

HỌC VIỆN HÀNG KHÔNG VIỆT NAM
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO TIỂU LUẬN TRÍ TUỆ NHÂN TẠO

XÂY DỰNG HỆ THỐNG QUẢN LÍ BIỂN SỐ XE

HỌC KỲ 1 – NĂM HỌC: 2023 - 2024

MÃ LỚP HỌC PHẦN: 010100085801

Giảng viên hướng dẫn: TS. Bùi Mạnh Quân

Nhóm sinh viên thực hiện: Trần Minh Tiến 2154810090

Lê Dương Bảo Ngọc 2154810098

Lê Trọng Huy 2154810055

TP. HCM, tháng 11 năm 2023

MỤC LỤC

| | Trang |
|---|-----------|
| Chương 1 GIỚI THIỆU..... | 5 |
| 1.1 Lí do chọn đề tài..... | 5 |
| 1.2 Đối tượng và mục tiêu nghiên cứu đề tài..... | 5 |
| 1.3 Phương pháp nghiên cứu của đề tài..... | 5 |
| Chương 2 CÔNG NGHỆ ỨNG DỤNG / CƠ SỞ LÝ THUYẾT..... | 6 |
| 2.1 PyQt5 và Qt Designer:..... | 6 |
| 2.1.1 PyQt: | 6 |
| 2.1.2 Qt Designer: | 7 |
| 2.2 Sơ lược về ngôn ngữ lập trình Python..... | 8 |
| 2.2.1 Tính năng của Python:..... | 8 |
| 2.2.2 Công cụ lập trình Python-Pycharm: | 8 |
| 2.3 Trí tuệ nhân tạo | 8 |
| 2.3.1 Khái niệm | 8 |
| 2.3.2 Ứng dụng của AI trong thực tế :..... | 9 |
| 2.4 Tổng quan bài toán hệ thống biển số xe | 9 |
| 2.4.1 Khái niệm biển số xe..... | 9 |
| 2.4.2 Xử lý ảnh và Open..... | 10 |
| 2.4.3 Hướng giải quyết..... | 10 |
| Chương 3 SẢN PHẨM ĐỒ ÁN..... | 11 |
| 3.1 Phân tích yêu cầu | 11 |
| 3.1.1 Yêu cầu chức năng..... | 11 |
| 3.1.2 Yêu cầu hệ thống..... | 11 |
| 3.2 Hướng giải quyết tách biển số..... | 12 |
| 3.2.1 Chuyển ảnh sang xám | 13 |
| 3.2.2 Tăng độ tương phản | 14 |
| 3.2.3 Giảm nhiễu bằng bộ lọc Gaussian | 14 |
| 3.2.4 Phát hiện cạnh Canny..... | 14 |
| 3.2.5 Vẽ contour | 14 |
| 3.2.6 Phân đoạn kí tự..... | 15 |

| | |
|---------------------------------------|-------------------------------------|
| 3.2.6.1 Xoay biến số..... | Error! Bookmark not defined. |
| 3.2.6.2 Tìm vùng đối tượng..... | 15 |
| 3.2.6.3 Tìm và tách kí tự..... | 16 |
| 3.2.7 Nhận diện kí tự..... | 16 |
| 3.2.7.1 Thuật toán KNN..... | 16 |
| 3.2.7.2 Hướng giải quyết..... | 17 |
| 3.3 Demo chương trình | 18 |
| 3.3.1 Hình ảnh Demo chương trình..... | 18 |
| 3.3.2 Code của chương trình | 20 |
| Chương 4 KẾT LUẬN | 23 |
| 4.1 Kết quả đạt được:..... | 23 |
| 4.2 Hạn chế:..... | 23 |
| 4.3 Hướng phát triển: | 23 |
| TÀI LIỆU THAM KHẢO | 24 |

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ĐÁNH GIÁ:

ĐIỂM SỐ: -----

ĐIỂM CHỮ: -----

Chữ ký giảng viên

LỜI NÓI ĐẦU

Số lượng phương tiện tham gia giao thông trên đường hiện nay rất lớn, đặt ra thách thức trong việc giữ gìn trật tự, an toàn giao thông. Với số lượng phương tiện ngày càng tăng, việc quản lý biển số xe ngày càng trở nên phức tạp. Vì vậy, hệ thống quản lý biển số xe là một giải pháp hiệu quả nhằm nâng cao hiệu quả và an toàn trong quản lý giao thông đô thị.

Ngoài ra, hệ thống cũng sẽ đóng vai trò quan trọng trong việc kiểm soát an ninh, giúp chống tội phạm giao thông và đảm bảo an toàn cho cộng đồng. Chúng ta sẽ thấy những cải thiện đáng kể trong việc giám sát và truy tố hành vi bất hợp pháp cũng như giải quyết nhanh chóng các vấn đề an toàn giao thông. Vì thế, chúng em đã viết một chương trình : “XÂY DỰNG HỆ THỐNG QUẢN LÝ BIỂN SỐ XE”

Song vì thời gian có hạn, và còn thiếu nhiều kinh nghiệm và kỹ năng còn hạn chế. Nên chắc chắn không tránh khỏi những thiếu sót. Vì vậy chúng em rất mong nhận được nhiều ý kiến đóng góp từ thầy cô và bạn bè để chúng em hoàn thành tốt đề tài này. Cảm ơn thầy và các bạn.

Chương 1 GIỚI THIỆU

1.1 Lí do chọn đề tài

Sự phát triển mạnh mẽ của công nghệ thông tin trong những năm gần đây đã mang lại những thay đổi to lớn trong đời sống con người. Nó ngày càng khẳng định vai trò của mình trong sự phát triển chung của xã hội: hầu hết mọi việc con người làm được giờ đây đều có thể được thực hiện bằng máy móc.

Hệ thống biển số xe đóng vai trò quan trọng trong việc nhận dạng và theo dõi phương tiện. Tự động hóa và hiện đại hóa quản lý biển số xe mang lại nhiều lợi ích cho cả cơ quan và cộng đồng. Đặc biệt, việc sử dụng công nghệ trong quản lý biển số xe có thể giúp cải thiện hiệu suất đồng thời cải thiện trật tự an toàn giao thông. Vì thế, chúng em đã viết một chương trình: “XÂY DỰNG HỆ THỐNG QUẢN LÝ BIỂN SỐ XE”. Để bắt kịp xu hướng hiện tại, chúng em nghiên cứu và giải quyết vấn đề nhận dạng này bằng ngôn ngữ lập trình Python. Mục tiêu dự định của chúng em là hiểu thêm về vấn đề phát hiện và áp dụng nó vào thực tế.

1.2 Đối tượng và mục tiêu nghiên cứu đề tài

- Xây dựng được chương trình giúp người dùng nhận dạng được các biển số thông qua các hình ảnh hoặc camera đầu vào.
- Chương trình có thể nhận dạng được số xe một cách chính xác và in nó ra màn hình.
- Hiểu rõ hơn về các thuật toán nhận dạng và cách hoạt động nhằm nâng cao hiểu biết về công nghệ hiện nay

1.3 Phương pháp nghiên cứu của đề tài

- Tìm hiểu về ngôn ngữ lập trình Python
- Tìm hiểu về công cụ lập trình Pycharm
- Tìm hiểu về các thư viện hỗ trợ lập trình
- Tham khảo các chức năng, các điểm ưu nhược của các ứng dụng đã có, từ đó quyết định các ứng dụng và hướng phát triển

- Tham khảo và lập trình giao diện cho ứng dụng
- Phân tích các chức năng và yêu cầu cần có
- Nghiên cứu về cách thức hoạt động của bài toán nhận dạng
- Nghiên cứu và hiểu được thuật toán của bài toán nhận dạng biển số xe
- Kiểm thử phần mềm, sửa chữa và phát triển thêm các ý tưởng mới
- Xây dựng báo cáo hoàn chỉnh

Chương 2 CÔNG NGHỆ ỨNG DỤNG / CƠ SỞ LÝ THUYẾT

2.1 PyQt5 và Qt Designer:

2.1.1 PyQt:

PyQt là Python interface của Qt, kết hợp của ngôn ngữ lập trình Python và thư viện Qt, là một thư viện bao gồm các thành phần giao diện điều khiển (widgets , graphical control elements).

PyQt được phát triển bởi Riverbank Computing Limited.

Các class của PyQt5 được chia thành các module, bao gồm :

- QtCore : là module bao gồm phần lõi không thuộc chức năng GUI, ví dụ dùng để làm việc với thời gian, file và thư mục, các loại dữ liệu, streams, URLs, mime type, threads hoặc processes.
- QtGui : bao gồm các class dùng cho việc lập trình giao diện (windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.
- QtWidgets : bao gồm các class cho widget, ví dụ : button, hộp thoại, ... được sử dụng để tạo nên giao diện người dùng cơ bản nhất.
- QtMultimedia : thư viện cho việc sử dụng âm thanh, hình ảnh, camera,...

- QtBluetooth : bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm.
- QtNetwork : bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client , server hỗ trợ việc lập trình mạng.
- QtPositioning : bao gồm các class giúp việc hỗ trợ xác định vị.
- Enginio : module giúp các client truy cập các Cloud Services của Qt.
- QtWebSockets : cung cấp các công cụ cho WebSocket protocol.
- QtWebKit : cung cấp các class dùng cho làm việc với các trình duyệt Web , dựa trên thư viện WebKit2.
- QtWebKitWidgets : các widget cho WebKit.
- QtXml : các class dùng cho làm việc với XML file.
- QtSvg : dùng cho hiển thị các thành phần của SVG file.
- QSql : cung cấp các class dùng cho việc làm việc với dữ liệu.
- QTest : cung cấp các công cụ cho phép test các đơn vị của ứng dụng với PyQt5.

2.1.2 Qt Designer:

Qt Designer là một công cụ Qt để thiết kế và xây dựng giao diện người dùng đồ họa (GUI) với Qt Widget. Bạn có thể soạn và tùy chỉnh các cửa sổ hoặc hộp thoại của mình theo cách bạn sẽ thấy (What you see is what you get – Những gì bạn đang thấy là những gì bạn sẽ nhận được) và kiểm tra chúng bằng các kiểu và độ phân giải khác nhau.

Các tiện ích và biểu mẫu được tạo bằng Qt Designer tích hợp liền mạch với mã được lập trình, sử dụng cơ chế vị trí và tín hiệu của Qt để bạn có thể dễ dàng gán hành vi cho các phần tử đồ họa. Tất cả các thuộc tính được đặt trong Qt Designer có thể được thay đổi động trong mã. Hơn nữa, các tính năng như quảng cáo widget và plugin tùy chỉnh cho phép bạn sử dụng các thành phần của riêng mình với Qt Designer.

2.2 Sơ lược về ngôn ngữ lập trình Python

Phiên bản sử dụng trong đồ án: Python 3.7.

2.2.1 Tính năng của Python:

- Miễn phí với mã nguồn mở.
- Có thể chạy python trên hầu hết các nền tảng khác nhau.
- Ngôn ngữ thông dịch cấp cao.
- Thư viện tiêu chuẩn lớn.
- Hướng đối tượng.

2.2.2 Công cụ lập trình Python-Pycharm:

Phiên bản sử dụng trong đồ án: PyCharm 2023.2.3 (Community Edition)
PyCharm là môi trường phát triển tích hợp đa nền tảng (IDE) được phát triển bởi Jet Brains và được thiết kế đặc biệt cho Python. PyCharm có mặt trên cả 3 nền tảng Windows, Linux và Mac OS.

2.3 Trí tuệ nhân tạo

2.3.1 Khái niệm

Trí tuệ nhân tạo (AI) là lĩnh vực của khoa học máy tính tập trung vào phát triển các hệ thống có khả năng thực hiện các nhiệm vụ yêu cầu sự hiểu biết, tư duy, và quyết định, tương tự như khả năng trí óc của con người. Nó liên quan đến việc thiết kế và phát triển máy tính có khả năng tự học, tư duy, và thích ứng với môi trường mới.

Hiện nay, công nghệ AI có 3 hướng chính:

- Xử lý hình ảnh.
- Xử lý ngôn ngữ tự nhiên.
- Xử lý tín hiệu âm thanh.

Trong đồ án này, đề tài chúng em thực hiện theo hướng xử lý hình ảnh.

2.3.2 Ứng dụng của AI trong thực tế :

- Hệ thống nhận diện khuôn mặt thông qua các camera được gắn tại sân bay và các tòa nhà.
- Các trợ lý ảo (như Siri, Google Assistant, Alexa) có khả năng nghe, hiểu, trả lời và làm việc cho mình
- Những ứng dụng trong y sinh, dùng AI để chẩn đoán bệnh dựa trên phim chụp X-quang, X-ray và MRI.

2.4 Tổng quan bài toán hệ thống biển số xe

2.4.1 Khái niệm biển số xe

Tại Việt Nam, biển kiểm soát xe cơ giới, hay còn được gọi là biển kiểm soát hoặc biển số xe, là một tấm biển đặt trên mỗi chiếc xe cơ giới. Các biển này được cấp bởi cơ quan công an (hoặc Bộ Quốc phòng đối với xe quân sự) khi xe mới được mua hoặc khi xe được chuyển nhượng. Chúng thường được làm từ hợp kim nhôm sắt và có dạng hình chữ nhật hoặc hơi vuông. Trên biển số xe, có in số và chữ, thể hiện vùng và địa phương quản lý, thông tin cụ thể về người chủ hoặc đơn vị mua xe, cũng như thời điểm mua xe, phục vụ cho công tác an ninh.

Về tiêu chuẩn kích thước, mỗi quốc gia thường có quy định riêng. Tại Việt Nam, tỷ lệ kích thước giữa các biển số thường gần như nhau. Có hai loại biển số xe: loại dài (cao 110 mm, dài 470 mm) và loại ngắn (cao 200 mm, dài 280 mm). Tỷ lệ cao/rộng được giới hạn trong khoảng $3.5 \leq \text{cao/rộng} \leq 6.5$ (biển một hàng) và $0.8 \leq \text{cao/rộng} \leq 1.5$ (biển hai hàng).

Số lượng ký tự trên biển số xe nằm trong khoảng từ 7 đến 9. Chiều cao của chữ và số là 80mm, chiều rộng là 40mm. Những đặc điểm này giúp thiết lập các thông số và điều khiển để lọc chọn những đối tượng phù hợp trong quá trình xử lý thông tin về biển số xe.

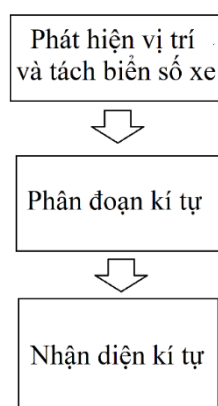
2.4.2 Xử lý ảnh và Open

Xử lý hình ảnh là một phân ngành xử lý tín hiệu số trong đó tín hiệu được xử lý là hình ảnh. Đây là một lĩnh vực khoa học mới được phát triển trong những năm gần đây. Xử lý ảnh bao gồm bốn lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy xuất ảnh. Sự phát triển của xử lý ảnh mang lại nhiều lợi ích cho đời sống con người. Ngày nay, xử lý ảnh được sử dụng rộng rãi trong cuộc sống như Photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết tay, xử lý ảnh thiên văn và ảnh y tế quốc tế.... OpenCV (Open Computer Vision) là thư viện mã nguồn mở hàng đầu về thị giác máy tính, học máy và xử lý hình ảnh. OpenCV được viết bằng C/C++ nên có tốc độ tính toán rất nhanh và có thể sử dụng với các ứng dụng thời gian thực. OpenCV có giao diện cho C/C++, Python Java và do đó hỗ trợ Windows, Linux, MacO cũng như Android, iOS. OpenCV có cộng đồng hơn 47. người dùng và số lượt tải xuống vượt quá 6 triệu lần.

2.4.3 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 đề án này chúng 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 2.1 Các bước chính trong nhận dạng biển số xe

Chương 3

SẢN PHẨM ĐỒ ÁN

3.1 Phân tích yêu cầu

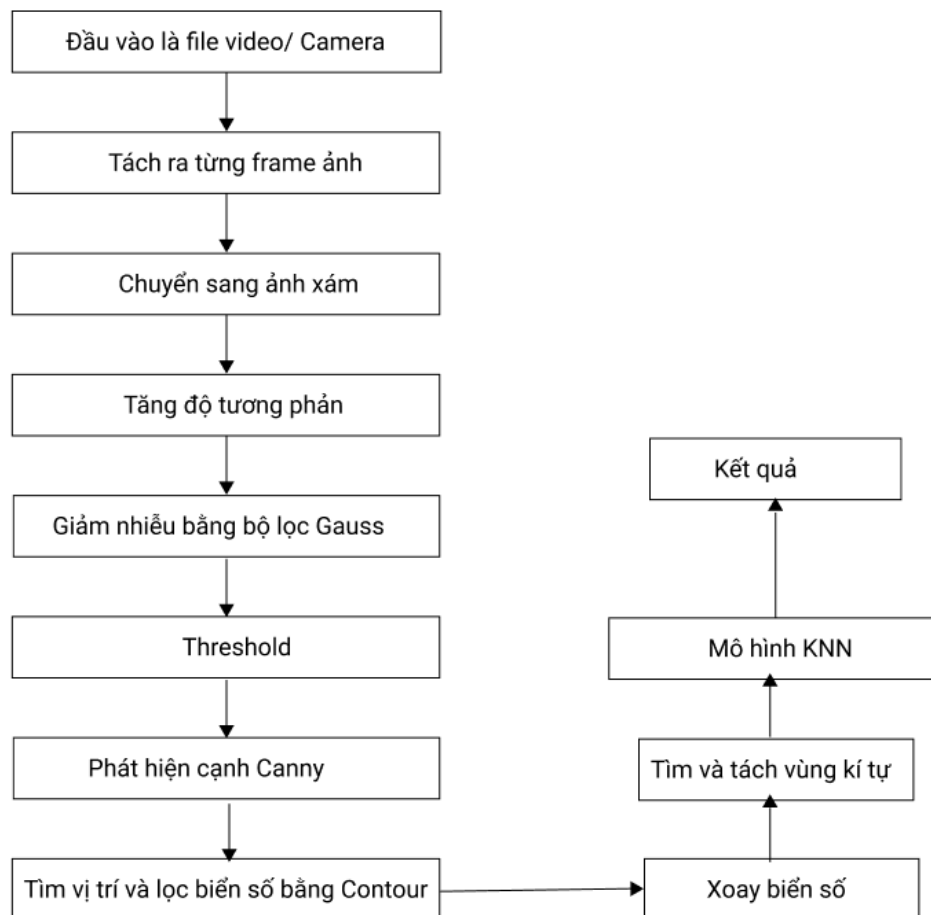
3.1.1 Yêu cầu chức năng

- Có thể đọc dữ liệu ảnh đầu vào.
- Có thể đọc dữ liệu từ video.
- Tách được biển số từ video và ảnh đầu vào.
- Đọc biển số đã tách.
- Hiện thị biển số nhận dạng được ra màn hình.
- Có giao diện dễ sử dụng.

3.1.2 Yêu cầu hệ thống

- Chương trình chạy trên hệ điều hành windows.
- Chương trình được viết trên Pycharm.
- Sử dụng python 3.7

3.2 Hướng giải quyết tách biển số



Hình 3.1 Lưu đồ nhận dạng biển số xe

Trước hết từ video hoặc ảnh đầu vào, chúng ta sẽ thực hiện việc cắt thành từng frame ảnh để bắt đầu quá trình xử lý và tách biển số. Sau đó, ảnh sẽ được chuyển đổi sang dạng màu xám. Tiếp theo, để làm nổi bật biển số trên nền phong, chúng ta sẽ tăng độ tương phản bằng cách sử dụng hai phép toán hình thái học Top Hat và Black Hat. Điều này giúp tạo ra sự nổi bật đặc biệt trên biển số, làm cho quá trình xử lý nhị phân sau này trở nên hiệu quả hơn. Sau đó, chúng ta sẽ giảm nhiễu bằng cách sử dụng bộ lọc Gauss để loại bỏ chi tiết nhiễu có thể gây ảnh hưởng đến quá trình nhận diện, đồng thời tăng tốc độ xử lý.

Việc lấy ngưỡng sẽ hỗ trợ chúng ta trong việc phân biệt thông tin giữa biển số và nền ảnh. Ở đây, tôi lựa chọn sử dụng ngưỡng động (Adaptive Threshold). Tiếp theo, chúng ta sử dụng thuật toán phát hiện cạnh Canny để

trích xuất những chi tiết cạnh của biên số. Trong quá trình xử lý, có thể có những chi tiết nhiễu gây nhầm lẫn giữa biên số và các đối tượng khác. Do đó, việc lọc cuối cùng bằng cách áp dụng các tiêu chí như tỉ lệ cao/rộng hay diện tích sẽ giúp chúng ta xác định chính xác biên số.

Cuối cùng, chúng ta sẽ xác định vị trí của biên số trong ảnh bằng cách vẽ Contour bao quanh nó. Điều này giúp chúng ta có cái nhìn rõ ràng về đường viền của biên số trong bức ảnh. Vì có quá nhiều đường bao quanh các vật không phải là biên số nên chúng ta sẽ áp dụng những đặc trưng riêng về tỉ lệ cao/ rộng, diện tích trong khung hình cố định để lọc ra đúng biên số.

Đầu tiên ta làm xấp xỉ contour thành một hình đa giác và chỉ lấy những đa giác nào chỉ có 4 cạnh. Nghĩa là lúc xấp xỉ contour bộ nhớ chỉ ghi nhớ vị trí các đỉnh của đa giác đó thành một mảng. Số cạnh của đa giác sẽ bằng số đỉnh và bằng chiều dài của mảng đó.

Tiếp theo ta tính toán tỉ lệ cao/rộng và diện tích của biên số phù hợp, sau đó ta lưu tất cả những biên số có trong hình dưới dạng tọa độ các đỉnh

Từ đây, ta cắt hình ảnh biên số từ các tọa độ vị trí đã biết để phục vụ cho mục đích tiếp theo “Tách các kí tự trong biên số”. Lưu ý ở đây ta cắt từ ảnh nhị phân luôn để máy tính xử lý nhanh hơn, tốn ít thời gian hơn.

3.2.1 Chuyển ảnh sang xám

Ảnh xám, hay còn gọi là Gray Scale, đơn giản là một dạng hình ảnh trong đó các màu sắc được biểu diễn dưới dạng các sắc thái của màu xám, với 256 cấp độ xám biến thiên từ màu đen đến màu trắng. Giá trị xám của mỗi điểm ảnh nằm trong khoảng từ 0 đến 255, cần 8 bits hay 1 byte để biểu diễn. Sự khác biệt giữa ảnh xám và các loại ảnh khác là ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Trong khi ảnh màu thông thường cung cấp 3 trường thông tin cho mỗi pixel, ảnh xám chỉ có 1 trường thông tin. Việc giảm khối lượng thông tin này không chỉ giúp tăng tốc độ xử lý mà còn đơn giản hóa quá trình giải thuật mà vẫn đảm bảo các tác vụ cần thiết.

Chúng em quyết định chuyển đổi ảnh xám từ không gian màu HSV thay vì RGB. Trong không gian màu HSV, có ba giá trị chính: Hue (Vùng màu), Saturation (Độ bão hòa), và Value (Cường độ sáng). Việc sử dụng không gian màu HSV có ưu điểm là thích nghi tốt hơn với sự thay đổi ánh sáng từ môi trường xung quanh. Khi thực hiện chuyển đổi, ảnh xám sẽ được biểu diễn dưới dạng ma trận chứa các giá trị cường độ sáng, được tách ra từ không gian màu HSV.

3.2.2 Tăng độ tương phản

Để làm tăng độ tương phản của biên số, chúng em chủ yếu sử dụng hai phép toán hình thái học là Top Hat và Black Hat. Ý tưởng chung là ảnh đầu ra sẽ được tạo bằng cách cộng thêm ảnh gốc với kết quả của phép Top Hat, sau đó trừ đi kết quả của phép Black Hat. Nhờ đó những chi tiết đã được làm sáng trước đó sẽ trở nên rõ nét hơn, trong khi những chi tiết tối lại trở nên tối hơn. Từ đó giúp làm tăng độ tương phản của biên số.

3.2.3 Giảm nhiễu bằng bộ lọc Gaussian

Bộ lọc Gaussian được coi là bộ lọc hiệu quả nhất, được thực hiện bằng cách thực hiện phép nhân chập giữa ảnh đầu vào với một ma trận lọc Gaussian, sau đó cộng chúng lại để tạo ra ảnh đầu ra.

3.2.4 Phát hiện cạnh Canny

Trong hình ảnh, thường xuất hiện các thành phần như vùng tròn, góc/cạnh và nhiễu. Các cạnh trong ảnh đóng vai trò quan trọng, thường là một phần của đối tượng trong bức ảnh. Vì vậy, để phát hiện cạnh trong ảnh, có nhiều giải thuật khác nhau như toán tử Sobel, toán tử Prewitt, Zero crossing, và nhiều giải thuật khác. Tuy nhiên, trong trường hợp này, chúng em lựa chọn giải thuật Canny vì phương pháp này hơn hẳn so với các phương pháp khác do ít bị ảnh hưởng bởi nhiễu và có khả năng phát hiện các biên yếu một cách hiệu quả hơn.

3.2.5 Vẽ contour

Có thể hiểu Contour là tập hợp các điểm tạo thành đường cong kín bao quanh một đối tượng nào đó. Thường dùng để xác định vị trí, đặc điểm của đối tượng.

Đầu tiên ta làm xấp xỉ contour thành một hình đa giác và chỉ lấy những đa giác nào chỉ có 4 cạnh. Nghĩa là lúc xấp xỉ contour bộ nhớ chỉ ghi nhớ vị trí các đỉnh của đa giác đó thành một mảng. Số cạnh của đa giác sẽ bằng số đỉnh và bằng chiều dài của mảng đó.

Tiếp theo ta tính toán tỉ lệ cao/rộng và diện tích của biến số phù hợp, sau đó ta lưu tất cả những biến số có trong hình dưới dạng tọa độ các đỉnh.

Từ đây, ta cắt hình ảnh biến số từ các tọa độ vị trí đã biết để phục vụ cho mục đích tiếp theo “Tách các kí tự trong biến số”. Lưu ý ở đây ta cắt từ ảnh nhị phân luôn để máy tính xử lý nhanh hơn, tốn ít thời gian hơn.

3.2.6 Phân đoạn kí tự

3.2.6.1 Xoay biến số

Khi chụp ảnh đầu vào, không phải lúc nào biến số cũng ở chính diện, có thể bị méo sang trái, sang phải, nghiêng góc dẫn đến nếu cứ sử dụng ảnh biến 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,...

3.2.6.2 Tìm vùng đối tượng

Từ ảnh nhị phân, chúng ta tiếp tục quá trình tìm contour cho các kí tự (là phần màu trắng). Sau đó vẽ hình chữ nhật bao quanh các kí tự này. Tuy nhiên, quá trình tìm contour có thể bị nhiễu, dẫn đến việc máy xử lý nhầm lẫn và nhận diện những hình ảnh không phải là kí tự.

Để khắc phục vấn đề này, chúng em áp dụng các đặc điểm liên quan đến tỉ lệ chiều cao/rộng của kí tự, diện tích của kí tự so với biến số. Qua so sánh với ảnh nhị phân, chúng ta loại bỏ các đường nhiễu như đường viền biến số, dấu gạch, dấu chấm, và các chi tiết không liên quan khác. Sau khi áp dụng các điều kiện này, chúng ta sẽ vẽ các hình chữ nhật màu xanh bao quanh các kí tự, tạo nên một kết quả cuối cùng chính xác và chứa ít nhiễu nhất có thể.

3.2.6.3 Tìm và tách kí tự

Sau khi đã xác định và nhận dạng từng kí tự thông qua hình chữ nhật, và cũng đã lưu trữ tọa độ vị trí của 4 đỉnh của mỗi hình chữ nhật, chúng ta có thể tiến hành cắt hình ảnh kí tự từ ảnh gốc. Điều này được thực hiện để chuẩn bị dữ liệu cho giai đoạn tiếp theo, tức là "Nhận diện kí tự".

3.2.7 Nhận diện kí tự

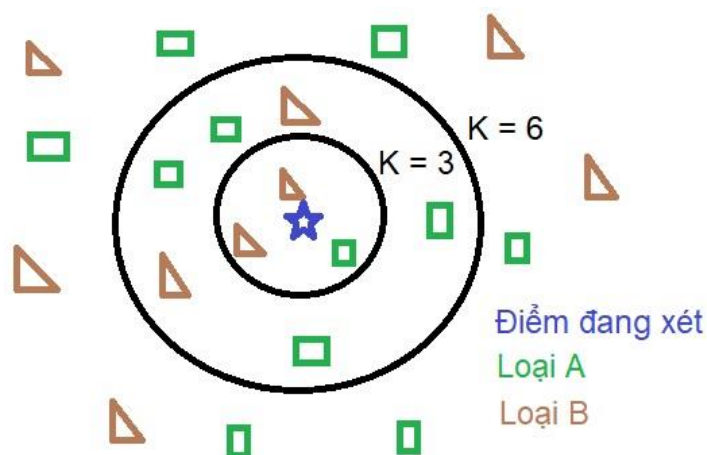
3.2.7.1 Thuật toán KNN

K-Nearest Neighbors (KNN) là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning, có thể được áp dụng cho cả bài toán phân loại và hồi quy. Một ví dụ điển hình là khi áp dụng KNN để phân biệt các giống chó khác nhau. Bằng cách so sánh các đặc điểm như mắt, lông, tai vv. với các giống chó đã biết trước, chúng ta có thể xác định giống chó mà điểm dữ liệu đang xét thuộc về.

Quy trình hoạt động của thuật toán KNN bao gồm các bước chính như sau:

1. Xác định Tham số K (số láng giềng gần nhất): Chọn giá trị K, tức là số lượng láng giềng gần nhất mà thuật toán sẽ xem xét khi phân loại.
2. Tính Khoảng Cách: Đo lường khoảng cách từ điểm đang xét đến tất cả các điểm trong tập dữ liệu đã biết trước, sử dụng một độ đo khoảng cách như khoảng cách Euclidean.
3. Sắp Xếp Khoảng Cách: Sắp xếp các khoảng cách này theo thứ tự tăng dần để xác định K láng giềng gần nhất.

Phân Loại Hoặc Dự Đoán: Dựa trên nhãn của K láng giềng gần nhất, quyết định nhãn của điểm đang xét. Nếu một loại nhãn chiếm ưu thế trong K láng giềng, điểm đó sẽ được phân loại thuộc về loại đó.



Hình 3.2 Ví dụ về KNN

Việc xác định loại của điểm đang xét trong thuật toán KNN không chỉ phụ thuộc vào giá trị K mà còn liên quan đến hệ số trọng số khoảng cách. Người dùng có thể điều chỉnh những tham số này để đáp ứng yêu cầu của bài toán cụ thể.

Ví dụ, trong hình minh họa, nếu ta thiết lập $K = 3$, điểm đang xét sẽ được phân loại thuộc loại B. Ngược lại, nếu ta chọn $K = 6$, điểm đó sẽ được phân loại vào loại A. Ngoài ra, người dùng có thể thay đổi hệ số trọng số để ưu tiên những điểm gần hơn. Một cách khác là sử dụng $K = 1$ để đảm bảo kết quả đầu ra được tối ưu. Tuy nhiên, điều này có thể làm tăng độ nhạy của mô hình đối với nhiễu trong dữ liệu. Quá trình điều chỉnh các tham số này giúp định hình mô hình KNN để phù hợp với đặc điểm và yêu cầu cụ thể của bài toán.

3.2.7.2 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 biến 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 tạo tập dữ liệu (tập hình ảnh của các chữ số và kí tự) để train từ phần mềm paint. Ta viết các chữ số và kí tự (trừ kí tự O, I, J) với phong chữ “Biển số xe Việt Nam”,

có thể xoay các kí tự này lần lượt với các góc -5° , 5° , -10° , 10° . Kết quả có dạng như sau:

```

A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3
4 5 6 7 8 9

A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3
4 5 6 7 8 9

A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3
4 5 6 7 8 9

A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3
4 5 6 7 8 9

A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3
4 5 6 7 8 9

```

Hình 3.2 Tập dữ liệu kí tự

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 diện cho hình ảnh đó. Cuối cùng ta in biên số xe ra hình.

3.3 Demo chương trình

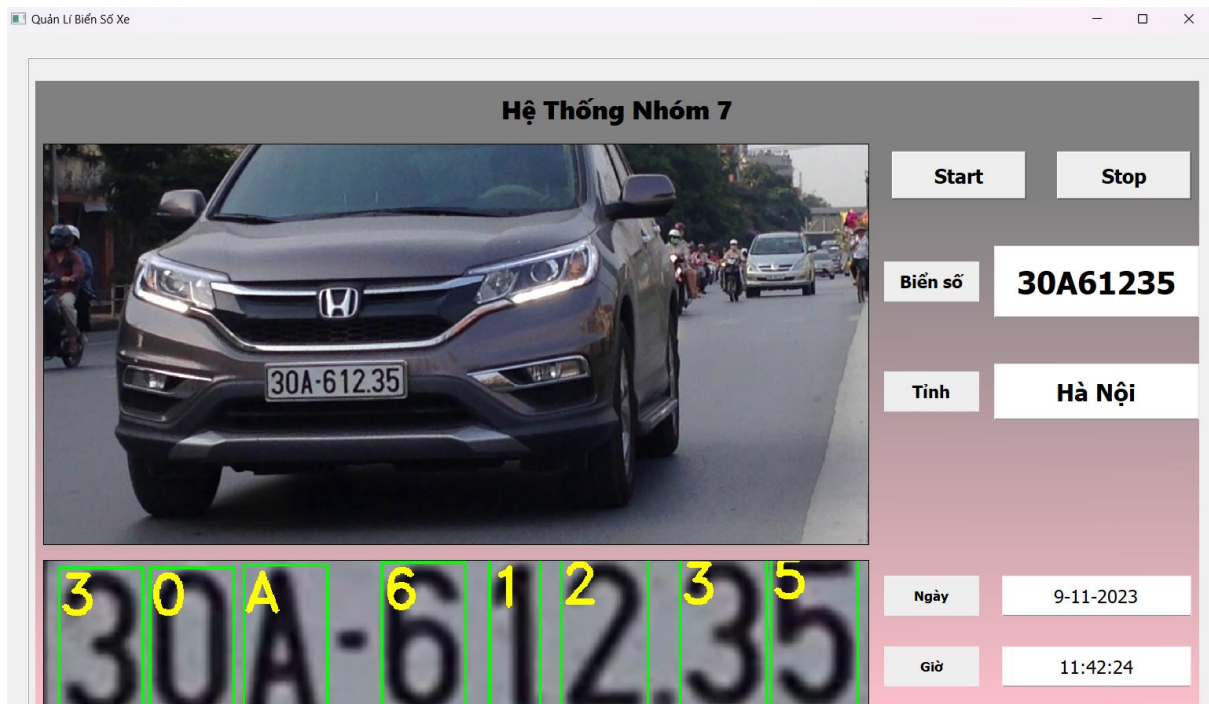
3.3.1 Hình ảnh Demo chương trình

Sau khi nhấn nút chọn ảnh từ thư mục tùy ý, ảnh sẽ hiện lên ở khung label góc trên bên phải. Tiếp theo ta nhấn nút nhận dạng để tiến hành nhận dạng biên số và in lên label chính giữa. Nút thông tin dùng để hiển thị thông tin của biên số (tính).



Hình 3.3 Ảnh demo nhận dạng biển số xe từ ảnh

Ta chọn ảnh từ thư mục của mình, sau khi bấm nhận dạng sẽ đọc biển số và ghi lên ô “Biển Số”. Khi bấm vào “Thông Tin”, sẽ hiện ra được tỉnh thành đăng kí xe đó.



Hình 3.4 Ảnh demo nhận dạng video

Ta nhấn nút bắt đầu để tiến hành chạy video và nhận dạng ảnh, các thông tin trích xuất từ ảnh sẽ hiện ra ở các label bên phải như: biển số, tỉnh,... Ta có thể nhấn nút dừng để dừng video và quá trình nhận dạng.

3.3.2 Code của chương trình

```
##### Tải lên mô hình KNN #####
npaClassifications = np.loadtxt("classificationS.txt", np.float32)
npaFlattenedImages = np.loadtxt("flattened_images.txt", np.float32)
# reshape numpy array to 1d, necessary to pass to call to train
npaClassifications = npaClassifications.reshape((npaClassifications.size, 1))
# instantiate KNN object
kNearest = cv2.ml.KNearest_create()
kNearest.train(npaFlattenedImages, cv2.ml.ROW_SAMPLE, npaClassifications)
#####

class MainWindow(QMainWindow, layout_image.Ui_Frame):
    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.setupUi(self)
        self.btn_chonanh.clicked.connect(self.loadImage)

        self.btn_nhandang.clicked.connect(self.imgae_license)
        self.btn_info.clicked.connect(self.info)

    def showtime(self):
        while True:
            QApplication.processEvents()
            dt = datetime.datetime.now()
            self.let_ngay.setText('%s-%s-%s' % (dt.day, dt.month, dt.year))
            self.let_gio.setText('%s:%s:%s' % (dt.hour, dt.minute, dt.second))

    def loadImage(self):
        self.img_path = QFileDialog.getOpenFileName(filter="Image (*.*)")[0]
        self.Ivehicle = cv2.imread(self.img_path)
```

Hình 3.5 Tải lên mô hình KNN và hiển thị ảnh lên

```
class MainWindow(QMainWindow, Ui_Frame):
    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.setupUi(self)
        self.btn_chonVideo.clicked.connect(self.start_capture_video)
        self.btn_nhandang.clicked.connect(self.stop_capture_video)
        self.thread = {}

    def closeEvent(self, event):
        self.stop_capture_video()

    def stop_capture_video(self):
        self.thread[1].stop()

    def showtime(self):
        while True:
            QApplication.processEvents()
            dt = datetime.datetime.now()
            self.let_ngay.setText('%s-%s-%s' % (dt.day, dt.month, dt.year))
            self.let_gio.setText('%s:%s:%s' % (dt.hour, dt.minute, dt.second))

    def start_capture_video(self):
        self.thread[1] = capture_video(index=1)
        self.thread[1].start()
        self.thread[1].signal.connect(self.show_webcam)
        self.thread[1].updateGui.connect(self.updateBs)
        self.thread[1].signal_roi.connect(self.updateRoi)
        self.thread[1].updateGui.connect(self.info)

    def updateBs(self, text):
```

Hình 3.6 Hàm showtime và hiển thị video

```
def info(self, text):
    in4 = self.let_bienso.text()
    in5 = int(in4[0:2])
    self.let_ten.setText('Nhóm 7')
    lang = {
        11: 'Cao Bằng', 12: 'Lạng Sơn', 14: 'Quảng Ninh', 15: 'Hải Phòng', 17: 'Thái Bình', 18: 'Nam Định',
        19: 'Phủ Thọ', 20: 'Thái Nguyên', 21: 'Yên Bái', 22: 'Tuyên Quang', 23: 'Hà Giang', 24: 'Lao Cai',
        25: 'Lai Châu', 26: 'Sơn La', 27: 'Điện Biên', 28: 'Hoà Bình', 29: 'Hà Nội', 30: 'HN', 31: 'Hà Nội',
        32: 'Hà Nội', 33: 'Hà Nội', 40: 'Hà Nội', 34: 'Hải Dương', 35: 'Ninh Bình', 36: 'Thanh Hóa', 37: 'Nghệ An',
        38: 'Hà Tĩnh', 43: 'Đà Nẵng', 47: 'Đak Lak', 48: 'Đắk Nông', 49: 'Lâm Đồng', 50: 'HCM', 51: 'HCM',
        52: 'HCM',
        53: 'HCM', 54: 'HCM', 55: 'HCM', 56: 'HCM', 57: 'HCM', 58: 'HCM', 59: 'HCM', 60: 'Đồng Nai',
        61: 'Bình Dương',
        62: 'Long An', 63: 'Tiền Giang', 64: 'Vĩnh Long', 65: 'Cần Thơ', 66: 'Đồng Tháp', 67: 'An Giang',
        68: 'Kiên Giang',
        69: 'Cà Mau', 70: 'Tây Ninh', 71: 'Bến Tre', 72: 'Vũng Tàu', 73: 'Quảng Bình', 74: 'Quảng Trị', 75: 'Huế',
        76: 'Quảng Ngãi', 77: 'Bình Định', 78: 'Phú Yên', 79: 'Nha Trang', 81: 'Gia Lai', 82: 'Kon Tum',
        83: 'Sóc Trăng',
        84: 'Trà Vinh', 85: 'Ninh Thuận', 86: 'Bình Thuận', 88: 'Vĩnh Phúc', 89: 'Hưng Yên', 90: 'Hà Nam',
        92: 'Quảng Nam',
        93: 'Bình Phước', 94: 'Bạc Liêu', 95: 'Hậu Giang', 97: 'Bắc Cạn', 98: 'Bắc Giang', 99: 'Bắc Ninh',
    }
```

Hình 3.7 Hàm để truy xuất thông tin tỉnh của biển số

```

strFinalString = ""
first_line = ""
second_line = ""

for i in char_x:
    (x, y, w, h) = cv2.boundingRect(cont[char_x_ind[i]])
    cv2.rectangle(roi, (x, y), (x + w, y + h), (0, 255, 0), 2)

    imgROI = thre_mor[y:y + h, x:x + w] # cắt kí tự ra khỏi hình

    imgROIResized = cv2.resize(imgROI, (
        RESIZED_IMAGE_WIDTH, RESIZED_IMAGE_HEIGHT)) # resize lại hình ảnh
    npaROIResized = imgROIResized.reshape(
        (1, RESIZED_IMAGE_WIDTH * RESIZED_IMAGE_HEIGHT)) # đưa hình ảnh về mảng 1 chiều
    # CHUYỂN ảnh thành ma trận có 1 hàng và số cột là tổng số điểm ảnh trong đó
    npaROIResized = np.float32(npaROIResized) # chuyển mảng về dạng float
    _, npaResults, neigh_resp, dists = knn.findNearest(npaROIResized,
        k=3) # call KNN function find_nearest; neigh_re

    strCurrentChar = str(chr(int(npaResults[0][0]))) # Lấy mã ASCII của kí tự đang xét
    cv2.putText(roi, strCurrentChar, (x, y + 50), cv2.FONT_HERSHEY_DUPLEX, 2, (0, 255, 255), 3)

    if (y < height / 3): # Biện số 1 hay 2 hàng
        first_line = first_line + strCurrentChar
    else:
        second_line = second_line + strCurrentChar

strFinalString = first_line + second_line
# print("\n License Plate " + str(n) + " is: " + first_line + " - " + second_line + "\n")
print("\n biện số là " + strFinalString + "\n")

```

Hình 3.8 Cắt và nhận diện kí tự

```

# lọc vùng biển số
new, contours, hierarchy = cv2.findContours(dilated_image, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10] # Lấy 10 contours có diện tích lớn nhất
screenCnt = []

for c in contours:
    peri = cv2.arcLength(c, True) # Tính chu vi
    approx = cv2.approxPolyDP(c, 0.06 * peri, True) # làm xấp xỉ đa giác, chỉ giữ contour có 4 cạnh
    [x, y, w, h] = cv2.boundingRect(approx.copy())
    ratio = w / h
    if (len(approx) == 4) and (0.8 <= ratio <= 1.5 or 4.5 <= ratio <= 6.5):
        screenCnt.append(approx)

if screenCnt is None:
    detected = 0
    print("No plate detected")
else:
    detected = 1

if detected == 1:
    n = 1
    for screenCnt in screenCnt:
        ##### Tính góc xoay#####
        (x1, y1) = screenCnt[0, 0]
        (x2, y2) = screenCnt[1, 0]
        (x3, y3) = screenCnt[2, 0]
        (x4, y4) = screenCnt[3, 0]
        array = [[x1, y1], [x2, y2], [x3, y3], [x4, y4]]
        sorted_array = array.sort(reverse=True, key=lambda x: x[1])
        (x1, y1) = array[0]
        (x2, y2) = array[1]

```

Hình 3.9 Lọc vùng biển số

Chương 4 KẾT LUẬN

4.1 Kết quả đạt được:

- Tìm hiểu được về ngôn ngữ lập trình Python
- Xây dựng được ứng dụng nhận dạng biển số
- Sử dụng được python và PyQt5 để chương xây dựng trình
- Sử dụng công cụ QT Designer để thiết kế giao diện

4.2 Hạn chế:

- Còn một số chức năng chưa hoàn thành
- Thiết kế còn đơn giản, sơ sài
- Tốc độ khởi chạy chương trình còn chậm, chưa tối ưu
- Nhận dạng có thể bị sai, tỉ lệ chưa được cao, nếu không nhận dạng được

sẽ tự thoát

- Ảnh/video đầu vào phải rõ nếu không sẽ không nhận dạng được

4.3 Hướng phát triển:

- Khắc phục các hạn chế, lập trình và kiểm thử các chức năng còn thiếu.
- Tối ưu mã nguồn tốc độ chạy của chương trình

Thiết kế giao diện dễ nhìn, đẹp mắt hơn.

TÀI LIỆU THAM KHẢO

1. Tài liệu nhận dạng biển số:

Bui Quang Manh(2018). Nhận diện biển số xe Việt Nam. Viblo. URL: <https://viblo.asia/p/nhan-dien-bien-so-xe-viet-nam-Do754P9L5M6>

2. Nghiên cứu nhận dạng biển số bằng OpenCV:

Ths Nguyễn Duy Linh, Nguyễn Văn Nhân, Đặng Văn Đạt(2018). Xây dựng ứng dụng nhận dạng biển số xe ô tô sử dụng mã nguồn mở Opencv. Tạp chí thông tin khoa học & công nghệ Quảng Bình.

URL: <https://skhcn.quangbinh.gov.vn/3cms/upload/khcn/File/TapChiKHCN/2018/so1/12.pdf>

3. K-nearest neighbor trong OpenCV2:

Nguyễn Phương Lan(2017). K-nearest neighbour trong OpenCV2. Viblo. URL: <https://viblo.asia/p/k-nearest-neighbour-trong-opencv2-V3m5W2wWlO7>

4. Chris Dahms (2016), KNN character recognition python:

Chris Dahm (2016). Opencv 3 KNN character recognition Python. Youtube. URL: <https://www.youtube.com/watch?v=c96w1JS28AY&t=6s>

5. Nghiên cứu về Contour:

Cộng đồng Opencv(2017). Tìm hiểu về contour, moments trong xử lý ảnh. URL: <https://congdongopencv.blogspot.com/2017/11/tim-hieu-ve-contour-moments-trong-xu-ly.html>

6. AI:

OpenAI.(2022). ChatGPT(Version 3.5). URL <https://chat.openai.com/>