TRƯỜNG ĐẠI HỌC NAM CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CƠ SỞ 2

NGÀNH CÔNG NGHỆ THÔNG TIN

Mã số ngành: 7480201

ĐỀ TÀI

ỨNG DỤNG XỬ LÍ ẢNH

Sinh Viên Thực Hiện

1. NGUYỄN THÀNH LUÂN MSSV: 2110684
2. NGUYỄN THÀNH LỘC MSSV: 213908

Khóa : K9

Cần Thơ, ..... tháng ..... năm 2024

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CƠ SỞ 2

Ngành: Công Nghệ Thông Tin

Mã số ngành: 7480201

ĐỀ TÀI

ỨNG DỤNG XỬ LÍ ẢNH

|  |  |
| --- | --- |
| Giảng viên hướng dẫn  ThS Trần Văn Thiện | Sinh viên thực hiện  NGUYỄN THÀNH LUÂN - 2110684  NGUYỄN THÀNH LỘC -213908 |

Cần Thơ, ..... tháng ..... năm 2024

LỜI CẢM ƠN

Sau một thời gian học tập và tích lũy kiến thức chuyên môn từ các môn học, cùng với việc thực hành các kiến thức, nhóm chúng em đã tìm hiểu thêm các kiến thức về lập trình. Điều này giúp chúng em có cái nhìn toàn diện hơn về lĩnh vực này. Chúng em đã áp dụng những kiến thức đã học trên lớp cùng các tài liệu bổ sung để triển khai dự án xây dựng và phát triển ứng dụng “Xử lí ảnh”.

Em xin chân thành cảm ơn thầy giáo, ThS Trần Văn Thiện – giảng viên khoa CNTT đã tận tâm và nhiệt tình dậy bảo trong suốt thời gian học và làm đồ án tốt nghiệp, thầy đã dành nhiều thời gian quý báu để tận tình chỉ bảo, hướng dẫn, định hướng cho em thực hiện đồ án, giúp em học hỏi những kinh nghiệm quý báu và đã đạt được những thành quả nhất định.

Em xin cảm ơn quý thầy cô Khoa Công nghệ thông tin Trường Đại học Nam Cần Thơ đã tận tâm giảng dạy và truyền đạt cho em những kiến thức quý báu để em hoàn thiện chương trình học tại trường.

Do chưa có nhiều kinh nghiệm cũng như các kiến thức còn hạn chế và khá mới, em đã có rất nhiều cố gắng nỗ lực để hoàn thành đề tài nhưng vẫn không tránh khỏi những thiếu sót. Kính mong nhận được góp ý của quý thầy cô để khắc phục và hoàn thiện đề tài tốt hơn.

Cuối lời, em xin chúc sức khỏe và thành công nhất đến qúy Thầy Cô Khoa Công Nghệ - Thông Tin Trường Đại Học Nam Cần Thơ.

Em xin chân thành cảm ơn thầy cô rất nhiều!

|  |  |
| --- | --- |
| Sinh viên thực hiện | Sinh viên thực hiện |
| Nguyễn Thành Luân | Nguyễn Thành Lộc |

LỜI CAM KẾT

Chúng em xin cam kết toàn bộ nội dung và kết quả được trình bày trong quyển đồ án chuyên ngành này về sản phẩm ứng dụng “Xử lí ảnh” được hoàn thiện dựa trên kiến thức, kinh nghiệm và trải nghiệm của nhóm. Kết quả này chưa từng được sử dụng cho bất kỳ đồ án chuyên ngành nào khác trên thị trường, và là thành quả từ sự nghiên cứu, sáng tạo, và nỗ lực của chúng em. Tất cả các thông tin, dữ liệu, và kết quả được trình bày đều là kết quả thực tế của quá trình nghiên cứu và phát triển ứng dụng của nhóm. Trong suốt quá trình thực hiện đồ án, chúng em đã đảm bảo tôn trọng sự riêng tư và quyền lợi cá nhân của các cá nhân, tổ chức liên quan và không có bất kỳ hành vi vi phạm nào xảy ra. Ngoài ra, trong bài báo cáo, chúng em có sử dụng một số nguồn tài liệu tham khảo và công cụ hỗ trợ. Toàn bộ thông tin, dữ liệu, hình ảnh, và tài liệu tham khảo đã được trích dẫn nguồn gốc và chú thích đầy đủ, minh bạch. Các thư viện, framework và công cụ hiện đại được áp dụng trong ứng dụng cũng được chúng em khai thác và trình bày rõ ràng, nhằm đảm bảo hiệu quả tối ưu cho việc lập trình và xử lý ảnh. Đồ án này được thực hiện với mục tiêu tạo ra một ứng dụng xử lý ảnh có khả năng hỗ trợ các tác vụ như nhận diện đối tượng, cải thiện chất lượng ảnh, và ứng dụng vào các lĩnh vực thực tiễn như y tế, giáo dục, hay phân tích thị giác. Với sự tận tâm, nỗ lực và tinh thần trách nhiệm cao nhất, chúng em cam kết rằng sản phẩm này không chỉ mang tính sáng tạo mà còn có giá trị ứng dụng thực tế, đóng góp vào sự phát triển của công nghệ xử lý ảnh tại Việt Nam cũng như trên thế giới. Chúng em sẽ tiếp tục nghiên cứu và phát triển sản phẩm này, với mong muốn tạo ra những bước tiến xa hơn trong lĩnh vực xử lý ảnh, góp phần giải quyết các thách thức mới và thúc đẩy sự đổi mới trong ngành công nghệ.

.

|  |  |
| --- | --- |
| Sinh viên thực hiện | Sinh viên thực hiện |
| Nguyễn Thành Luân | Nguyễn Thành Lộc |

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

*Cần Thơ, ngày .... tháng ..... năm 2024*

|  |  |
| --- | --- |
|  | Giảng viên hướng dẫn |

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

*Cần thơ, ngày ..... tháng ..... năm 2024*

|  |  |
| --- | --- |
|  | Giảng viên phản biện |

LỜI NÓI ĐẦU

Trong những năm gần đây, lĩnh vực xử lý ảnh đã và đang trở thành một trong những trọng tâm phát triển của ngành công nghệ thông tin, với những ứng dụng thực tiễn mang tính đột phá trong nhiều lĩnh vực như y tế, giáo dục, nông nghiệp, an ninh, và công nghiệp giải trí. Việc tận dụng sức mạnh của trí tuệ nhân tạo và các thuật toán xử lý ảnh hiện đại không chỉ giúp con người tiếp cận và hiểu rõ hơn về thế giới xung quanh mà còn mở ra vô vàn cơ hội cho sự sáng tạo và đổi mới.

Trong bối cảnh đó, chúng em giới thiệu đề tài “Ứng dụng xử lý ảnh”, với mục tiêu xây dựng một công cụ hỗ trợ việc phân tích và xử lý hình ảnh nhanh chóng, hiệu quả, và có giá trị ứng dụng cao. Sản phẩm được thiết kế như một minh họa điển hình về sự kết hợp giữa khả năng phân tích dữ liệu trực quan và sự chính xác của công nghệ máy học hiện đại.

Ứng dụng này tập trung vào các tính năng nổi bật như nhận diện đối tượng, tăng cường chất lượng hình ảnh, và phân loại ảnh. Với giao diện thân thiện, ứng dụng không chỉ mang lại sự tiện lợi mà còn mở ra cơ hội khám phá khả năng sáng tạo trong việc xử lý hình ảnh. Các thuật toán được tối ưu hóa nhằm đảm bảo hiệu năng cao, đồng thời tạo điều kiện để người dùng khai thác các ứng dụng như nhận diện khuôn mặt, phân tích ảnh y tế, hay xử lý hình ảnh cho quảng cáo và truyền thông.

Điểm nhấn của đề tài nằm ở tính tương tác và ứng dụng thực tế. Chúng em không chỉ mong muốn tạo ra một công cụ xử lý ảnh thông minh mà còn khuyến khích người dùng khám phá, học hỏi và ứng dụng công nghệ này trong các lĩnh vực khác nhau của cuộc sống. Ứng dụng không chỉ là một sản phẩm công nghệ mà còn là bước đầu cho những ý tưởng sáng tạo, giúp định hình các xu hướng mới trong lĩnh vực xử lý ảnh.

Chúng em hy vọng rằng đề tài **“**Ứng dụng xử lý ảnh” sẽ không chỉ là một bài toán công nghệ thú vị mà còn truyền cảm hứng cho các nghiên cứu và phát triển trong tương lai, đóng góp vào sự phát triển chung của ngành công nghệ thông tin.

.

**MỤC LỤC**

[CHƯƠNG 1 GIỚI THIỆU 1](#_Toc185348483)

[1.1. Tên đề tài 1](#_Toc185348484)

[1.2. Lý do chọn đề tài 1](#_Toc185348485)

[1.3. Hướng tiếp cận và Ưu nhược điểm của đề tài. 1](#_Toc185348486)

[CHƯƠNG 2 CƠ SỞ VÀ PHƯƠNG PHÁP NGHIÊN CỨU 4](#_Toc185348487)

[2.1. Cơ sở lý luận và ý nghĩa thực tiễn của đề tài 4](#_Toc185348488)

[2.2 Tổng quan về Qt Creator 4](#_Toc185348489)

[2.3 Lập trình với Qt Creator 7](#_Toc185348490)

[2.3.8 Các thư viện hỗ trợ khác 13](#_Toc185348491)

[2.4 Tổng quan về ngôn ngữ lập trình C++ 14](#_Toc185348492)

[2.4.3 Phương pháp nghiên cứu 15](#_Toc185348493)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM 16](#_Toc185348494)

[3.1 Phân tích yêu cầu và tính năng của phần mềm 16](#_Toc185348495)

[3.1.1 Mục tiêu và tính năng của phần mềm 16](#_Toc185348496)

[3.1.2 Các chức năng cơ bản 16](#_Toc185348497)

[3.1.3 Thiết kế phần mềm 16](#_Toc185348498)

[Giao diện chính: 16](#_Toc185348499)

[CHƯƠNG 4 GIAO DIỆN PHẦN MỀM 18](#_Toc185348500)

[4.1 Giao diện chính của phần mềm 18](#_Toc185348501)

[4.2 Giao diện khi thực thi phần mềm 19](#_Toc185348502)

[CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 24](#_Toc185348503)

[5.1 Kết luận 24](#_Toc185348504)

[5.2 Định hướng phát triển 24](#_Toc185348505)

[CHƯƠNG 6 TÀI LIỆU THAM KHẢO - PHỤ LỤC 26](#_Toc185348506)

[6. Tài liệu tham khảo 26](#_Toc185348507)

DANH SÁCH HÌNH ẢNH

[Hình 2. 1 Qt Creator 5](#_Toc185316918)

[Hình 2. 2 Đa nền tảng 6](#_Toc185316919)

[Hình 2. 3 Giao diện MenuBar 8](#_Toc185316920)

[Hình 2. 4 Giao diện Design 9](#_Toc185316921)

[Hình 2. 5 Giao diện Edit trong Qt 10](#_Toc185316922)

[Hình 2. 6 Giao diện Debug 11](#_Toc185316923)

[Hình 2. 7 Giao diện Project 12](#_Toc185316924)

[Hình 2. 8 Giao diện Console Window 12](#_Toc185316925)

[Hình 4. 1 Giao diện chính 18](#_Toc185316926)

[Hình 4. 2 Giao diện khi thực thi 19](#_Toc185316927)

[Hình 4. 3 Giao diện khi mở ảnh 19](#_Toc185316928)

[Hình 4. 3 Giao diện khi dùng chức năng(thêm sáng) 20](#_Toc185316929)

[Hình 4. 3 Giao diện biểu đồ ảnh xám 20](#_Toc185316930)

# CHƯƠNG 1 GIỚI THIỆU

## 1.1. Tên đề tài

Đề tài "Ứng dụng Xử lý Ảnh Thông minh" tập trung phát triển một công cụ công nghệ hiện đại sử dụng các thuật toán xử lý ảnh và trí tuệ nhân tạo để giải quyết các bài toán thực tiễn liên quan đến hình ảnh. Ứng dụng không chỉ hỗ trợ phân tích và cải thiện chất lượng hình ảnh mà còn mở rộng khả năng ứng dụng trong nhiều lĩnh vực như nhận diện đối tượng, phân loại ảnh, và tăng cường hình ảnh.

Với giao diện thân thiện và khả năng xử lý linh hoạt, ứng dụng mang lại giải pháp tối ưu cho người dùng, giúp đơn giản hóa các công việc liên quan đến hình ảnh như chỉnh sửa, nhận diện đặc điểm hay tạo ra các kết quả phân tích chuyên sâu. Đề tài nhấn mạnh sự kết hợp giữa tính chính xác, tốc độ và khả năng thích ứng của công nghệ hiện đại, góp phần nâng cao hiệu quả công việc trong các ngành nghề liên quan.

Không chỉ dừng lại ở mặt kỹ thuật, ứng dụng còn hướng tới việc tạo ra trải nghiệm người dùng tốt nhất, hỗ trợ họ tận dụng tối đa sức mạnh của công nghệ xử lý ảnh trong các bài toán đời sống hàng ngày hoặc công việc chuyên môn. Chúng em hy vọng rằng đề tài sẽ đóng góp tích cực vào việc ứng dụng trí tuệ nhân tạo trong đời sống, đồng thời mở ra cơ hội để khám phá và phát triển các giải pháp mới trong lĩnh vực xử lý ảnh.

## 1.2. Lý do chọn đề tài

Đề tài "Ứng dụng Xử lý Ảnh Thông minh" tập trung phát triển một công cụ công nghệ hiện đại sử dụng các thuật toán xử lý ảnh và trí tuệ nhân tạo để giải quyết các bài toán thực tiễn liên quan đến hình ảnh. Ứng dụng không chỉ hỗ trợ phân tích và cải thiện chất lượng hình ảnh mà còn mở rộng khả năng ứng dụng trong nhiều lĩnh vực như nhận diện đối tượng, phân loại ảnh, và tăng cường hình ảnh.

Với giao diện thân thiện và khả năng xử lý linh hoạt, ứng dụng mang lại giải pháp tối ưu cho người dùng, giúp đơn giản hóa các công việc liên quan đến hình ảnh như chỉnh sửa, nhận diện đặc điểm hay tạo ra các kết quả phân tích chuyên sâu. Đề tài nhấn mạnh sự kết hợp giữa tính chính xác, tốc độ và khả năng thích ứng của công nghệ hiện đại, góp phần nâng cao hiệu quả công việc trong các ngành nghề liên quan.

Không chỉ dừng lại ở mặt kỹ thuật, ứng dụng còn hướng tới việc tạo ra trải nghiệm người dùng tốt nhất, hỗ trợ họ tận dụng tối đa sức mạnh của công nghệ xử lý ảnh trong các bài toán đời sống hàng ngày hoặc công việc chuyên môn. Chúng em hy vọng rằng đề tài sẽ đóng góp tích cực vào việc ứng dụng trí tuệ nhân tạo trong đời sống, đồng thời mở ra cơ hội để khám phá và phát triển các giải pháp mới trong lĩnh vực xử lý ảnh.

## 1.3. Hướng tiếp cận và Ưu nhược điểm của đề tài.

### 1.3.1. Hướng tiếp cận của đề tài

Đề tài "Ứng dụng Xử lý Ảnh Thông minh" tập trung vào các chức năng cơ bản để xử lý và phân tích hình ảnh một cách hiệu quả. Hướng tiếp cận của chúng em bắt đầu từ việc nghiên cứu và áp dụng các thuật toán xử lý ảnh cơ bản như chuyển đổi định dạng ảnh, làm mịn, phát hiện cạnh và lọc nhiễu. Bên cạnh đó, ứng dụng tích hợp một số công nghệ hiện đại từ thư viện OpenCV để hỗ trợ nhận diện và xử lý hình ảnh nhanh chóng. Mục tiêu chính là xây dựng một công cụ thân thiện với giao diện đơn giản, cho phép người dùng dễ dàng tải ảnh lên, áp dụng các bộ lọc hoặc chỉnh sửa cơ bản, và lưu kết quả. Chúng em tập trung vào việc tạo ra một ứng dụng dễ sử dụng nhưng hiệu quả, hướng đến người dùng phổ thông hoặc những ai mới tiếp cận lĩnh vực xử lý ảnh.

Ứng dụng được thiết kế với giao diện đơn giản, cho phép người dùng tải ảnh, thực hiện các thao tác xử lý nhanh chóng, và lưu lại kết quả. Hướng tiếp cận này nhằm tạo ra một công cụ hữu ích, dễ sử dụng, phù hợp với nhu cầu cơ bản của người dùng phổ thông.

### 1.3.2. Ưu điểm của đề tài:

Đề tài "Ứng dụng Xử lý Ảnh Thông minh" không chỉ đáp ứng nhu cầu cơ bản của người dùng mà còn mang lại nhiều ưu điểm đáng chú ý. Ứng dụng tập trung vào các chức năng như chuyển đổi định dạng ảnh, lọc nhiễu, phát hiện cạnh, và chỉnh sửa ảnh cơ bản, giúp người dùng cải thiện chất lượng hình ảnh một cách nhanh chóng và hiệu quả. Giao diện đơn giản, dễ sử dụng giúp người dùng dễ dàng thao tác mà không gặp khó khăn, đặc biệt là những người mới tiếp cận công nghệ xử lý ảnh. Hơn nữa, việc sử dụng thư viện OpenCV giúp tối ưu hóa hiệu suất và giảm thiểu chi phí phát triển, đồng thời tạo nền tảng vững chắc cho các tính năng mở rộng trong tương lai. Ứng dụng còn có khả năng xử lý nhiều loại ảnh khác nhau, từ ảnh cá nhân đến ảnh sản phẩm, phù hợp với nhiều đối tượng người dùng và mục đích sử dụng.

Ngoài những ưu điểm đã nêu, đề tài "Ứng dụng Xử lý Ảnh Thông minh" còn có khả năng mở rộng và tính linh hoạt cao. Ứng dụng được thiết kế để dễ dàng tích hợp các tính năng mới trong tương lai, như hỗ trợ các công cụ chỉnh sửa phức tạp hơn hoặc tối ưu hóa chất lượng ảnh cho các ứng dụng đặc thù (chẳng hạn như ảnh y tế hoặc ảnh chụp sản phẩm). Hệ thống xử lý ảnh của ứng dụng được xây dựng để đảm bảo tốc độ xử lý nhanh, ngay cả với các bức ảnh có độ phân giải cao, giúp tiết kiệm thời gian cho người dùng. Hơn nữa, việc sử dụng mã nguồn mở không chỉ giảm chi phí phát triển mà còn mở ra cơ hội cộng tác và phát triển thêm các tính năng mới từ cộng đồng lập trình viên.

### 1.3.3. Nhược điểm của đề tài:

Mặc dù "Ứng dụng Xử lý Ảnh Thông minh" có nhiều ưu điểm, nhưng cũng tồn tại một số nhược điểm cần được lưu ý. Đầu tiên, vì tập trung vào các chức năng cơ bản, ứng dụng có thể chưa đáp ứng được nhu cầu xử lý ảnh phức tạp hoặc các tác vụ chuyên sâu như phân tích ảnh y tế, nhận diện đối tượng trong ảnh động (video), hoặc xử lý ảnh 3D. Thứ hai, tốc độ xử lý có thể giảm khi làm việc với ảnh có độ phân giải quá cao hoặc khi thực hiện các tác vụ nặng như xử lý hàng loạt ảnh cùng lúc, đặc biệt trên những thiết bị có cấu hình thấp. Thứ ba, do là một ứng dụng dành cho người dùng phổ thông, các tính năng nâng cao hoặc tính tùy chỉnh có thể bị hạn chế, khiến người dùng chuyên sâu hoặc các chuyên gia xử lý ảnh gặp khó khăn khi muốn áp dụng các kỹ thuật phức tạp hơn. Cuối cùng, việc phụ thuộc vào các thư viện mã nguồn mở như OpenCV có thể gây ra vấn đề về khả năng tương thích hoặc cập nhật phiên bản trong tương lai, ảnh hưởng đến sự ổn định của ứng dụng.

Một nhược điểm khác của đề tài là hạn chế về khả năng mở rộng và tính năng nâng cao. Vì ứng dụng tập trung vào các chức năng cơ bản để xử lý ảnh, nó có thể gặp khó khăn trong việc phát triển các tính năng phức tạp hơn trong tương lai nếu không có kế hoạch mở rộng rõ ràng. Điều này có thể khiến ứng dụng không đáp ứng được yêu cầu của người dùng có nhu cầu xử lý ảnh chuyên sâu hơn, như nhận diện đối tượng nâng cao, phân tích ảnh bằng AI, hoặc xử lý video. Việc thiếu tính năng mở rộng này có thể làm giảm khả năng cạnh tranh của ứng dụng so với các sản phẩm xử lý ảnh chuyên nghiệp hơn trên thị trường.

# CHƯƠNG 2 CƠ SỞ VÀ PHƯƠNG PHÁP NGHIÊN CỨU

## 2.1. Cơ sở lý luận và ý nghĩa thực tiễn của đề tài

Cơ sở lý luận Đề tài “Ứng dụng Xử lý Ảnh” được xây dựng trên nền tảng các lý thuyết và phương pháp trong lĩnh vực xử lý ảnh và trí tuệ nhân tạo. Xử lý ảnh là một nhánh con của khoa học máy tính, sử dụng các thuật toán và kỹ thuật để cải thiện hoặc phân tích hình ảnh từ các nguồn dữ liệu như camera, cảm biến, hoặc các hình ảnh tĩnh. Các phương pháp xử lý ảnh cơ bản như chuyển đổi định dạng, lọc nhiễu, phát hiện cạnh, và làm sắc nét ảnh thường được sử dụng để cải thiện chất lượng hình ảnh và chuẩn bị dữ liệu cho các tác vụ phức tạp hơn. Các công cụ mạnh mẽ như OpenCV được ứng dụng rộng rãi trong lĩnh vực này nhờ vào các hàm và thuật toán hỗ trợ, giúp giải quyết các bài toán từ đơn giản đến phức tạp trong việc xử lý và phân tích hình ảnh.

Ngoài ra, đề tài còn dựa trên các lý thuyết về giao diện người dùng (UI) và trải nghiệm người dùng (UX) nhằm tạo ra một ứng dụng dễ sử dụng và thân thiện với người dùng, đặc biệt là đối với những người chưa có nhiều kinh nghiệm về công nghệ xử lý ảnh.

Ý nghĩa thực tiễn ứng dụng xử lý ảnh có ý nghĩa thực tiễn sâu rộng trong nhiều lĩnh vực. Đầu tiên, nó mang lại giải pháp cho người dùng trong việc chỉnh sửa và cải thiện chất lượng hình ảnh một cách nhanh chóng và dễ dàng. Những tính năng cơ bản như lọc nhiễu, phát hiện cạnh, hay thay đổi định dạng ảnh giúp người dùng cá nhân, các nhà thiết kế đồ họa, hoặc các doanh nghiệp cải thiện hình ảnh của họ một cách hiệu quả mà không cần đến phần mềm phức tạp.

Trong các lĩnh vực như y tế, ứng dụng có thể được sử dụng để hỗ trợ việc cải thiện chất lượng hình ảnh y tế, giúp bác sĩ nhận diện và phân tích các bức ảnh từ chẩn đoán hình ảnh (như X-quang, MRI) với độ chính xác cao hơn. Trong ngành công nghiệp và thương mại điện tử, việc xử lý ảnh sản phẩm hoặc hình ảnh quảng cáo có thể giúp nâng cao chất lượng hình ảnh và tăng tính hấp dẫn đối với người tiêu dùng.

Bên cạnh đó, việc phát triển ứng dụng này còn mang lại những lợi ích về mặt giáo dục và nghiên cứu, khi người dùng và các nhà nghiên cứu có thể dễ dàng tiếp cận các công cụ xử lý ảnh cơ bản để phục vụ cho việc học hỏi và nghiên cứu trong lĩnh vực công nghệ thông tin và xử lý dữ liệu hình ảnh.

## 2.2 Tổng quan về Qt Creator

### 2.2.1. Giới thiệu về Qt Creator

Qt Creator là một môi trường phát triển tích hợp (IDE) mạnh mẽ, được thiết kế dành riêng cho việc phát triển các ứng dụng sử dụng Qt Framework. Nó hỗ trợ phát triển các ứng dụng đa nền tảng với C++ và QML, với mục tiêu giúp lập trình viên tạo ra các ứng dụng giao diện người dùng (GUI) dễ dàng và hiệu quả. Qt Creator không chỉ hỗ trợ Qt, mà còn có thể được sử dụng cho các dự án C++ thuần túy và nhiều ngôn ngữ lập trình khác.



Hình 2. 1 Qt Creator

### 2.2.2 Quá trình phát triển của Qt Creator

Qt Creator là môi trường phát triển tích hợp (IDE) do The Qt Company phát triển, hỗ trợ lập trình viên phát triển ứng dụng sử dụng Qt Framework. Ra mắt vào năm 2009, Qt Creator đã được cải tiến với các tính năng như hỗ trợ Qt Designer, công cụ xây dựng qmake, tích hợp CMake và phát triển đa nền tảng (Windows, macOS, Linux, Android, iOS). Nó cung cấp các công cụ mạnh mẽ như gỡ lỗi, thiết kế giao diện, kiểm soát phiên bản và kiểm thử, giúp nâng cao hiệu suất và hiệu quả phát triển phần mềm. Qt Creator là lựa chọn lý tưởng cho phát triển ứng dụng Qt đa nền tảng và di động.

### 2.2.3 Ưu điểm của Qt Creator

Qt Creator có nhiều ưu điểm nổi bật, bao gồm hỗ trợ đa nền tảng như Windows, macOS, Linux, Android và iOS. Với trình soạn thảo mã nguồn thông minh, công cụ gỡ lỗi mạnh mẽ và thiết kế giao diện người dùng trực quan, việc phát triển ứng dụng trở nên dễ dàng. Nó cũng hỗ trợ các hệ thống xây dựng như qmake và CMake, công cụ kiểm thử tích hợp và khả năng làm việc với Git, SVN, giúp tăng hiệu quả công việc.

A group of icons with text

Description automatically generated with medium confidence

Hình 2. 2 Đa nền tảng

Giao diện thân thiện: Qt Creator có giao diện trực quan, dễ sử dụng và học hỏi, thích hợp cho cả lập trình viên mới và chuyên gia. Công cụ này cung cấp các tính năng được sắp xếp hợp lý, giúp người dùng dễ dàng tiếp cận và làm việc hiệu quả.

Hệ sinh thái phong phú: Qt Creator tích hợp với Qt Framework và hỗ trợ Qt Marketplace, nơi cung cấp nhiều tài nguyên phát triển như thư viện, widget, và các công cụ mở rộng, giúp tăng tốc quá trình phát triển và tiết kiệm thời gian.

Hỗ trợ mạnh mẽ cho đa nền tảng: Qt Creator hỗ trợ phát triển ứng dụng cho nhiều nền tảng như Windows, macOS, Linux, Android, và iOS. Nó cung cấp các công cụ mạnh mẽ để phát triển cả ứng dụng 2D và 3D, giúp lập trình viên tạo ra các ứng dụng từ đơn giản đến phức tạp.

### 2.2.4 Các khái niệm trong Unity

**Qt Framework**: Là bộ công cụ phần mềm mở rộng của C++ được sử dụng để phát triển ứng dụng GUI (giao diện người dùng đồ họa) đa nền tảng. Qt Framework cung cấp các thư viện, công cụ và widget để xây dựng các ứng dụng có giao diện đẹp mắt và hiệu suất cao.

**Qt Creator**: Là môi trường phát triển tích hợp (IDE) dành riêng cho việc phát triển ứng dụng với Qt Framework. Qt Creator cung cấp các công cụ cần thiết như trình soạn thảo mã nguồn, gỡ lỗi, công cụ xây dựng, và thiết kế giao diện người dùng.

**QMake**: Là một công cụ xây dựng (build system) mặc định trong Qt, giúp biên dịch và xây dựng các dự án Qt. QMake tạo ra các file cấu hình để xây dựng ứng dụng trên các nền tảng khác nhau.

**CMake**: Là một hệ thống xây dựng mở rộng có thể thay thế cho QMake trong Qt Creator. CMake được sử dụng khi lập trình viên muốn xây dựng ứng dụng Qt hoặc các ứng dụng đa nền tảng mà không cần phải phụ thuộc vào QMake.

**Qt Designer**: Là công cụ thiết kế giao diện người dùng trong Qt Creator. Qt Designer cho phép lập trình viên tạo và chỉnh sửa các giao diện người dùng (GUI) bằng cách kéo và thả các widget mà không cần phải viết mã.

**Widget**: Là các phần tử giao diện người dùng trong Qt, như nút bấm, hộp văn bản, bảng điều khiển,... Các widget này giúp xây dựng giao diện người dùng của ứng dụng.

**Signals and Slots**: Là cơ chế giao tiếp trong Qt, giúp kết nối các đối tượng và xử lý các sự kiện. Signals được phát ra khi một sự kiện xảy ra, và Slots là các hàm được gọi để xử lý sự kiện đó.

**QML (Qt Modeling Language)**: Là một ngôn ngữ mô tả giao diện người dùng động và mạnh mẽ, thường được sử dụng kết hợp với Qt Quick để phát triển giao diện người dùng hiện đại và tương tác cao.

**Qt Quick**: Là một framework trong Qt để phát triển ứng dụng với giao diện người dùng động sử dụng QML. Qt Quick hỗ trợ phát triển các ứng dụng có giao diện mượt mà và tương tác tốt, đặc biệt là cho các ứng dụng di động.

**Debugging**: Là công cụ gỡ lỗi trong Qt Creator, giúp lập trình viên kiểm tra mã nguồn, tìm và sửa lỗi trong quá trình phát triển ứng dụng. Qt Creator hỗ trợ gỡ lỗi cho cả ứng dụng Qt và C++.

**Build** **&** **Run**: Là quá trình biên dịch và chạy ứng dụng. Qt Creator cung cấp công cụ để biên dịch mã nguồn thành mã máy và sau đó chạy ứng dụng trên môi trường đích, giúp lập trình viên kiểm tra và phát triển ứng dụng.

## 2.3 Lập trình với Qt Creator

### 2.3.1Thanh Menu (Menu Bar)

Mô tả: ThanhMenuBar trong QtCreator là thanh công cụ chính nằm ở phía trên cùng của cửa sổ làm việc. Nó cung cấp các tùy chọn và lệnh cơ bản để làm việc với dự án, biên dịch và chạy chương trình.

Chức năng: MenuBar trong QtCreator đóng vai trò trung tâm trong việc cung cấp các chức năng như chỉnh sửa mã nguồn, xây dựng và gỡ lỗi dự án, tùy chỉnh giao diện, và phântíchhiệusuất. Việc hiểu rõ các chức năng này sẽ giúp lập trình viên tối ưu hóa quá trình phát triển ứng dụng với Qt Creator.

Các công cụ trong MenuBar: Các công cụ như quản lý tệp và dự án: tạo mới, mở, lưu, đóng tệp/dự án và thoát khỏi Qt Creator (File), chỉnh sửa mã nguồn: sao chép, dán, tìm kiếm, thay thế và hoàn tác/thực hiện lại thao tác. (Edit), tùy chỉnh giao diện: phóng to, thu nhỏ, chia màn hình, bật/tắt các thanh công cụ và cửa sổ con. (View), xây dựng và biên dịch: biên dịch, làm sạch và chạy dự án đang phát triển(Build), gỡ lỗi chương trình: chạy, dừng gỡ lỗi và kiểm tra từng bước thực thi của mã nguồn(Debug), phân tích mã nguồn: tìm lỗi, tối ưu hóa bộ nhớ và đánh giá hiệu suất chương trình(Analyze), tùy chỉnh và sử dụng các công cụ bên ngoài: cài đặt trình biên dịch, định dạng mã nguồn và tìm kiếm nhanh(Tools), quản lý cửa sổ làm việc: chia nhỏ, sắp xếp và khôi phục bố cục cửa sổ(Window) và cuối cùng là hỗ trợ và tài liệu: truy cập tài liệu, hướng dẫn sử dụng và thông tin về phiên bản Qt Creator(Help).

A screenshot of a computer

Description automatically generated

Hình 2. 3 Giao diện MenuBar

### 2.3.2 Design:

Thiết kế giao diện đồ họa trực quan: Người dùng có thể tạo và chỉnh sửa giao diện ứng dụng bằng cách kéo thả các widget (nút bấm, ô nhập liệu, nhãn...) hoặc các thành phần QML. Công cụ này phù hợp với việc phát triển các ứng dụng C++ với Qt Widgets hoặc QML.

Cửa sổ làm việc chia thành các phần:

Widget Library/ComponentLibrary: Thư viện các thành phần giao diện (nút bấm, hộp thoại, thanh trượt...).

FormEditor: Khu vực chính để thiết kế giao diện (kéo thả và sắp xếp các widget).

PropertyEditor: Chỉnh sửa thuộc tính của các widget như kích thước, màu sắc, văn bản hiển thị,...

Object Inspector: Hiển thị cấu trúc và danh sách các widget đã thêm vào giao diện.

Tạo và chỉnh sửa tệp .ui hoặc QML: Tệp **.**ui: Là tệp XML mô tả giao diện khi bạn thiết kế trong chế độ kéo thả với Qt Widgets. QML là ngôn ngữ khai báo giao diện dùng để thiết kế giao diện hiện đại, hỗ trợ hoạt ảnh và các tương tác động.

Xem trước giao diện: Cho phép Preview (xem trước) giao diện ứng dụng ngay trong Qt Creator mà không cần biên dịch.

Kết nối sự kiện (Signal và Slot): Cho phép kết nối các widget với logic của ứng dụng bằng cách sử dụng Signal-Slot trong Qt Widgets hoặc xử lý các sự kiện trong QML.

A screenshot of a computer

Description automatically generated

Hình 2. 4 Giao diện Design

### 2.3.3 Edit:

Mô tả: Phần Edit là khu vực chứa các công cụ và chức năng hỗ trợ việc chỉnh sửa mã nguồn nhanh chóng và hiệu quả. Menu Edit cung cấp các tính năng cần thiết cho việc viết, chỉnh sửa và tổ chức mã nguồn trong dự án.

Chức năng: Edit trong Qt Creator cung cấp các công cụ hỗ trợ chỉnh sửa mã nguồn một cách hiệu quả và nhanh chóng. Các chức năng cơ bản như Undo/Redo (hoàn tác và thực hiện lại thao tác), Cut/Copy/Paste (cắt, sao chép và dán đoạn mã) giúp lập trình viên thao tác nhanh chóng. Tính năng Find/Replace cho phép tìm kiếm và thay thế từ khóa trong tệp hoặc toàn bộ dự án, cùng với Go to Line/Definition giúp điều hướng nhanh đến dòng hoặc định nghĩa của hàm và biến. Công cụ Indentation và Formatting hỗ trợ tự động thụt lề, định dạng mã nguồn và thêm/xóa chú thích. Ngoài ra, Qt Creator còn hỗ trợ Code Completion để gợi ý và tự động hoàn thành mã nguồn, cùng với các công cụ Refactor như đổi tên biến, tái cấu trúc mã giúp tối ưu hóa và cải thiện chất lượng mã nguồn. Cuối cùng, tính năng Bookmarks cho phép đánh dấu và chuyển nhanh giữa các dòng mã quan trọng, giúp tăng năng suất và tiết kiệm thời gian khi làm việc.

A screenshot of a computer

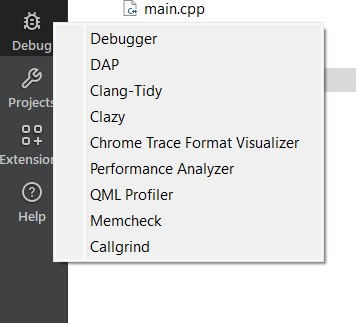
Description automatically generated

Hình 2. 5 Giao diện Edit trong Qt

### 2.3.4 Debug

Mô tả: Phần Debug trong Qt Creator là công cụ giúp lập trình viên thực hiện quá trình gỡ lỗi chương trình một cách trực quan và chính xác. Nó cho phép theo dõi hoạt động của chương trình trong chế độ Debug, xác định các lỗi và kiểm tra dòng mã trong quá trình thực thi. Qt Creator tích hợp mạnh mẽ với các công cụ gỡ lỗi như GDB (GNU Debugger) hoặc LLDB, giúp lập trình viên kiểm tra và khắc phục lỗi dễ dàng.

Chức năng: Debug trong Qt Creator cung cấp các chức năng mạnh mẽ để gỡ lỗi chương trình, bao gồm khả năng chạy ứng dụng trong chế độ Debug, thiết lập các điểm dừng (Breakpoints) để tạm dừng chương trình tại các dòng mã cụ thể, và thực thi từng bước mã nguồn với các công cụ như Step Over, Step Into, và Step Out. Ngoài ra, Debug cho phép theo dõi giá trị của các biến trong thời gian thực, kiểm tra và đánh giá biểu thức hoặc điều kiện thông qua Expression Evaluator, cùng với việc hiển thị Call Stack để phân tích trình tự gọi hàm trong chương trình. Công cụ này cũng hỗ trợ quan sát bộ nhớ, phát hiện lỗi liên quan đến tràn hoặc truy cập sai bộ nhớ, và theo dõi các luồng thực thi trong các ứng dụng đa luồng. Với các tính năng toàn diện này, Debug giúp lập trình viên phát hiện và sửa lỗi nhanh chóng, đảm bảo chương trình hoạt động chính xác và hiệu quả.



Hình 2. 6 Giao diện Debug

### 2.3.5 Projects

Mô tả: Projects trong Qt Creator là nơi để quản lý và cấu hình các thiết lập liên quan đến dự án. Nó cung cấp giao diện trực quan để lập trình viên tùy chỉnh cách biên dịch, chạy và gỡ lỗi cho từng nền tảng hoặc môi trường làm việc. Phần này thường được sử dụng để điều chỉnh các cấu hình như kit phát triển, mục tiêu triển khai, tùy chọn build, và các tham số chạy ứng dụng.

Chức năng: Cửa sổ Projects trong Qt Creator cung cấp các chức năng quản lý và cấu hình dự án một cách chi tiết, bao gồm việc lựa chọn và cấu hình kit phát triển với các công cụ biên dịch, liên kết và môi trường chạy phù hợp cho từng nền tảng. Lập trình viên có thể tùy chỉnh cấu hình Build để điều chỉnh đường dẫn biên dịch, thư viện liên kết, cờ biên dịch, và các bước xử lý trước hoặc sau khi build. Ngoài ra, phần Run Configuration cho phép thiết lập các tham số chạy ứng dụng như dòng lệnh, thư mục làm việc, hoặc biến môi trường. Deploy Targets hỗ trợ cấu hình môi trường triển khai, từ thiết bị nhúng đến máy ảo, trong khi Debug Settings cho phép tùy chỉnh công cụ gỡ lỗi và các tham số liên quan. Hỗ trợ đa nền tảng của Projects cho phép chuyển đổi nhanh giữa các cấu hình Debug, Release hoặc Profile, giúp tối ưu hóa quy trình phát triển ứng dụng và triển khai linh hoạt.

A screenshot of a computer

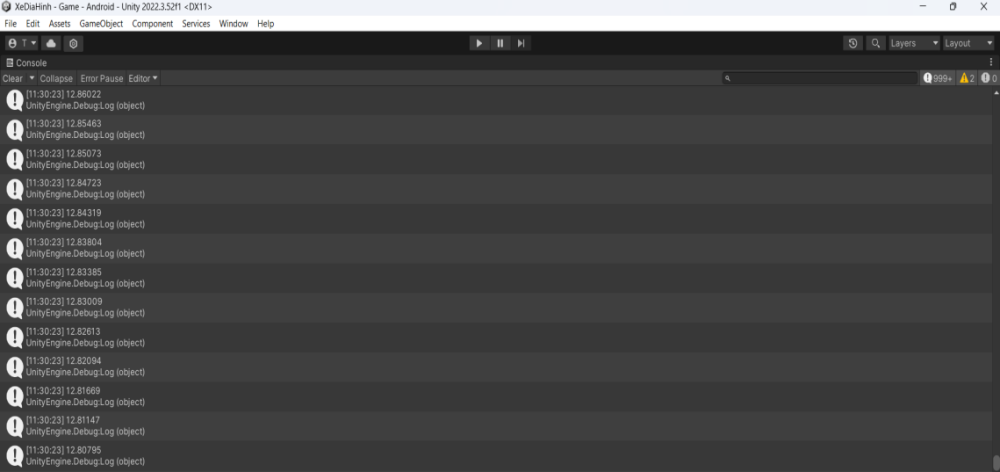
Description automatically generated

Hình 2. 7 Giao diện Project

### 2.3.6 Console Window

Mô tả: Cửa sổ Console là nơi hiển thị các thông báo, lỗi, và cảnh báo trong quá trình phát triển.

Chức năng: Khi bạn chạy trò chơi hoặc ứng dụng, Console sẽ hiển thị các log thông báo (debug), các lỗi (errors) hoặc các cảnh báo (warnings) từ mã nguồn của bạn. Đây là một công cụ quan trọng để phát hiện và sửa lỗi trong dự án.



Hình 2. 8 Giao diện Console Window

### 2.3.7 Extensions

Mô tả: Extensions trong Qt Creator cung cấp khả năng mở rộng và tùy chỉnh môi trường phát triển thông qua việc cài đặt, kích hoạt hoặc cấu hình các tiện ích bổ sung (plugins). Đây là công cụ giúp lập trình viên bổ sung các tính năng hoặc tích hợp công cụ mới vào Qt Creator, mở rộng khả năng phát triển ứng dụng.

Chức năng: Phần Extensions trong Qt Creator cung cấp các chức năng quản lý và mở rộng môi trường phát triển thông qua việc cài đặt và cấu hình các tiện ích bổ sung (plugins). Người dùng có thể bật, tắt hoặc cập nhật các plugin hiện có, cũng như cài đặt thêm các tiện ích mở rộng từ nguồn bên ngoài để bổ sung tính năng như kiểm tra mã, phân tích hiệu suất hoặc hỗ trợ thêm ngôn ngữ lập trình khác. Extensions còn cho phép tích hợp công cụ bên ngoài, như hệ thống quản lý mã nguồn (Git, SVN) hoặc công cụ CI/CD, và tùy chỉnh môi trường làm việc với các thành phần mới phù hợp với yêu cầu dự án. Bên cạnh đó, nó hỗ trợ kiểm tra trạng thái plugin và gỡ lỗi nếu xảy ra vấn đề, giúp lập trình viên tối ưu hóa quy trình phát triển một cách linh hoạt và hiệu quả.

## 2.3.8 Các thư viện hỗ trợ khác

OpenCV:

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được sử dụng rộng rãi trong xử lý ảnh và thị giác máy tính. OpenCV hỗ trợ nhiều ngôn ngữ lập trình như C++, Python, Java, và cũng có thể sử dụng với các công cụ như Qt.

Các tính năng chính của OpenCV:

Đọc, ghi và chuyển đổi ảnh: OpenCV hỗ trợ nhiều định dạng ảnh như JPEG, PNG, BMP, và TIFF. Chỉnh sửa ảnh xử lý các thao tác cơ bản như thay đổi kích thước, xoay, cắt ảnh, điều chỉnh độ sáng, độ tương phản, và các bộ lọc.

**Phân tích ảnh và nhận diện đối tượng:**

Phân đoạn ảnh: Tách biệt các đối tượng trong ảnh.

Phát hiện biên: Sử dụng các thuật toán như Canny edge detection.

Nhận diện đối tượng: Nhận diện các đối tượng trong ảnh qua các phương pháp như Haar Cascades, Deep Learning và HOG (Histogram of Oriented Gradients).

Theo dõi đối tượng: Theo dõi chuyển động của đối tượng trong video hoặc chuỗi hình ảnh. Tăng hiệu suất và khả năng mở rộng:

**Thị giác máy tính nâng cao:**

Nhận dạng khuôn mặt: Phát hiện và nhận diện khuôn mặt trong ảnh hoặc video.

Đo lường chiều cao, chiều rộng, và các thuộc tính khác của đối tượng: Dùng các phương pháp phân tích hình học và các thuật toán nhận dạng.

Học máy và Deep Learning:

OpenCV tích hợp với DNN module để hỗ trợ các mô hình học sâu (deep learning). Bạn có thể sử dụng các mô hình đã huấn luyện sẵn như TensorFlow, Caffe, Torch, và PyTorch để nhận dạng ảnh, phân loại và phát hiện đối tượng.

Ứng dụng của OpenCV:

Nhận diện và theo dõi khuôn mặt: Được ứng dụng trong các hệ thống nhận dạng khuôn mặt, bảo mật và giám sát.

Phân tích video: OpenCV có thể dùng trong các hệ thống giám sát video, nhận diện chuyển động, theo dõi đối tượng, và phân tích hành vi.

Thực tế tăng cường (AR): OpenCV hỗ trợ nhận diện và theo dõi các đối tượng trong môi trường thực tế.

## 2.4 Tổng quan về ngôn ngữ lập trình C++

### 2.4.1 Giới thiệu tổng quát về ngôn ngữ lập trình C++

C++ là một ngôn ngữ lập trình đa năng, hiệu quả và có tính mở rộng cao, được phát triển bởi Bjarne Stroustrup vào năm 1979 tại AT&T Bell Labs.

C++ là một sự mở rộng của ngôn ngữ C, kết hợp giữa lập trình thủ tục và lập trình hướng đối tượng (OOP), đồng thời hỗ trợ các tính năng lập trình khác như lập trình generic và lập trình hàm. Được xem là một trong những ngôn ngữ mạnh mẽ và phổ biến nhất trong ngành công nghiệp phần mềm.

C++ được sử dụng để phát triển các hệ thống phần mềm hiệu suất cao, ứng dụng đồ họa, trò chơi, hệ thống nhúng và nhiều lĩnh vực khác.

### 2.4.2 Các đặc trưng về ngôn ngữ lập trình C++

C++ hỗ trợ các đặc tính của lập trình hướng đối tượng như encapsulation (đóng gói), inheritance (kế thừa), và polymorphism (đa hình), giúp mã nguồn trở nên dễ bảo trì, mở rộng và tái sử dụng.

C++ cho phép lập trình viên kiểm soát trực tiếp bộ nhớ thông qua con trỏ (pointers), giúp tối ưu hóa hiệu suất và sử dụng bộ nhớ hiệu quả. Điều này giúp C++ trở thành một lựa chọn lý tưởng cho các ứng dụng yêu cầu hiệu suất cao và tài nguyên hạn chế.

C++ là ngôn ngữ biên dịch, có thể biên dịch trực tiếp thành mã máy, giúp tăng tốc độ thực thi và tối ưu hóa hiệu suất của chương trình. Nó rất phù hợp cho các ứng dụng yêu cầu xử lý nhanh chóng, như game, hệ thống thời gian thực và phần mềm đồ họa.

C++ hỗ trợ template, cho phép viết mã chung (generic code) mà không cần phải lo lắng về loại dữ liệu. Điều này làm cho C++ trở nên mạnh mẽ trong việc xây dựng các thư viện phần mềm có thể tái sử dụng và mở rộng.

Ngoài những đặc trưng trên thì C++ còn có một số ứng dụng như:

Phát triển phần mềm hệ thống: Hệ điều hành, trình biên dịch, và các công cụ phần mềm yêu cầu hiệu suất cao.

Phát triển game: C++ là lựa chọn chính cho việc phát triển game nhờ vào khả năng tối ưu hóa và hỗ trợ đồ họa mạnh mẽ.

Ứng dụng tài chính và khoa học: Các phần mềm mô phỏng, phân tích dữ liệu, và tính toán phức tạp.

Hệ thống nhúng: C++ được sử dụng trong các thiết bị điện tử, robot, và các hệ thống nhúng yêu cầu kiểm soát phần cứng trực tiếp.

## 2.4.3 Phương pháp nghiên cứu

Phương pháp nghiên cứu về thiết kế phần mềm xử lý ảnh tập trung vào việc phát triển và cải tiến các công cụ, thuật toán để thực hiện các tác vụ như nhận diện hình ảnh, phân tích dữ liệu hình ảnh và chuyển đổi các tín hiệu hình ảnh thành thông tin có thể sử dụng được. Cũng như trong các lĩnh vực phần mềm khác, nghiên cứu thiết kế phần mềm xử lý ảnh có thể bao gồm các phương pháp lý thuyết, kỹ thuật triển khai và các thử nghiệm thực tế.

**Nghiên cứu các thuật toán xử lý ảnh:** Các thuật toán này bao gồm việc phát triển các kỹ thuật phân tích hình ảnh như lọc ảnh, nhận diện cạnh, phát hiện đối tượng và phân loại hình ảnh. Những nghiên cứu này chủ yếu tập trung vào việc tối ưu hóa tốc độ xử lý, giảm độ phức tạp tính toán và cải thiện độ chính xác của thuật toán.

Ví dụ: Các thuật toán nổi bật như Canny Edge Detection, Histogram Equalization, Convolutional Neural Networks (CNNs) và Image Segmentation Algorithms đang được nghiên cứu và ứng dụng rộng rãi.

**Ứng dụng và giao diện người dùng**: Một phần nghiên cứu quan trọng khác là phát triển giao diện người dùng (UI) dễ sử dụng, cho phép người dùng tương tác hiệu quả với phần mềm xử lý ảnh. Các nghiên cứu này giúp xác định cách thức thiết kế phần mềm sao cho có thể dễ dàng tích hợp các công cụ xử lý ảnh vào ứng dụng thực tế mà không làm giảm trải nghiệm người dùng.

**Thử Nghiệm :** Sau khi xây dựng một phiên bản cơ bản của phần mềm, tiến hành thử nghiệm với người dùng để đảm bảo rằng giao diện người dùng dễ sử dụng và thân thiện với người dùng. Điều này bao gồm việc kiểm tra khả năng điều hướng, giao diện đồ họa và phản hồi của phần mềm khi người dùng tương tác với các công cụ xử lý ảnh.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**

## 3.1 Phân tích yêu cầu và tính năng của phần mềm

## 3.1.1 Mục tiêu và tính năng của phần mềm

*3.1.1.1 Mục tiêu của phần mềm*

**Xử lý ảnh hiệu quả**: Phần mềm được phát triển nhằm xử lý ảnh từ các nguồn đầu vào khác nhau, thực hiện các phép biến đổi cơ bản như làm sắc nét, làm mờ, chuyển đổi màu sắc và cắt xén ảnh.

**Dễ dàng sử dụng**: Phần mềm hướng đến sự đơn giản trong giao diện, giúp người dùng dễ dàng thao tác mà không cần kiến thức chuyên sâu về lập trình hoặc xử lý ảnh.

**Hiệu suất cao**: Đảm bảo phần mềm có thể xử lý ảnh một cách nhanh chóng và hiệu quả, ngay cả với các bức ảnh có độ phân giải cao.

**Các chức năng chính**:

Tải và lưu ảnh: Người dùng có thể tải lên ảnh từ các tệp hoặc từ máy ảnh và lưu lại kết quả sau khi xử lý.

Chỉnh sửa ảnh cơ bản: Phần mềm cung cấp các công cụ như cắt xén, thay đổi độ sáng, độ tương phản, điều chỉnh màu sắc, chuyển đổi ảnh thành đen trắng, làm sắc nét hoặc làm mờ.

Áp dụng bộ lọc ảnh: Các bộ lọc giúp người dùng dễ dàng tạo ra các hiệu ứng đặc biệt như làm mờ nền, biến đổi phong cách nghệ thuật, hoặc tạo ra các hiệu ứng vintage.

Cải thiện chất lượng ảnh: Chức năng làm rõ các chi tiết trong ảnh, khử nhiễu, và phục hồi ảnh chất lượng thấp.

## 3.1.2 Các chức năng cơ bản

Di chuyển và phóng to ảnh: Người dùng có thể di chuyển và phóng to/thu nhỏ ảnh để dễ dàng chỉnh sửa chi tiết.

Công cụ chỉnh sửa hình ảnh: Các công cụ như chỉnh độ sáng, tương phản, độ bão hòa màu, và cắt xén ảnh để thay đổi kích thước.

Áp dụng bộ lọc ảnh: Bộ lọc như làm mờ, biến đổi màu sắc, vẽ tranh, và các bộ lọc nghệ thuật khác.

## 3.1.3 Thiết kế phần mềm

### 3.2.1 Thiết kế giao diện người dùng (UI)

## Giao diện chính:

Thanh menu: Các tùy chọn như Tệp (File), Chỉnh sửa (Edit), Xem (View), và Công cụ (Tools) giúp người dùng dễ dàng truy cập vào các chức năng chính của phần mềm.

Không gian làm việc: Một khu vực chính để hiển thị ảnh, nơi người dùng có thể thao tác trực tiếp với ảnh như cắt, xoay, hoặc phóng to.

Bảng điều khiển công cụ: Các công cụ chỉnh sửa ảnh nằm trong bảng điều khiển bên trái hoặc phải, cho phép người dùng chọn các công cụ như làm sắc nét, làm mờ, hoặc áp dụng bộ lọc.

**Chức năng xử lý ảnh:**

Tải ảnh: Người dùng có thể tải ảnh từ máy tính hoặc kéo thả ảnh vào phần mềm.

Chỉnh sửa cơ bản: Các công cụ cho phép người dùng thay đổi độ sáng, độ tương phản, độ bão hòa của ảnh. Các nút bấm trực quan giúp chỉnh sửa dễ dàng.

Áp dụng hiệu ứng: Các bộ lọc hiệu ứng như vẽ tranh, chuyển ảnh thành đen trắng hoặc hiệu ứng cổ điển có thể được áp dụng bằng cách nhấp vào các nút chức năng trong bảng điều khiển.

Cải thiện chất lượng ảnh: Tích hợp các thuật toán để cải thiện chất lượng ảnh, giảm nhiễu, và làm sắc nét các chi tiết.

**Thiết kế kiến trúc phần mềm:**

Model: Các lớp xử lý dữ liệu ảnh, thực hiện các phép toán như biến đổi màu sắc, làm sắc nét, làm mờ, v.v.

View: Giao diện người dùng (UI), nơi người dùng có thể tương tác trực tiếp với ảnh và các công cụ.

Controller: Điều khiển sự tương tác giữa người dùng và phần mềm, nhận yêu cầu từ giao diện người dùng và cập nhật mô hình ảnh tương ứng.

**Các thư viện sử dụng:** Qt dùng cho việc xây dựng giao diện người dùng và xử lý sự kiện.

OpenCV: Dùng cho các thuật toán xử lý ảnh phức tạp như làm mờ, phát hiện biên, và các kỹ thuật nâng cao khác.

**Kiểm thử:** Phần mềm sẽ được kiểm tra kỹ lưỡng với nhiều loại ảnh khác nhau (ảnh đen trắng, ảnh màu, ảnh có độ phân giải cao, v.v.) để đảm bảo các chức năng xử lý ảnh hoạt động chính xác.

Các tình huống thử nghiệm bao gồm việc tải ảnh, chỉnh sửa, áp dụng hiệu ứng, và lưu kết quả.

**Tối ưu hóa hiệu suất:** Sử dụng các thuật toán tối ưu để giảm thời gian xử lý ảnh, đặc biệt là đối với ảnh có độ phân giải lớn.Sử dụng bộ nhớ hiệu quả để xử lý ảnh mà không gặp phải vấn đề tràn bộ nhớ.

**CHƯƠNG 4 GIAO DIỆN** **PHẦN MỀM**

## 4.1 Giao diện chính của phần mềm

A screenshot of a computer

Description automatically generated

Hình 4. 1 Giao diện chính

**Màn hình chính:**

**File: Chứa các lựa chọn liên quan đến thao tác với tệp, như mở tệp ảnh, lưu và đóng tệp.**

**Ánh sáng: Chứa các tùy chọn điều chỉnh độ sáng hoặc các bộ lọc ánh sáng cho ảnh.**

**Biểu đồ: Liên quan đến việc hiển thị các biểu đồ như biểu đồ histograms để phân tích ảnh.**

**Biên: Giúp phát hiện ra đường biên của một tấm ảnh và lưu ảnh vào project.**

**Mờ ảnh: Làm mờ tấm ảnh trắng và đen.**

**Giao diện này cung cấp một số công cụ xử lý ảnh cơ bản đến nâng cao, cho phép người dùng thực hiện các tác vụ như xử lý ảnh xám, ảnh màu, phân ngưỡng, cải thiện chất lượng ảnh, và áp dụng các bộ lọc như Sobel, Robert, Prewitt, và Laplace.**

**Nhưng do khi thực thi phần mềm khiến cho các nút chức năng bị xáo trộn thì tôi đã gôm chu****ng lại trên thanh menubar.**

## 4.2 Giao diện khi thực thi phần mềm

A screenshot of a computer

Description automatically generated

Hình 4. 2 Giao diện khi thực thi

**File:** có chức năng mở ảnh , lưu ảnh và thoát ứng dụng được gán bởi các phím tắt khác nhau.

Mở ảnh (Ctrl + O)

Lưu ảnh (Ctrl + S)

Thoát chương trình (Ctrl + X)

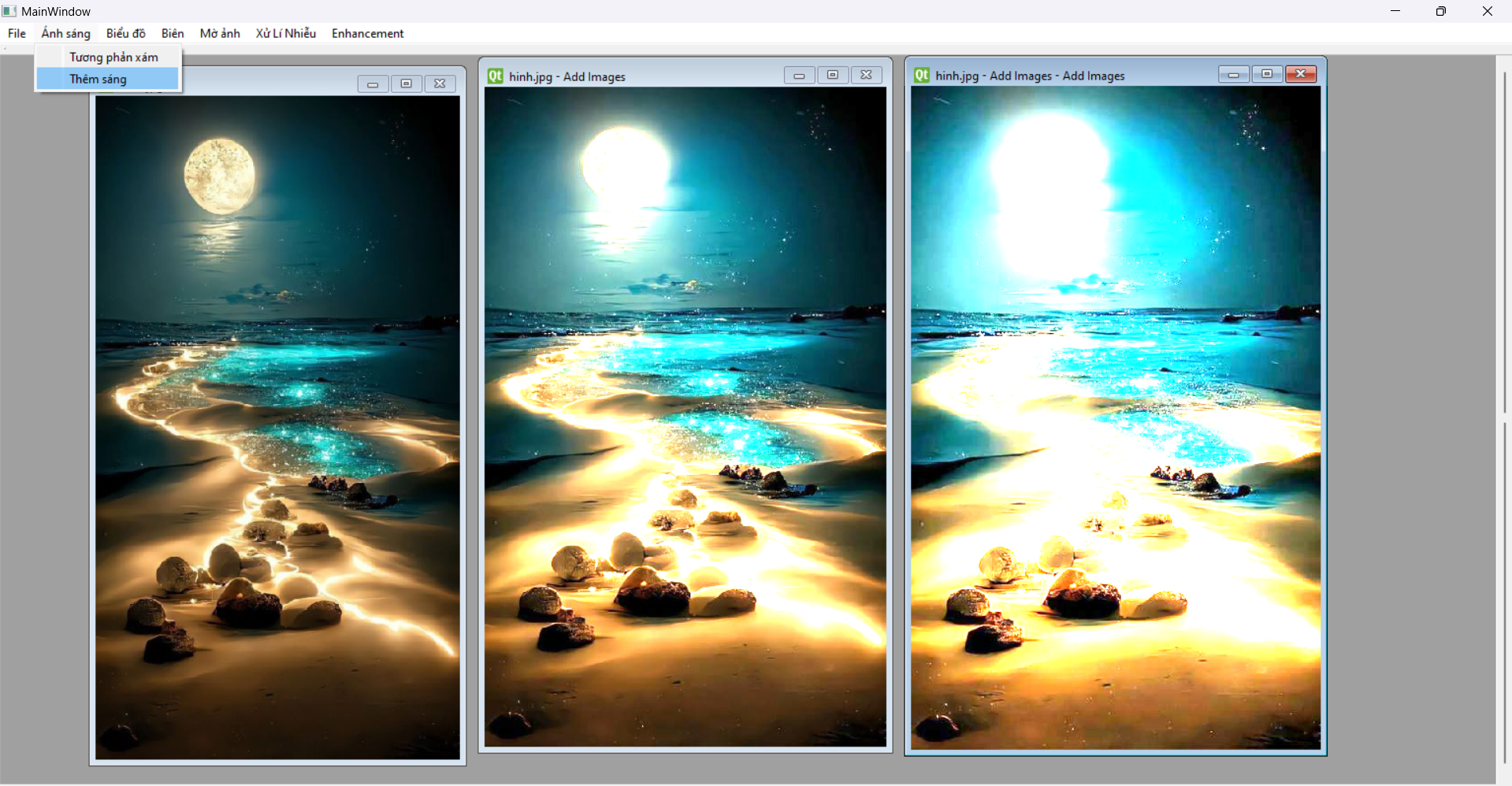
A screenshot of a computer

Description automatically generated

Hình 4. 3 Giao diện khi mở ảnh

**Chức năng ánh sáng:** Gồm 2 chức năng là tương phản xám và thêm sáng, đối với tương phản xám thì ảnh được thêm vào phải ở định dạng là ảnh xám thì chức năng mới có thể thực thi được còn ngược lại nếu ảnh đó là ảnh màu thì hệ thống sẽ thông báo lỗi “ảnh phải ở định dạng ảnh xám”.

Đối với chức năng thêm sáng thì sẽ được cấu hình ở mức cố định , có nghĩa là khi bấm vào chức năng thì hệ thống sẽ tự động tăng sáng cho bạn và khi bạn ấn một lần nữa vào chức năng thì hệ thống lại tiếp tục tăng sáng dần cho bạn .



Hình 4. 3 Giao diện khi dùng chức năng(thêm sáng)

**Biểu đồ:**

**Hiển thị biểu đồ (ảnh xám):** giúp phân tích và đánh giá đặc trưng của hình ảnh xám

Phân phối mức xám: Biểu đồ cho thấy sự phân bố các mức độ xám (0-255) trong hình ảnh. Trục ngang thể hiện các giá trị mức xám, còn trục dọc là tần suất xuất hiện của mỗi giá trị

**A screenshot of a computer

Description automatically generated**

Hình 4. 3 Giao diện biểu đồ ảnh xám

**Hiển thị biểu đồ (ảnh màu):** giúp phân tích và đánh giá đặc trưng của hình ảnh màu bằng cách biểu diễn thông tin màu sắc trong các kênh màu (thường là RGB: Red, Green, Blue).

Phân phối màu sắc:  
Biểu đồ màu cho biết sự phân bố của từng thành phần màu (R, G, B) trong hình ảnh.

Trục ngang: Giá trị cường độ màu (0-255).

Trục dọc: Tần suất xuất hiện của mỗi giá trị màu.

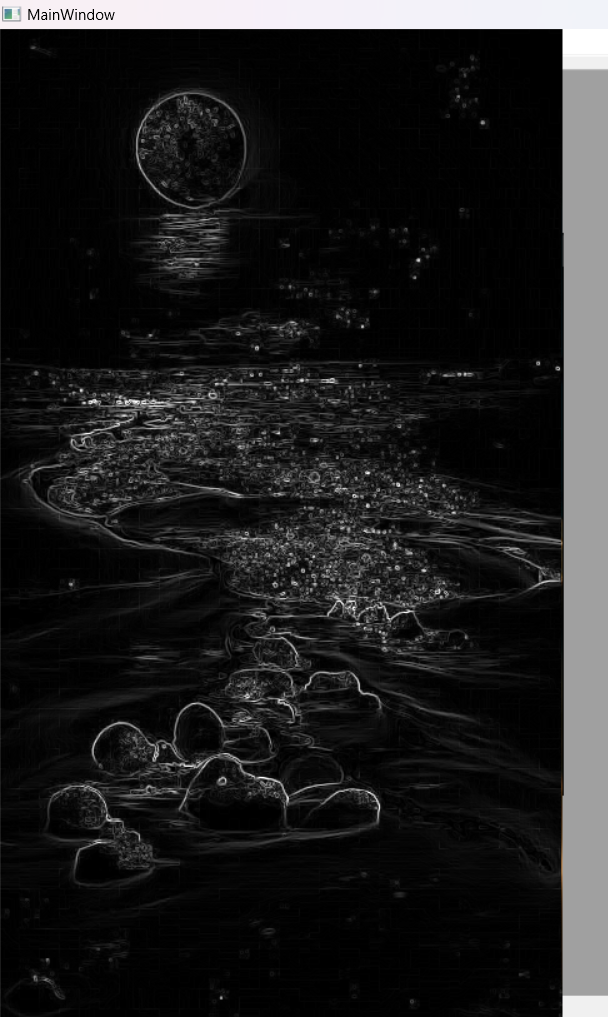
A screenshot of a computer

Description automatically generated

*Hình 4. 3 Giao diện biểu đồ ảnh màu*

**Biên** (Phát hiện biên): là một tính năng quan trọng trong xử lý ảnh, có nhiệm vụ xác định và làm nổi bật các đường biên (rìa) trên hình ảnh. Sau đó tấm ảnh sẽ được lưu vào thư mục của dự án.

Phát hiện biên là bước quan trọng trong xử lý ảnh và thị giác máy tính, giúp tăng cường khả năng nhận diện và phân tích hình ảnh một cách hiệu quả.



Hình 4. 3 Giao diện phát hiện biên

**Mờ ảnh:**

Mờ ảnh sáng là một thao tác trong xử lý ảnh có nhiệm vụ làm giảm độ sắc nét và chi tiết của hình ảnh bằng cách thay đổi cường độ sáng của các điểm ảnh.

Giảm các chi tiết nhỏ hoặc các điểm nhiễu không mong muốn trên ảnh.

Thích hợp khi cần làm sạch hình ảnh trước khi thực hiện các bước xử lý tiếp theo

A screenshot of a computer

Description automatically generated

*Hình 4. 3 Giao diện mờ ảnh sáng*

Mờ ảnh đen là một thao tác xử lý ảnh nhằm làm giảm độ sắc nét và chi tiết của hình ảnh, đặc biệt với các hình ảnh có tông màu tối hoặc đen. Chức năng này thường được thực hiện bằng các bộ lọc mờ làm thay đổi giá trị điểm ảnh trong khu vực lân cận

A screenshot of a computer

Description automatically generated

Hình 4. 3 Giao diện mờ ảnh đen

# CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

## 5.1 Kết luận

Phần mềm xử lý ảnh được phát triển trên nền tảng Qt Creator đã hoàn thành các chức năng cơ bản và đáp ứng được mục tiêu đề ra ban đầu. Phần mềm cho phép người dùng thực hiện các thao tác xử lý ảnh nhanh chóng và hiệu quả, từ các tính năng đơn giản như chuyển ảnh xám, âm bản, đến các tính năng nâng cao như lọc nhiễu, phát hiện biên và cân bằng histogram.

Các tính năng chính bao gồm:

Xử lý ảnh xám và ảnh màu với các thao tác cơ bản như thay đổi độ sáng, độ tương phản.

Lọc nhiễu và làm mờ ảnh thông qua các bộ lọc trung bình, lọc trung vị và Gaussian.

Phát hiện biên bằng các thuật toán nổi tiếng như Sobel, Robert, Prewitt, Laplace và gradient.

Biểu đồ histogram: Hiển thị và cân bằng biểu đồ để cải thiện chất lượng ảnh.

Xử lý ảnh nâng cao: Các thuật toán như phân ngưỡng (Otsu) và các phép giãn nở, ăn mòn trong xử lý ảnh nhị phân.

Phần mềm giúp người dùng có thể nhanh chóng thực hiện các thao tác xử lý ảnh cơ bản, hỗ trợ tốt trong việc học tập, nghiên cứu và thực hành kỹ thuật xử lý ảnh. Giao diện trực quan, dễ sử dụng, kết hợp với khả năng xử lý thời gian thực, đã tạo nên một công cụ hiệu quả và dễ tiếp cận cho người dùng.

## 5.2 Định hướng phát triển

### 5.2.1 Mở rộng tính năng xử lý ảnh nâng cao

Tích hợp thêm các thuật toán xử lý ảnh nâng cao như phát hiện đối tượng, nhận diện khuôn mặt và phân loại ảnh dựa trên các mô hình học máy (Machine Learning).

Phát triển các thuật toán lọc nhiễu hiện đại như Bilateral Filter hoặc lọc thích nghi để nâng cao chất lượng ảnh xử lý.

Áp dụng biến đổi Fourier để xử lý ảnh trong miền tần số, phục vụ cho các ứng dụng như làm sắc nét ảnh hoặc khử nhiễu tần số cao.

### 5.2.2 Tích hợp các kỹ thuật phân loại và nhận dạng ảnh

Tích hợp khả năng phân loại ảnh thông qua các thư viện như OpenCV và kết nối với các mô hình trí tuệ nhân tạo (AI).

Phát triển chức năng nhận diện ký tự quang học (OCR) để đọc văn bản từ ảnh, hỗ trợ trong các bài toán tự động hóa văn bản.

### 5.2.3 Tăng cường giao diện người dùng (UI/UX)

Nâng cấp giao diện phần mềm trở nên thân thiện và trực quan hơn bằng cách sử dụng Qt Designer với các bảng điều khiển linh hoạt và trực quan.

Cải thiện trải nghiệm người dùng khi thao tác, như cung cấp các công cụ xem trước (preview) kết quả xử lý ảnh theo thời gian thực.

### 5.2.4 Xử lý ảnh thời gian thực

Tích hợp khả năng xử lý ảnh từ camera hoặc video trực tuyến để áp dụng các kỹ thuật xử lý ảnh thời gian thực.

Áp dụng các bộ lọc thích ứng để cải thiện chất lượng hình ảnh trong các điều kiện ánh sáng kém.

Hỗ trợ nhiều định dạng và chức năng lưu trữ ảnh

Bổ sung khả năng xử lý và lưu ảnh dưới các định dạng khác nhau như PNG, JPEG, BMP, TIFF,...

Cho phép xuất kết quả dưới dạng báo cáo hoặc các file có chứa thông tin histogram và thông số xử lý.

### 5.2.5 Kết hợp tính năng đồ họa và hiển thị nâng cao

Cải tiến hệ thống biểu đồ histogram để người dùng có thể điều chỉnh trực tiếp và tương tác với biểu đồ.

Phát triển các công cụ vẽ và chú thích trực tiếp trên hình ảnh để phục vụ trong các bài toán phân tích ảnh.

### 5.2.6 Tích hợp tính năng đa nền tảng và mở rộng

Tối ưu hóa phần mềm để hoạt động mượt mà trên nhiều hệ điều hành khác nhau như Windows, Linux và MacOS.

Phát triển các tính năng mở rộng qua plugin, cho phép người dùng tích hợp thêm các thuật toán và công cụ tùy chỉnh.

Việc phát triển phần mềm xử lý ảnh trên Qt Creator đã đạt được những kết quả tích cực và cung cấp nhiều công cụ hữu ích cho người dùng. Trong tương lai, việc mở rộng các tính năng nâng cao và cải thiện giao diện sẽ giúp phần mềm trở thành một công cụ mạnh mẽ, đáp ứng tốt các nhu cầu từ cơ bản đến phức tạp trong lĩnh vực xử lý ảnh.

**CHƯƠNG 6 TÀI LIỆU THAM KHẢO - PHỤ LỤC**

## 6. Tài liệu tham khảo

### 6.1 Xử lý ảnh và Thị giác máy tính

OpenCV Documentation - Hướng dẫn chi tiết về thư viện OpenCV dùng trong xử lý ảnh và thị giác máy tính. Cung cấp hướng dẫn triển khai và tích hợp OpenCV với các ứng dụng phần mềm https://docs.opencv.org

*Bradski, G., & Kaehler, A. Learning OpenCV: Computer Vision with the OpenCV Library.*

Đây là tài liệu toàn diện về OpenCV, bao gồm hướng dẫn cơ bản và nâng cao trong xử lý ảnh.

*Hands-On Image Processing with Python - Sandipan Dey*.

Tài liệu cấp nhật kiến thức từng bước trong xử lý ảnh: làm mờ, nhận diện biên, và xử lý histogram.

### 6.2 Qt Creator và Phát triển giao diện

Qt Documentation - Cung cấp các API chi tiết để tác nghiệp với giao diện, multimedia, và xử lý đa luồng trong *Qt. https://doc.qt.io*

*Blanchette, J., & Summerfield, M. C++ GUI Programming with Qt 5.*

Cuốn sách hướng dẫn chi tiết các tính năng trong Qt, từ thiết kế giao diện đến xử lý sự kiện và *multimedia*.

*Blanchette, J. The Art of Building Qt Applications*.Tập trung vào tối ưu giao diện người dùng, giúp ứng dụng vận hành nhanh hơn.

### 6.3 Tích hợp OpenCV và Qt

*Integrating OpenCV with Qt:* Hướng dẫn tích hợp OpenCV và Qt để xây dựng giao diện xử lý ảnh một cách trơn tru. Bài viết chi tiết về thiết lập và triển khai các API OpenCV trong *Qt. https://opencv.org/integration-with-qt*

*Image Processing with OpenCV and Qt:* Lồng ghép các module giao diện Qt với các chức năng làm mờ, lọc nhiễu và biển đổ histogram của OpenCV.

### 6.4 Bài viết nghiên cứu về xử lý ảnh

Phân tích *Histogram* ảnh trong Xử lý số: Nghiên cứu hướng dẫn làm việc với histogram để phân đoạn ảnh, tính tần suất và lồng ghép với giao diện đồ họa.

Kỹ thuật phát hiện biên trong xử lý ảnh: Bài viết trên Thị giác Máy Tính tác nghiệp với các thuật toán *Sobel, Canny và Laplacian* để phát hiện biên.

### 6.5 Nghiên cứu phát triển giao diện phần mềm xử lý ảnh

Phát triển phần mềm giao diện người dùng: Nghiên cứu ứng dụng Qt trong các phần mềm xử lý ảnh chuyên dụng như *PixInsight*.

PixInsight Documentation - Hệ thống xử lý ảnh chuyên dụng với các tính năng như khởi phục chi tiết ảnh, ghép kết histogram.

### 6.6 Hỗ trợ Công cụ

*Microsoft Visual Studio Documentation* - Tích hợp Qt và OpenCV trên Visual Studio.

*CMake Documentation* - Dùng CMake tối ưu hóa quy trình build và tích hợp thư viện.

*GitHub OpenCV Projects* - Bộ sưu tập các dự án xử lý ảnh sử dụng OpenCV.

### 6.7 Sách và tài liệu bổ sung

Gonzalez, R., & Woods, R. *Digital Image Processing* (ấn bản mới nhất). Cuốn sách nổi tiếng cung cấp kiến thức nền tảng và nâng cao về xử lý ảnh kỹ thuật số, bao gồm các thuật toán và kỹ thuật chính.

Laganiere, R. *OpenCV 4 Computer Vision Application Programming Cookbook*. Hướng dẫn chi tiết cho lập trình viên sử dụng OpenCV trong các ứng dụng thực tế.

Qt Contributors. *Mastering Qt 5*. Tài liệu nâng cao về Qt, bao gồm tối ưu hóa hiệu suất và tích hợp xử lý ảnh.