước cơ bản của một hệ thống xử lý ảnh điển hình, cơ bản. Đã đưa ra và bước đầu làm sáng tỏ bước đầu tiên của mọi hệ thống xử lý ảnh số, không phụ thuộc vào mục đích và lĩnh vực hệ thống đó được sử dụng. Đó chính là bước thu thập ảnh số. Tuy không đi sâu vào chi tiết cách thực hiện như các thuật toán lấy mẫu, lượng tử hóa diễn ra như thế nào. Nhưng chương 1 đã cung cấp cái nhìn tổng quan về sự hình thành nên ảnh số. Cách ảnh số được lưu trữ và biểu diễn trên máy tính. Tuy được lưu trữ bằng nhiều cách khác nhau, nhưng tựu chung lại ảnh số chính là một ma trận các điểm ảnh. Mỗi điểm ảnh có giá trị độ sáng và màu sắc. Các giá trị hàng và cột được gọi là độ phân giải của ảnh. Và quá trình xử lý ảnh sẽ chủ yếu làm việc với các điểm ảnh.

Bản chất của video cũng giống với ảnh. Video là một tập hợp của nhiều khung hình được chụp liên tiếp nhau của một cảnh. Tần số chụp lại cảnh đó càng cao, tức số khung hình được chụp trong một đơn vị thời gian càng lớn thì video sẽ càng chuyển động mượt mà. Đó có thể coi như sự số hóa về thời gian. Bên cạnh đó, video cũng có thể lưu tữ thêm kiểu dữ liệu âm thanh. Thường khi video được phát, dữ liệu âm thanh và hình ảnh sẽ được phát đồng bộ.

# Chương 2 BÀI TOÁN PHÂN LOẠI ĐỐI TƯỢNG TRONG ẢNH

## 2.1 Bài toán phân loại đối tượng trong ảnh

Như đã nói đến trong chương 1, tùy vào điều kiện và yêu cầu, ta có các bài toán về thị giác máy như nhận dạng, phân loại đối tượng. Vậy sự khác biệt giữa 2 lớp bài toán này là gì ?

Xét bài toán nhận dạng. Xét ví dụ 1 : cho một hình ảnh chụp đường giao thông. Hãy cho biết liệu trong hình có một chiếc xe motor mang biển số xác định nào đó cho trước.

Và ví dụ 2 : cho một hình ảnh chụp đường giao thông. Hãy cho biết liệu trong hình có một chiếc xe motor nào hay không.

Ở ví dụ thứ nhất. Chúng ta có thể nhận ra được vấn đề không quá khó khăn khi mà ta đã có một biển số để đối chiếu. Vì biến số xe đã là một mẫu. Chỉ cần so khớp với mẫu là ta có thể giải quyết được vấn đề của ví dụ 1. Thậm chí nếu số lượng biến số xe có nhiều lên thì việc tìm kiếm một mẫu khớp với biển số cho trước vẫn khả thi.

Ở ví dụ thứ hai. Vấn đề trở nên khó khăn hơn khi số lượng chủng loại xe motor lưu thông có thể rất lớn. Ta cũng có thể tập hợp nên rất nhiều mẫu xe motor để làm thành bộ so khớp. Nhưng sẽ có những chiếc motor có hình dạng khác hoàn toàn so với các xe có trong bộ mẫu. Bên canh đó, dù là hai chiếc xe cùng một mẫu, cùng một chủng loại khi xuất hiện trên đường cũng có thể rất khác nhau tùy vào góc chụp, người ngồi trên xe, hoặc cũng có thể là các mảnh vỡ của vỏ xe,…

Vấn đề ở ví dụ 2 khác với ở ví dụ 1 ở chỗ đối tượng đối tượng cần xem xét ở ví dụ 2 lớn hơn ở ví dụ 1. Và ở ví dụ 1, đã có một khuôn mẫu đối chiếu, trong khi ở ví dụ 2 không có một khuôn mẫu cụ thể nào.

Đó chính là bài toán phân loại đối tượng trong ảnh mà ta cần làm rõ trong chương này. Đối tượng cần nhận dạng trong bài toán phân loại phức tạp và có số lượng các đối tượng trong cùng một lớp lớn.

Chúng ta cần tìm ra một mô hình chung cho lớp đối tượng cần nhận dạng. Mô hình này có thể đúng với phần lớn các thể hiện của đối tượng. Trong khóa luận này, em đã sử dụng thuật toán “Viola and Jones” được triển khai và tích hợp trong thư viện OpenCV3.4.1 để nhận dạng phương tiện giao thông và giải quyết bài toán phân loại đối tượng giao thông trong ảnh.

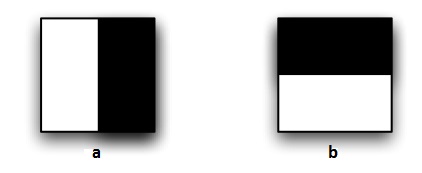
## 2.2 Thuật toán “Viola and Jones”

### 2.2.1 Đặc trưng Haar

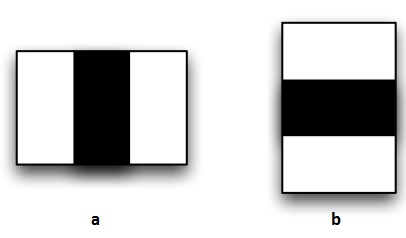
Xét bài toán nhận dạng khuôn mặt. Khuôn mặt được đặc trưng bởi tập hợp các pixel trong vùng khuôn mặt mà các pixel này tạo lên những điểm khác biệt so với các vùng pixel khác. Tuy nhiên với một ảnh đầu vào, việc sử dụng các pixel riêng lẻ lại không hiệu quả. Vì vậy những nhà nghiên cứu đã đưa ra tư tưởng kết hợp các vùng pixel với nhau tạo đặc trưng có khả năng phân loại tốt các vùng của khuôn mặt. Trong số đó đặc trưng Haar đã được ứng dụng.

Đặc trưng Haar được tạo thành bằng việc kết hợp các hình chữ nhật đen, trắng với nhau theo một trật tự, một kích thước nhất định.

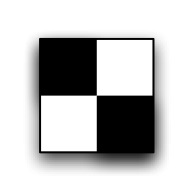
Các đặc trưng được phân loại như các hình sau:



Hình 2.1 Đặc trưng cạnh



Hình 2.2 Đặc trưng đường



Hình 2.3 Đặc trưng chữ nhật

### 2.2.1 Tính giá trị đặc trưng Haar

Giả sử với khung hình 24 x 24.

Để tính giá trị các đặc trưng haarlike, ta tính sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng theo công thức sau: vùng đen (pixel) – Tổng vùng trắng (pixel).

Hình 2.4 miêu tả cách tính đặc trưng haarlike với đặc trưng 2.1.a với chiều cao là 1 điểm ảnh. Như vậy của đặc trưng sẽ là hình chữ nhật kích thước 1x2. Cho hình chữ nhật này trượt trên hàng đầu tiên. Nó sẽ tiếp cận được 23 vị

trí. Như vậy với 24 hàng ta sẽ tính được 23 x 24 đặc trưng.

Tiếp tục với hình chữ nhật kích thước 1 x 4. Ở hàng đầu tiên, nó sẽ tiếp cận được 21 vị trí và cũng tương tự như vậy với các hàng tiếp theo. Như vậy ta sẽ tính được 21 x 24 đặc trưng.

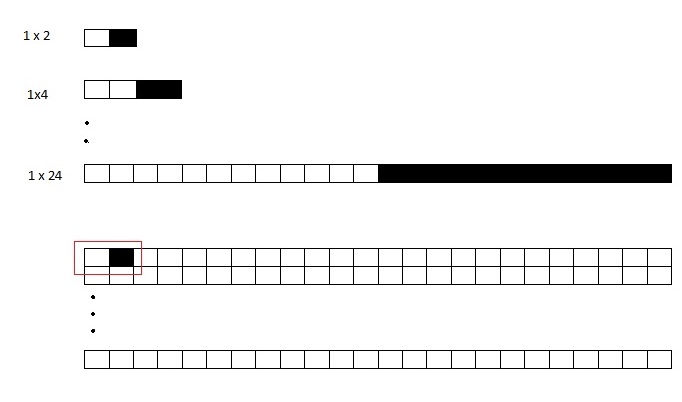
Tiếp tục qui trình trên với các hình chữ nhật 1 x 6, 1x8,…, 1 x 24.

Ta sẽ tính được có : ( 23 + 21 + 19 + … + 1 ) x 24 đặc trưng.

Lại áp dụng cách tính tương tự với các hình chữ nhật có chiều cao 2, 3, …, 24 điểm ảnh, và chiều dài 2, 4, 6, …, 24 điểm ảnh.

Ta có tổng số đặc trưng Haar cần tính cho đặc trưng 2.1.a là:

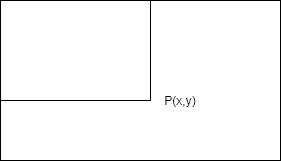
( 23 + 21 + 19 + … + 1 ) x ( 24 + 23 + 22 + … + 1) = 43200 đặc trưng 2.1.a .

Hình 2.4 Minh họa tính đặc trưng Haarlike với đặc trưng 1 x 2

Áp dụng nguyên lý tương tự. Với đặc trưng 2.1.b là 43200 đặc trưng. Với đặc trưng 2.2.a và 2.2.b là 27600 đặc trưng. Với đặc trưng 2.3 là 20736 đặc trưng. Như vậy tổng số 162336 đặc trưng cho tất cả.

Như vậy, để tính toán các giá trị của đặc trưng haarlike cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn. Do đó những nhà nghiên cứu đã đưa ra một khái niệm gọi là Intergral Image để tính toán nhanh cho các đặc trưng cơ bản. Intergral Image là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính các đặc trưng Haar, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên và bên trái của nó. Bắt đầu từ vị trí trên, bên trái đến vị trí dưới, phải của ảnh, việc tính toán này chỉ dựa trên phép cộng số nguyên đơn giản.

Cách tính Intergral Image của ảnh:

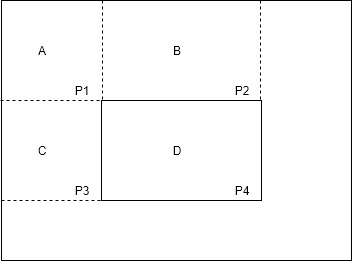


Hình 2.5 Ví dụ tính Intergral Image

Ta có: P(x,y) = x’≤x,y’≤y i(x’,y’)

Sau khi đã tính được Intergral Image của ảnh, việc tính tổng điểm ảnh của một vùng bất kì nào đó trên ảnh được thực hiện như sau:

Giả sử ta cần tính tổng điểm ảnh của vùng D như trong hình 2.6:



Hình 2.6 Tính Intergral Image của vùng ảnh D

D = A + B + C + D – (A + B) – (A +C) + A

Với A + B + C + D chính là giá trị tại điểm P4 trên Intergral Image, A + B là giá trị tại điểm P2, A + C là giá trị tại điểm P3, và A là giá trị tại điểm P1.

Vậy ta có thể tính lại biểu thức D ở trên như sau:

D = P(x4,y4) – P(x2,y2) – P(x3,y3) + P(x1,y1)

Để chọn các đăc trưng Haar dùng cho việc thiết lập ngưỡng, Viola và Jones sử dụng một phương pháp máy học được gọi là Adaboost. Adaboost sẽ kết hợp các bộ phân loại yếu để tạo thành một bộ phân loại mạnh.

### 2.2.2 Thuật toán Adaboost

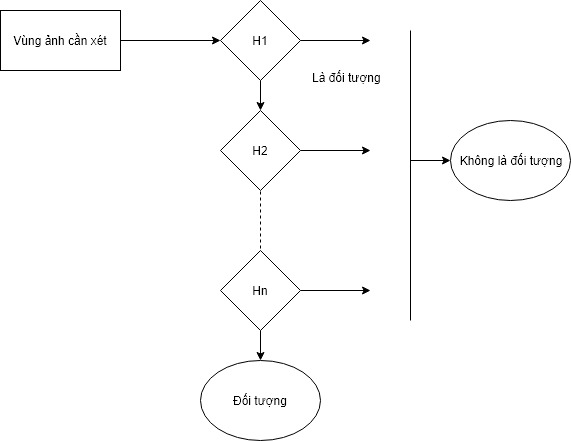
Adaboost là một cách trong hướng tiếp cận dựa trên diện mạo, Viola và Jones dùng AdaBoost kết hợp mô hình phân tầng cascade để xác định khuôn mặt người với các đặc trưng dạng Haar.

Tốc độ xử lý khá nhanh và tỷ lệ chính xác hơn 80% trên ảnh xám. Adaboost là một bộ phân loại mạnh phi tuyến phức dựa trên tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995.

Adaboost cũng hoạt động dựa trên nguyên tắc kết hợp tuyến tính các weak classifiers để hình thành một strong classifier. Là một cải tiến của của tiếp cận boosting. Adaboost sử dụng thêm khái niệm trọng số (weight) để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi weak classifiers được xây dựng, thuật toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng weak classifier kế tiếp: tăng trọng số của các mẫu bị nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi các weak classifier vừa xây dựng. Bằng cách này weak classifier sau có thể tập trung vào các mẫu mà các weak classifier trước nó làm chưa tốt. Sau cùng, các weak classifier sẽ được kết hợp tùy theo mức độ tốt của chúng để tạo nên strong classifier.

Ví dụ: để biết một ảnh có phải là ảnh mặt người không, ta hỏi N người (tương đương với N classifier xây dựng từ N vòng lặp của boosting), đánh giá của mỗi người (tương đương với một weak classifier) chỉ cần tốt hơn ngẫu nhiên một chút (tỉ lệ sai dưới 50%). Sau đó ta sẽ đánh trọng số cho đánh giá của từng người, người nào có khả năng đánh giá tốt các mẫu khó thì mức độ quan trọng của người đó trong kết luận cuối cùng sẽ cao hơn những người chỉ đánh giá tốt được các mẫu dễ. Việc cập nhật lại trọng số của các mẫu sau mỗi vòng lặp boosting chính là để đánh giá độ khó của các mẫu (mẫu có nhiều người đánh giá sai là mẫu càng khó).

“Viola and Jones” dùng Adaboost kết hợp các bộ phân loại yếu sử dụng các đặc trưng haar-like theo mô hình phân tầng (cascade) như sau:



Hình 2.7 Mô hình phân tầng kết hợp các bộ phân loại yếu để xác định đối tượng

Các weak classifier hk(x) là các bộ phân loại yếu được biểu diễn như sau: Trong đó:

- x: cửa sổ con cần xét

- : ngưỡng

- : giá trị của đặc trưng Haar

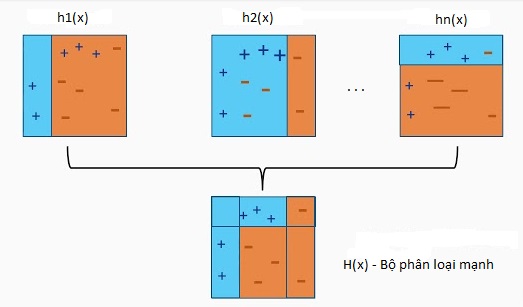
- : hệ số quyết định chiều của phương trình

Nếu giá trị vectơ đặc trưng của mẫu cho bởi hàm lượng giác của bộ phân loại vượt qua một ngưỡng cho trước thì mẫu object (đối tượng cần nhận dạng), ngược lại mẫu là background (không phải đối tượng).

Adaboost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau:

H(x) = sgn(a1h1(x) + a2h2(x) + … + anhn(x) = alpha

Với an >= 0 là hệ số chuẩn hóa cho các bộ phân loại yếu



Hình 2.8 Kết hợp các bộ phân loại yếu thành bộ phân loại mạnh

Mỗi bộ phân loại yếu sẽ quyết định kết quả cho một đặc trưng haarlike, được xác định ngưỡng đủ nhỏ sao cho có thể vượt được tất cả các bộ dữ liệu mẫu trong tập dữ liệu huấn luyện (số lượng ảnh khuôn mặt trong tập huấn luyện lớn). Trong quá trình xác định khuôn mặt người, mỗi vùng ảnh con sẽ được kiểm tra với các đặc trưng trong chuỗi các đặc trưng haarlike, nếu có một đặc trưng haarlike nào cho ra kết quả là khuôn mặt người thì các đặc trưng khác không cần xét nữa. Thứ tự xét các đặc trưng trong chuỗi các đặc trưng haarlike sẽ được dựa vào trọng số (weight) của đặc trưng đó do Adaboost quyết định dựa vào số lần và thứ tự xuất hiện của các đặc trưng Haar.

### 2.2.3 Khai triển thuật toán Adaboost

- Cho một tập huấn luyện gồm N mẫu có đánh dấu (x1, y2), (x2, y2), , (xn, yn). Với xk X = (x1k, x2k,…,xmk) là vector đặc trưng và yk (-1,1) là nhãn của mẫu (1 ứng với object, -1 ứng với background).

- Khởi tạo trọng số ban đầu cho tất cả các mẫu: với m là số mẫu đúng (ứng với object và y= 1) và l là số mẫu sai (ứng với background và y= -1).

w1 = ; k =

- Xây dựng T weak classifier

Lặp t = 1 đến T

+ Với mỗi đặc trưng trong vector đặc trưng, xây dựng một weak classifier hj với ngưỡng θj và lỗi εj.

+ Chọn ra hj với εj nhỏ nhất, ta được ht:

ht : X → {1,-1}

+ Cập nhật lại trọng số:

+ Trong đó:

Zt: hệ số dùng để đưa Wt+1 về đoạn [0, 1] (normalization factor) .

- Strong classifier được xây dựng:

Quá trình huấn luyện bộ phân loại được thực hiện bằng một vòng lặp mà ở mỗi bước lặp, thuật toán sẽ chọn ra weak classifier ht thực hiện việc phân loại với lỗi nhỏ nhất (do đó sẽ là bộ phân loại tốt nhất) để bổ sung vào strong classifier. Mỗi khi chọn được một bộ phân loại ht Adaboost sẽ tính giá trị theo công thức ở trên. cũng được chọn trên nguyên tắc làm giảm thiểu giá trị lỗi .

Hệ số nói lên mức độ quan trọng của ht.

Trong công thức của bộ phân loại H(x), ta thấy tất cả bộ phân loại ht đều có đóng góp vào kết quả phân loại của H(x), và mức độ đóng góp của chúng phụ thuộc và giá trị at tương ứng. at càng lớn thì nó càng có vai trò quan trọng trong H(x).

Trong công thức tính at:

Dễ thấy giá trị at tỉ lệ nghịch với ej. Bởi vì ht được chọn với tiêu chí ej nhỏ nhất, do đó at sẽ lớn nhất. Công thức này do Freund và Schapire đưa ra.

Sau khi tính được giá trị at, Adaboost tiến hành cập nhật lại trọng số của các mẫu: tăng trọng số các mẫu mà ht phân loại sai, giảm trọng số các mẫu mà ht phân loại đúng. Bằng cách này, trọng số các mẫu phản ánh được độ khó nhận dạng của mẫu đó và ht+1 sẽ ưu tiên học cách phân loại các mẫu này.

Vòng lặp xây dựng strong classifier sẽ dừng lại sau T lần lặp. Trong thực tế cài đặt huấn luyện, người ta ít sử dụng giá trị T vì không có công thức đảm bảo tính được giá trị T tối ưu cho quá trình huấn luyện. Thay vào đó, ta sử dụng giá trị max false positive hay max false alarm ( tỉ lệ nhận dạng sai tối đa các mẫu background). Tỉ lệ này của bộ phận cần xây dựng không được phép vượt quá giá trị này. Khi đó, qua các lần lặp, false alarm của strong classifier Ht(x) xây dựng được sẽ giảm dần, và vòng lặp sẽ kết thúc khi tỉ lệ này thấp hơn max false alarm.

2.3 Kết thúc chương 2   
 Chương 2 đã nêu lên điểm khác biệt giữa bài toán nhận dạng và bài toán phân loại đồng thời cũng là phạm vi áp dụng của mỗi bài toán.

Mô hình Cascade là mô hình phân tầng với mỗi tầng là một mô hình Adaboost sử dụng bộ phân lớp yếu là cây quyết định với đặc trưng Haar. Trong quá trình huấn luyện, bôh phân lớp phải duyệt qua tất cả các đặc trưng của mẫu trong tập huấn luyện. Điều này có thể mất chi phí thời gian rất lớn. Tuy vậy, trong các mẫu nhận dạng, sẽ tồn tại những mẫu background dễ nhận ra ( mẫu background đơn giản). Đối với những mẫu này, chỉ cần xét một vài đặc trưng đơn giản là có thể nhận ra chứ không cần xét hết tất cả các đặc trưng. Nhưng đối với các bộ phân loại thông thường thì cho dù mẫu nhận dạng dễ hay khó thì nó vẫn xét hết tất cả các đặc trưng mà nó được huấn luyện. Do đó đôi khi chứng gây ra sự tiêu tốn chi phí thời gian không cần thiết.

Mô hình Cascade of Classifiers được xây dựng nhằm tiết kiệm chi phí thời gian, giảm thiểu thỉ lệ nhận nhầm ( false alarm ) cho bộ phân loại. Cascade trees gồm nhiều tầng ( stage ), mỗi tầng là một mô hình AdaBoost với bộ phân loại yếu là các cây quyết định. Một mẫu để được phân loại là đối tượng thì phải đi qua tất cả các tầng. Các tầng sau được huấn luyện bằng những mẫu phủ định ( không phải là đối tượng cần phân loại) mà tầng trước nó nhận dạng sai, tức là nó sẽ tập trung học từ các mẫu background khó hơn, đo đó sự kết hợp này sẽ giúp tối thiểu hóa sự phân loại nhầm. Đồng thời, những mẫu background dễ nhận dạng sẽ được loại bỏ từ những tầng đầu. Giúp tiết kiệm chi phí thời gian mà vẫn giữ được hiệu quả.

# Chương 3 KỸ THUẬT TRACKING

## 3.1 Khái niệm Tracking

Tracking(Theo dõi) là một phần cơ bản của thị giác máy và được sử dụng rộng rãi trong các ứng dụng liên quan đến an ninh, giám sát, tìm đường… Có thể lấy ví dụ như các hệ thống dẫn đường tự động, các hệ thống cảnh báo va chạm,… Theo dõi có thể được định nghĩa thông qua bài toán gán nhãn cho một đối tượng được theo dõi trong nhiều khung hình khác nhau của một video.

Trong thực tế, việc theo dõi đối tượng có thể trở nên rất phức tạp vì các nguyên nhân: nhiễu ảnh, vật thể không xác định, vật thể chỉ xuất hiện một phần- không hoàn chỉnh,…

Để giải quyết bài toán này, chúng ta cần tìm hiểu về các thuật toán tracking. Trong giới hạn của khóa luận này em xin trình bày về thuật toán tracking được sử dụng trong chương trình. Đó là thuật toán tracking dựa trên trọng tâm (centroid based tracking).

## 3.2 Kỹ thuật Tracking

### 3.2.1 Quy trình chung

Nguyên lý chung của một hệ thống tracking là:

1. Nhận dạng tập các đối tượng các vật thể cần tracking.
2. Gán cho mỗi vật thế một định danh duy nhất.
3. Theo dõi các đối tượng khi chúng chuyển động trong video, tức là thay đối vị trí trong mỗi khung hình, và đảm bảo việc giữ đúng định danh của chúng.

Chúng ta đã có công cụ để giải quyết bước (1) trong quá trình trên. Sau đó sử dụng thuật toán centroid based tracking ở mức đơn giản để gắn nhãn và đếm số lượng xe máy lưu thông qua một đoạn đường.

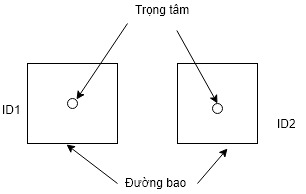
### 3.2.2 Thuật toán centroid based tracking

*3.2.2.1 Tạo đường bao và tính giá trị trọng tâm*

Thuật toán yêu cầu tập vị trí của các đối tượng được xác định trong mỗi khung hình của video.

Giả sử rằng chúng ta đã nhận diện và có một tập các đối tượng cần tracking. Có nghĩa là ta đã có được đường bao và tâm của đối tượng. Giả sử rằng đường bao là hình chữ nhật và trọng tâm của đối tượng là trọng tâm của hình chữ nhật bao quanh chúng.

Với bước khởi tạo, các đối tượng lần đầu được phát hiện trong khung hình nên sẽ được gán các giá trị định danh.

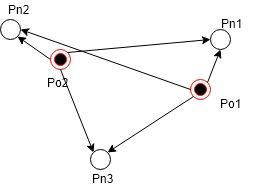


Hình 3.1 Khởi tạo định danh cho các đối tượng cần tracking

*3.2.2.2 Tính khoảng cách giữa các đường bao và các đối tượng đang xuất hiện*

Với mỗi khung hình trong video, chúng ta lặp lại bước 1, tuy nhiên thay vì gán định danh cho các đối tượng được phát hiện. Chúng ta cần phải xem xét xem liệu đối tượng trong khung hình mới có thực sự là đối tượng mới xuất hiện hay nó vẫn là một đối tượng cũ trong khung hình trước với một vị trí mới.

Phương pháp được sử dụng đó là tính khoảng cách Euclidean của mỗi đối tượng trong khung hình cũ với các đối tượng mới được phát hiện trong khung hình mới. Giá trị nhỏ nhất trong số đó cho ta dự đoán về vị trí của đối tượng trong khung hình cũ.



Hình 3.2 Tính giá trị Euclidean của các cặp hình cũ mới

Trong hình 3.2, điểm Po1 và Po2 là các điểm trọng tâm của các đối tượng trong khung hình cũ F(t), các điểm Pn1, Pn2, Pn3 là trọng tâm của các đối tượng mới F(t+1).

*3.2.2.3 Cập nhật tọa độ của đối tượng*

Tại thời điểm khung hình F(t+1) xuất hiện. Ta đã phát hiện các đối tượng trong khung hình F(t+1). Và mỗi đối tượng trong khung hình F(t) đều có khoảng cách Euclidean của mình với các đối tượng xuất hiện trong khung hình mới. Giá trị nhỏ nhất trong số đó chỉ ra vị trí của đối tượng đó trong khung hình F(t+1).

Trong hình 3.1 ta nhận thấy trong các giá trị khoảng cách của các đối tượng mới với điểm Po1, khoảng cách Po1 và Pn1 là nhỏ nhất. Vậy điểm Pn1 là vị trí mới của đối tượng trước đó có vị trí Po1. Tương tự với điểm Pn2.

*3.2.2.4 Gán nhãn cho đối tượng mới*

Đối với điểm Pn3. Ta không thể tìm được điểm nào có trong khung hình có thể bắt cặp với nó. Vậy ta coi nó như một đối tượng mới xuất hiện và gán định danh cho nó.

*3.2.2.5 Hủy định danh của một đối tượng.*

Vì lí do nào đó, đối tượng không còn xuất hiện trong khung hình nữa. Chúng ta cần phải hủy định danh của đối tượng đó. Tùy vào bài toán yêu cầu ta có thể đưa ra các điều kiện để hủy chúng. Có thể kể đến như: khi khoảng cách giữa vị trí của vật thể đến vị trí đầu tiên mà nó xuất hiện vượt quá giá trị giới hạn. Hoặc tọa độ của đối tượng vượt ra ngoài khoảng giá trị cho trước.

## 3.3 Kết luận chương 3.

Thuật toán tracking centroid based tracking có 2 hạn chế chính:

Thứ nhất, thuật toán đòi hỏi việc phát hiện phải được thực hiện trong mỗi khung hình. Tuy vậy với các kỹ thuật phát hiện nhanh ( ví dụ sử dụng Haar cascade ) có thể khắc phục nhược điểm này. Đó là trong trường hợp tỉ lệ nhận dạng thành công khá cao. Tất cả các đối tượng có trong khung hình cũ được nhận dạng. Còn trong trường hợp nhận dạng không thành công, điều này có thể dẫn đến việc gây ra nhầm lẫn trong việc cập nhật nhãn.

Thứ hai, đó là vấn đề nằm trong chính thuật toán. Nếu có một đối tượng trong khung hình di chuyển quá nhanh đến mức khoảng cách Euclidean của nó với vị trí cũ quá lớn. Điều này có thể gây nhâm lẫn cho việc cập nhật nhãn.

Vấn đề tiếp theo là khi các đối tượng di chuyển hỗn loạn, không tuân theo quy luật, ta có thể cũng có thể rơi vào các trường hợp mà vị trí mới của một đối tượng nằm gần với vị trí cũ của một đối tượng khác hơn là so với vị trí cũ của chính nó.

Tuy vậy, thuật toán trên cũng có những ưu điểm là nhanh. Dễ triển khai.

Hơn nữa, trường hợp áp dụng trong khóa luận này là điều kiện giao thông trên đường thẳng. Việc các phương tiện giao thông di chuyển hỗn loạn là có thể xảy ra. Tuy vậy, nhìn chung các phương tiện đều di chuyển với tốc độ ổn định, trong một làn đường và hướng nhất định. Điều này có thể giảm thiểu sự nhầm lẫn giữa các đối tượng.

# Chương 4 XÂY DỰNG PHẦN MỀM PHÁT HIỆN VÀ PHÂN LOẠI PHƯƠNG TIỆN GIAO THÔNG

## 4.1 Bài toán

### 4.1.1 Yêu cầu 1

Đầu vào: Ảnh giao thông chụp chính diện các phương tiện giao thông, hướng di chuyển hoặc hướng chính diện của phương tiện giao thông đối diện trực tiếp với thiết bị thu ảnh.

Đầu ra: Đóng khung vào vị trí xuất hiện phương tiện giao thông. Cho biết loại phương tiện giao thông.

### 4.1.2 Yêu cầu 2

Đầu vào: Video giao thông quay chính diện các phương tiện giao thông, hướng di chuyển hoặc hướng chính diện của phương tiện giao thông đối diện trực tiếp với thiết bị quay video.

Đầu ra: Đóng khung vào vị trí xuất hiện phương tiện giao thông xe máy. Đếm số lượng phương tiện giao thông. Chụp ảnh phương tiện phát hiện được.

## 4.2 Phân tích yêu cầu và lựa chọn công cụ

### 4.2.1 Phân tích yêu cầu

Trong đề tài này, em tập trung vào phát triến một ứng dụng Desktop nhằm phân tích và chỉ ra , phân loại phương tiện giao thông trong ảnh cũng như đếm số lượng xe máy xuất hiện trong một video.

Do các khó khăn của bài toán nhận dạng như: Tư thế góc chụp, sự xuất hiện hoặc thiếu một số thành phần của đối tượng, sự đa dạng và phong phú của các chủng loại phương tiện giao thông, sự che khuất, hướng của ảnh, điều kiện của ảnh...

Vì thế trong luận văn này em đưa ra những giả định và ràng buộc sau làm giảm độ phức tạp bài toán nhưng không làm bài toán trở nên xa rời thực tế:

- Góc ảnh trực diện (frontal) hoặc gần như trực diện, không bị che khuất (no occlusion) .

- Đối tượng giao thông được phân loại vào hai lớp:

+ Xe hai bánh, có các đặc trưng như xe máy.

+ Xe bốn bánh, có các đặc trưng như xe sedan, SUV, xe bán tải và xe tải cỡ nhỏ.

- Ảnh được chụp trong điều kiện ánh sáng bình thường.

- Cơ sở dữ liệu ảnh nhỏ, độ lớn dữ liệu khoảng một ngàn ảnh.

- Ảnh được xét trong luận văn là ảnh số, nghĩa là ảnh mặt được chụp từ các thiết bị như máy ảnh số, camera,…. Luận văn không xem xét các ảnh mặt người nhân tạo như vẽ, điêu khắc …

### 4.2.2 Sơ lược về OpenCV

4.2.2.1 OpenCV là gì

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 14 triệu lần.

4.2.2.2 Các thành phần chính trong thư viện OpenCV

- core

Đây là module chứa các cấu trúc, class  cơ bản mà OpenCV sẽ sử dụng trong việc lưu trữ và xử lý hình ảnh như Mat, Scale, Point, Vec… và các phương thức cơ bản sử dụng cho các module khác

- imgproc

Đây là một module xử lý hình ảnh của OpenCV bao gồm các bộ lọc (filter) linear và non-linear và các phép biến đổi hình học (tranformation) như resize hoặc các phép biến đổi affine, chuyển đổi hệ mày và các thuật toán liên quan đến histogram (biểu đồ) của hình ảnh.

- highgui

Đây là một module cho phép tương tác với người dùng trên UI (User Interface) như hiển thị hình ảnh, video capturing.

- features2d

Module tìm các đặc trưng (feature) của hình ảnh. Trong module có implement các thuật toán rút trích đặc trưng như PCA…

- calib3d

Hiệu chuẩn máy ảnh và xây dựng lại 3D.

- objdetect

Module cho việc phát hiện các đối tượng như khuôn mặt, đôi mắt, cốc, người, xe hơi, vv trong hình ảnh. Các thuật toán được sử dụng trong module này là Haar‐like Features.

- ml

Module này chứa các thuật toán về Machine Learning phục vụ cho các bài toàn phân lớp (Classfitication) và bài toán gom cụm (Clustering). Ví dụ như thuật toán SVM (Support Vector Machine), ANN…

- video

Module phân tích video gồm ước lượng chuyển động, trừ nền, và các thuật toán theo dõi đối tượng (object tracking).

## 4.3 Huấn luyện bộ phát hiện

### 4.3.1 Thu thập dữ liệu

Đây là giai đoạn đầu và cũng là giai đoạn rất quan trọng trong quá trình huấn luyện bộ phát hiện cũng như trong xử lý bài toán. Như đã trình bày trong phần phân tích yêu cầu. Trong phần thu thập dữ liệu để huấn luyện cho xe máy, em đã đặt camera tại một điểm giao thông để thu thập dữ liệu giao thông.

Dữ liệu sau đó được trích xuất và dùng để huấn luyện bộ nhận dạng xe hai bánh. Dữ liệu mẫu và dữ liệu kiểm thử cũng được lấy ngay tại điểm giao thông đó, với cùng một góc máy, cùng một độ cao, cùng điều kiện thời tiết, cùng điều kiện ánh sáng. Dước đây minh họa một khung hình thu được từ camera, các phương tiện giao thông sau đó được trích xuất để huấn luyện bộ nhận dạng.

Hình 4.1 Minh họa dữ liệu mẫu

Đối với dữ liệu khẳng định cho xe hai bánh, chương trình dùng 140 ảnh xe máy được trích xuất từ camera. Bộ dữ liệu phủ định cho xe hai bánh dùng 1101 ảnh. Bộ dữ liệu phủ định này tập hợp các hình ảnh không phải xe hai bánh gồm các thành phần : nền trong các khung hình mà xe hai bánh xuất hiện, xe bốn bánh, các sự vật khác nhau,…

Đối với bộ nhận dạng cho xe bốn bánh. Xét bài toán nhận diện xe ô-tô đã phổ biến và nhận được sự quan tâm của cộng đồng từ rất lâu. Các bộ nhận dạng xe bốn bánh có cơ sở dữ liệu rất lớn và hoàn thiện. Và nhiều bộ dữ liệu hoàn thiện đã vượt quá khả năng mà các máy tính bình thường có thể huấn luyện được. Tuy vậy việc sử dụng những bộ nhận dạng này lại không gặp phải vấn đề gì về khả năng xử lý của máy tính.

Với những lí do trên, cộng với việc tác giả không yêu cầu về bản quyền và chia sẻ miễn phí bộ nhận dạng xe ô tô hướng chính diện có độ chính xác cao.

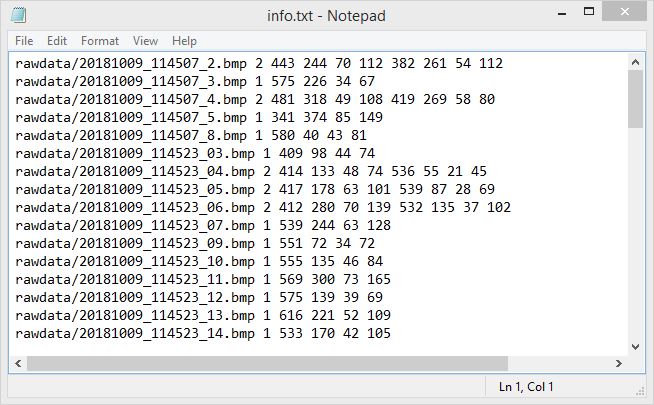
Em đã sử dụng bộ nhận dạng này cho mục đích phân loại ô tô, nguồn của nó tại <https://github.com/andrewssobral/vehicle_detection_haarcascades>.

Các chương trình hỗ trợ cho việc huấn luyện dữ liệu thủ công cũng đều được tích hợp sẵn trong thư viện OpenCV. Sau đây sẽ trình bày các bước để huấn luyện bộ nhận dạng sau khi đã thu thập dữ liệu ảnh về đối tượng xe máy.

### 4.3.2 Tạo file miêu tả cho mẫu dữ liệu khẳng định và phủ định

Mục đích của quá trình này là tạo ra một tệp chứa vị trí cụ thể của đối tượng cần nhận dạng trong một ảnh toàn cảnh.

Chương trình được tích hợp sẵn trong OpenCV3.4.1 là opencv\_annotation.exe sẽ giúp chúng ta thực hiện việc này. Chúng ta cần một file info.txt chứa tên của các tệp ảnh và miêu tả cị trí của các đối tượng như hình sau :



Hình 4.2 Ví dụ về file info.txt

Mỗi dòng bao gồm đường dẫn của ảnh và mô tả của các đối tượng trong ảnh lớn chứa nó. Ví dụ:

rawdata/20181009\_114507\_2.bmp 2 443 244 70 112 382 261 54 112

Như ví dụ trên giá trị đầu tiên chỉ đường dẫn của ảnh, giá trị tiếp theo chỉ ra số lượng đối tượng trong hình, mỗi đối tượng được miêu tả gồm 4 tham số thành phần như sau :

{tạo độ x, tọa độ y, chiều rộng, chiều cao }

Trường phái thứ hai ta của quá trình này đó là, mỗi file ảnh thuộc tập ảnh khẳng định chỉ chứa duy nhất một đối tượng hoàn chỉnh. Diện tích của đối tượng chiếm ít nhất 80% diện tích toàn ảnh. Và các miêu tả của các ảnh này đo đó đơn giản hơn và chỉ bao gồm 6 tham số lần lượt là đường dẫn ảnh, 1, 0, 0, chiều rộng ảnh, chiều cao ảnh.

Đối với file miêu tả dữ liệu phủ định, nó cũng có định dạng tương tự nhưng không cần thiết phải có các tham số chỉ ra đối tượng. Vì trong ảnh đó không chứa đối tượng và bản thân toàn bộ ảnh đó đã là dữ liệu phủ định.

### 4.3.3 Tạo file có định dạng “OpenCV data vector” từ tệp dữ liệu khẳng định

Chương trình được tích hợp sẵn trong OpenCV3.4.1 là opencv\_createsamples.exe sẽ giúp chúng ta thực hiện việc này.

Chạy chương trình:

opencv\_createsamples.exe –info annotations.txt –vec images.vec –bg

negatives.txt –num amountSamples –w model\_width –h model\_height

Trong đó:

- annotations.txt : là file miêu tả đã nhận được.

- images.vec : file kết quả của chương trình.

- negatives.txt : chứa đường dẫn của các file ảnh phủ định.

- amountSamples : số lượng các đối tượng có trong file annotations.txt

- model\_width, model\_height: kích thước nhỏ nhất của đối tượng mà bộ nhận dạng phát hiện được, với bộ nhận dạng xe máy, em dùng kích thước 15 x 30.

### 4.3.4 Huấn luyện bộ nhận dạng

Chương trình được tích hợp sẵn trong OpenCV3.4.1 là opencv\_traincascade.exe sẽ giúp chúng ta thực hiện việc này.

Chạy chương trình :

opencv\_traincascade.exe -data cascades -vec -bg -npos -nneg -nstages -mem -mode -w -h .

Một số tham số cơ bản :

- data : Nơi chứa kết quả của quá trình huấn luyện. Mặc định bao gồm file .xml kết quả của mỗi bước huấn luyện.

- numPos: Số lượng ảnh khẳng định cho mỗi bước huấn luyện bộ nhận dạng yếu.

- numNeg: Số lượng ảnh phủ định cho mỗi bước huấn luyện bộ nhận dạng yếu.

- numStages: Số bước huấn luyện các bộ nhận dạng yếu.

- bg: đường dẫn của negatives.txt

- vec: đường dẫn của images

- precalcValBufSize và –precalcIdxBufSize: lượng bộ nhớ dùng để tính toán các đặc trưng của các bộ nhận dạng yếu.

-minHitRate: định nghĩa số phần trăm ảnh khẳng định được phân loại chính xác mỗi bước. Số bước sẽ được thuật toán tự tính toán để đạt được tỉ lệ này.

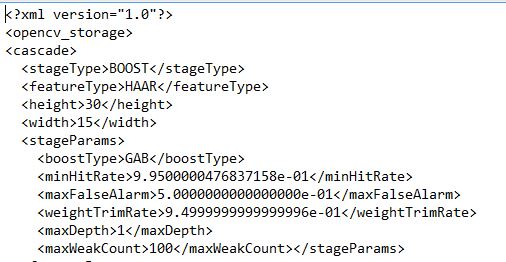
Mặc định tỉ lệ này là 0.995.

-maxFalseAlarmRate: định nghĩa tỉ lệ ảnh phủ định được phân loại trước khi quá trình boost dừng thêm bộ phân loại yếu vào bước hiện tại. Mặc định là 0.5.

Kết thúc quá trình huấn luyện ta sẽ thu được một file .xml chính là bộ nhận dạng được sử dụng.

### 4.3.5. Giải thích ý nghĩa của file cascade.

Phần đầu của file cascade chứa các thông số cơ bản như: đặc trưng được sử dụng, chiều cao, chiều rộng của đối tượng được sử dụng để huấn luyện, số lượng bộ phân loại yếu,…



Hình 4.3 Minh họa phần đầu của file cascade.

Tiếp theo đó, được mô tả trong hình 4.4.a là phần thông tin về cấu trúc của bộ phân loại, các kết quả tổng hợp và các ngưỡng cho mỗi quá trình boosting.

Thẻ maxWeakCount cho biết số lượng bộ phân loại yếu được sử dụng trong bước này.

Thẻ stageThreshold chỉ ra ngưỡng của bước này.

Thẻ internalNodes:

- Hai giá trị đầu chỉ ra 2 lá của gốc.

- Giá trị tiếp theo chỉ ra chỉ số của đặc trưng được sử dụng theo vị trí trong danh sách đặc trưng được sử dụng.

- Giá trị cuối cùng là ngưỡng được cài đặt cho bộ phân loại yếu này.

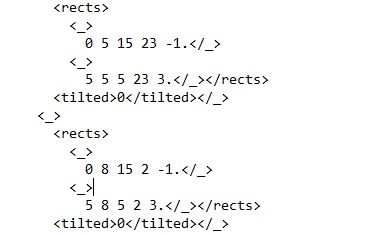


Hình 4.4.a Cấu trúc trong file cascade

Các giá trị trong mỗi nút của thẻ rects được mô tả trong hình 4.4.b

Bốn giá trị đầu thể hiện : tọa độ x, y của hình chữ nhật trong màn hình đồ họa,

Giá trị thứ 5 thể hiện trọng số của vùng đặc trưng đó.



Hình 4.4.b Cấu trúc trong file cascade.

## 4.4 Thiết kế giao diện



Hình 4.5 Giao diện chương trình

Giao diện chương trình bao gồm các thành phần:

- Khu vực hiển thị kết quả:

- Khu vực “Chọn đầu vào”:

+ Nút “Ảnh” : Chọn ảnh cần phân tích.

+ Nút “Video” : Chọn video cần phân tích.

- Khu vực “Chọn đầu ra”:

+ Nút “Lưu tại”: Chọn thư mục nơi lưu kết quả xử lý video.

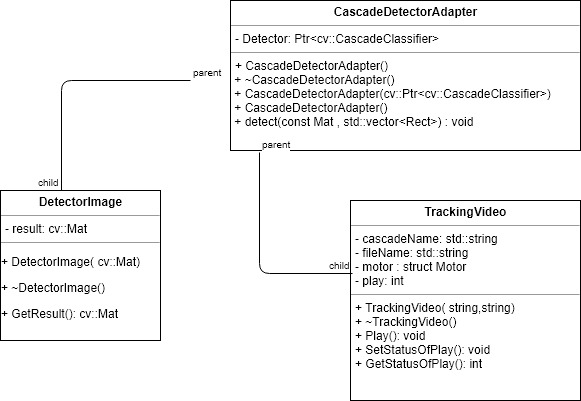
- Khu vực “Chọn”:

+ Nút “Tiếp tục”: Bắt đầu phân tích video được chọn.

+ Nút “Dừng”: Tạm dừng video.

- Nút “Thoát” : Đóng chương trình.

## 4.5 Thiết kế các lớp chính



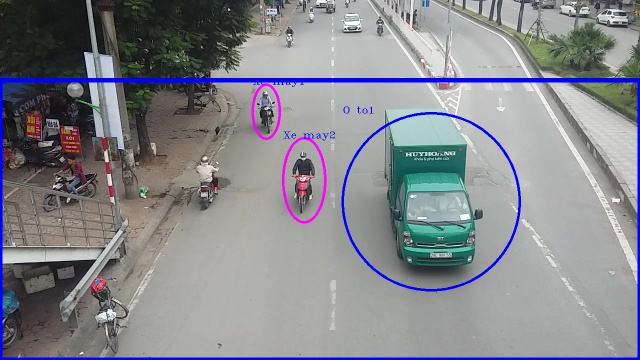
Hình 4.6 Các lớp chính trong chương trình

## 4.6 Kiểm thử

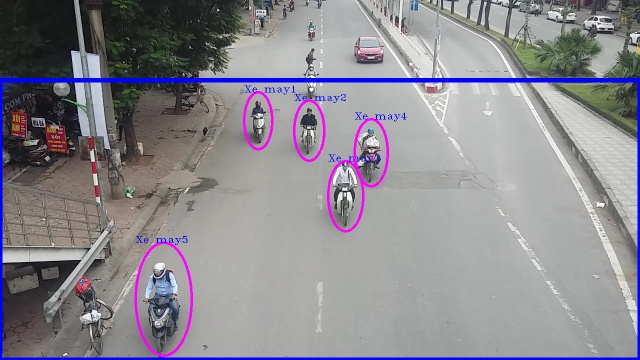
Tiến hành kiểm thử chương trình trong điều kiện giao thông bình thường tại một nút giao thông. Điều kiện ánh sáng ban ngày bình thường. Góc đặt camera hướng thẳng vào trục đường.

### 4.6.1 Tính năng phân loại phương tiện giao thông trong ảnh

Kết quả đạt được

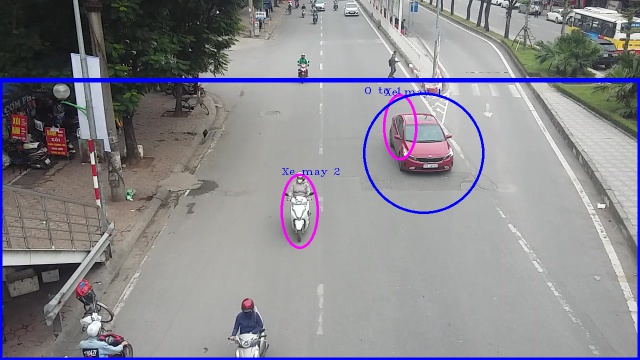


Hình 4.7.a Kết quả phân loại



Hình 4.7.b Kết quả phân loại

Chương trình đã đạt được kết quả khả quan cho mục tiêu phân loại ô tô và xe máy. Trong quá trình thực hiện kiểm thử. Em đã phát hiện một số trường hợp bộ phân loại xe máy nhận nhầm phần gương , đèn trước của ô tô là xe máy.



Hình 4.8.a Các vùng nhận dạng bị trùng lặp

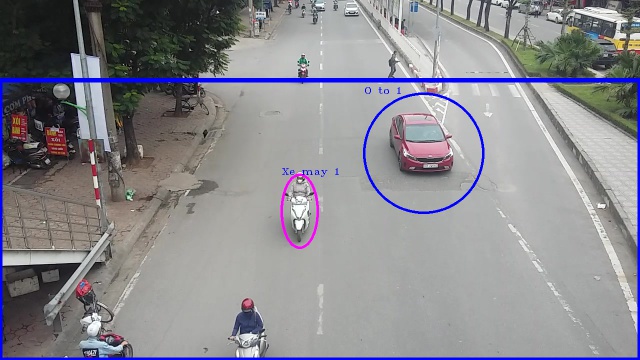


Hình 4.8.b Các vùng nhận dạng bị trùng lặp

Giải pháp cho tình trạng này được em đưa ra đó là:

1. Giới hạn kích thước phân loại của xe máy. Cụ thể, trong vùng nhận diện kích thước của xe máy không thể lớn hơn ô tô.
2. Kiểm tra sự trùng lặp giữa các vùng tìm thấy đối tượng. Nếu có, vùng nhận diện nhỏ hơn sẽ được loại bỏ.

Kết quả nhận được sau khi áp dụng giải pháp

 Hình 4.9.a Loại bỏ trùng lặp

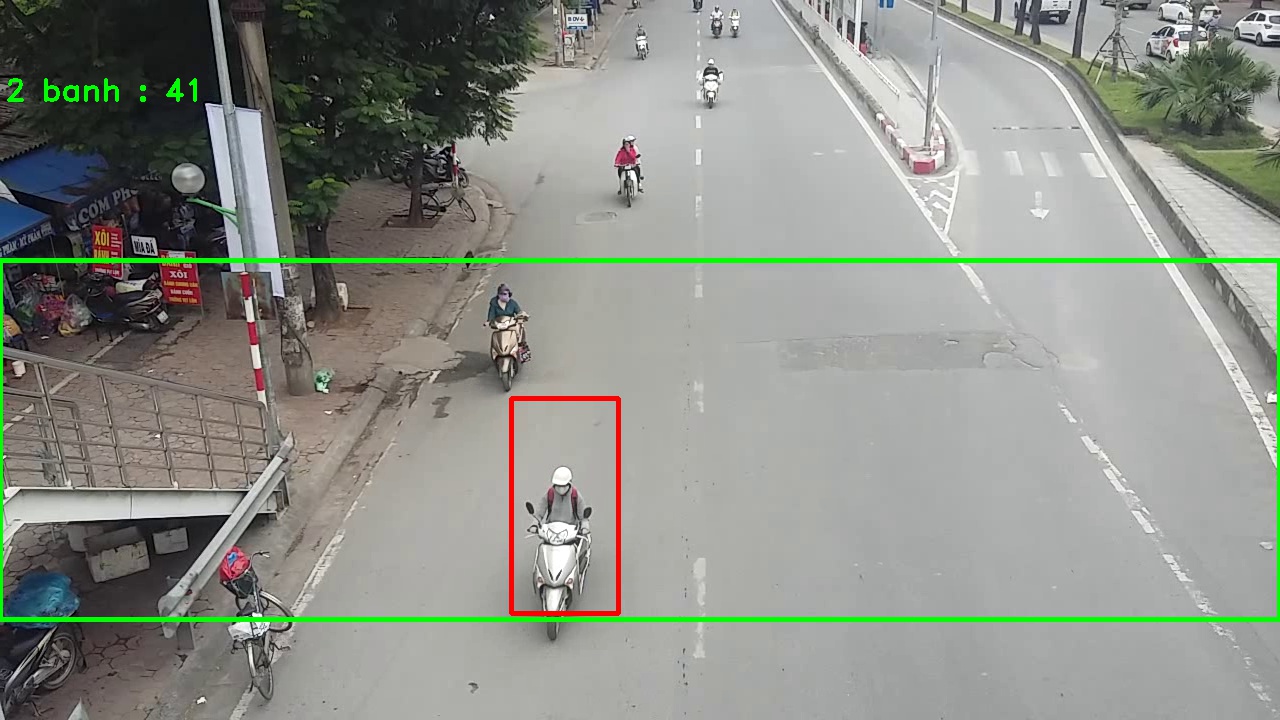
 Hình 4.9.b Loại bỏ trùng lặp

### 4.6.2 Tính năng phân loại và đếm xe máy trong video

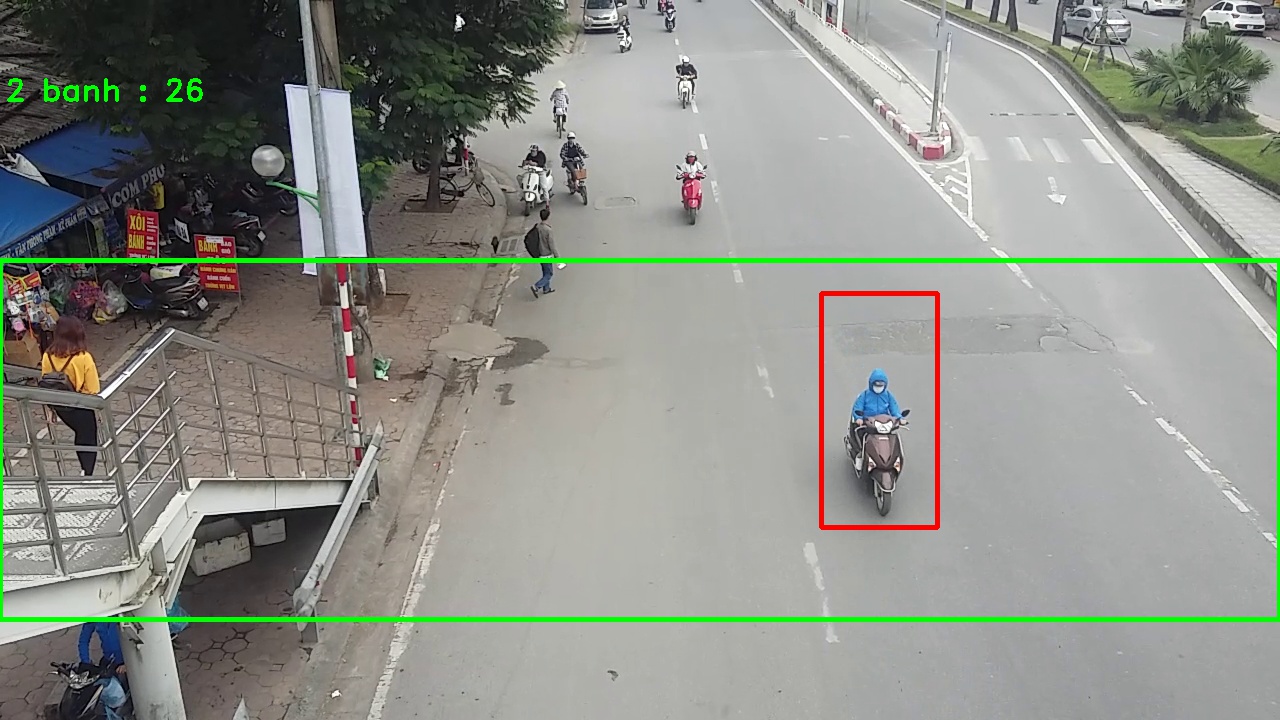
So sánh giữa 2 cách đếm : cách đếm bằng mắt thường và cách đếm bằng chương trình. Với giả thiết cách đếm bằng mắt thường là chính xác 100%. Thực hiện trên 2 video. Mỗi video có dộ dài khoảng 1 phút. Chương trình sẽ đếm các xe máy đi qua vùng nhận diện.

*Bảng 4‑1 Kết quả thử nghiệm đếm xe máy*

|  |  |  |
| --- | --- | --- |
| Số xe máy | Theo cách đếm bằng mắt thường | Theo kết quả của chương trình |
| Video 1 | 53 | 41 |
| Video 2 | 31 | 26 |



Hình 4.10 Kết quả đếm xe máy trong video 1



Hình 4.11 Kết quả đếm xe máy trong video 2

Kết quả phụ thuộc vào các yếu tố:

1. Tốc độ phương tiện.
2. Mật độ tham gia giao thông.

# KẾT LUẬN

1. Một số kết quả đạt được

Qua quá trình thực hiện khóa luận, với những nỗ lực hết mình, em đã nắm bắt được một số kỹ thuật cơ bản trong kỹ thuật thị giác máy cũng như một số khái niệm cơ bản về học máy và đã xây dựng được một ứng dụng đơn giản có thể phân biệt được phương tiện giao thông xe máy và ô tô trong . Ứng dụng cũng có thể đếm được lưu lượng xe máy lưu thông trên một số đoạn đường có mật độ giao thông vừa phải, không quá đông đúc. Tuy nhiên đến thời điểm hiện tại, ứng dụng vẫn chưa phong phú, đặc sắc; cần thời gian để phát triển hơn nữa.

Qua quá trình thực hiện khóa luận, bản thân em cũng tích lũy được nhiều kiến thức và kinh nghiệm giúp ích cho em trong quá trình làm việc sau này.

2. Hướng phát triển ứng dụng

Trong tương lai, ứng dụng sẽ mở rộng về tính năng và chức năng. Ví dụ như cảnh báo đi sai làn đường, phát hiện vi phạm luật giao thông, thống kê lưu lượng giao thông giúp pháttriển cơ sở hạ tầng giao thông.

Do thời gian có hạn và kinh nghiệm thực tế còn thiếu nên khóa luận không tránh được những sai sót, em rất mong nhận được sự góp ý của các thầy cô và các bạn để hoàn thiện hơn.

Em xin chân thành cảm ơn.

**TÀI LIỆU THAM KHẢO**

[1] Nguyễn Hoài Anh, bài giảng Phân tích thiết kế hệ thống, Học viện Kỹ thuật Quân sự, 2017.

[2] Lương Mạnh Bá, *“Nhập Môn Xử Lý Ảnh Số”* , NXB Khoa Học Kỹ Thuật 2007).

[3] Rafael C. Gonzalez, Richard E. Woods, *“Digital Image Processing Third Edition”*.

[4] Joseph Howse, Steven Puttemans, Quan Hua, Utkarsh Sinha , *“OpenCV 3 Blueprints : Expand your knowledge of computer vision by building amazing projects with OpenCV 3”.*

[5] Các website: Wikipedia “Q&A Site”, google.com.