

[illegible]

Vĩnh Long, ngày ..... tháng ..... năm .....  
**Giáo viên hướng dẫn**  
*(Ký tên và ghi rõ họ tên)*

[illegible]

**Giáo viên phản biện**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến các thầy cô, những người đã hết mực chỉ bảo, hướng dẫn và sẵn sàng giải đáp thắc mắc, cung cấp những kiến thức hữu ích giúp em hoàn thành công việc trong suốt quá trình thực hiện báo cáo này.

Mặc dù đã cố gắng để thực hiện đề tài một cách nghiêm túc và tập trung trong suốt quá trình thu thập, tìm hiểu, phân tích và tổng hợp nhiều nguồn tài liệu khác nhau, song do sự hiểu biết còn chưa cao, trình độ chuyên môn còn hạn chế ở nhiều mặt và chưa có nhiều kỹ năng trong việc thực hiện các công việc trong chuyên ngành nên trong quá trình biên soạn đồ án không thể tránh khỏi các thiết sót. Do đó nếu như có những điểm trong báo cáo còn chưa hợp lý hoặc sai sót em rất mong được các thầy cô đóng góp thêm các ý kiến để có thể học hỏi và tiếp thu thêm các kiến thức còn thiếu sót.

Em xin chân thành cảm ơn!

*Sinh viên ký và ghi rõ họ tên*

Hoàng Tuấn Kiệt

Nguyễn Đỗ Thành Lộc

Nguyễn Hoàng Phúc

## TÓM TẮT ĐỀ TÀI

Đề tài "Xây dựng hệ thống nén ảnh" được thực hiện nhằm giải quyết những hạn chế trong việc xử lý và tối ưu hóa dung lượng ảnh số thông qua các phương thức truyền thống như phần mềm chuyên nghiệp phức tạp hoặc các công cụ online yêu cầu upload ảnh lên server. Các phương thức này thường gặp khó khăn về chi phí bản quyền, lo ngại bảo mật dữ liệu và thiếu tính tiện lợi. Vì vậy, việc xây dựng một ứng dụng web nén ảnh hoạt động hoàn toàn trên trình duyệt là cần thiết để tạo ra môi trường xử lý nhanh chóng, bảo mật và hiệu quả.

Trong quá trình thực hiện, nhóm tập trung nghiên cứu các kỹ thuật nén ảnh hiện đại (JPEG, PNG, WebP), phân tích nhu cầu người dùng về tối ưu dung lượng ảnh cho web, mạng xã hội và lưu trữ. Trên cơ sở đó, đề tài xác định các chức năng chính gồm tải ảnh lên, điều chỉnh chất lượng nén linh hoạt, so sánh kết quả trước/sau và tải xuống ảnh đã nén. Giải pháp được đề xuất là xây dựng ứng dụng web với các tính năng: tải và hiển thị ảnh song song; nén ảnh với 3 định dạng (JPEG lossy, PNG lossless, WebP); điều chỉnh chất lượng bằng slider (0-100%); so sánh chi tiết dung lượng, tỷ lệ nén; và tải xuống ảnh đã xử lý. Hệ thống sử dụng HTML5 Canvas API, JavaScript ES6+ và CSS3 với thiết kế responsive, xử lý hoàn toàn trên client-side không cần backend.

Kết quả đạt được là một ứng dụng web nén ảnh hoàn chỉnh với giao diện trực quan chuyên nghiệp, hoạt động ổn định trên mọi trình duyệt hiện đại, cho phép nén ảnh hiệu quả (giảm 80-90% dung lượng với JPEG, 87% với WebP), xử lý nhanh chóng (< 1 giây) và đảm bảo bảo mật tuyệt đối vì ảnh không rời khỏi máy tính người dùng. Đề tài góp phần số hóa quy trình tối ưu ảnh, giúp người dùng tiết kiệm dung lượng lưu trữ, băng thông và nâng cao hiệu quả sử dụng ảnh số trong thời đại công nghệ.

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	10
1.1 Giới thiệu chung .....	10
1.2 Vấn đề đặt ra .....	10
1.3 Mục tiêu đề tài .....	11
1.4 Phạm vi và đối tượng nghiên cứu .....	12
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	14
2.1 Khái niệm ảnh số .....	14
2.1.1 Định nghĩa ảnh số .....	14
2.2.2 Biểu diễn ảnh số .....	14
2.2 Khái niệm nén ảnh .....	15
2.2.1 Định nghĩa nén ảnh .....	15
2.2.2. Mục đích của nén ảnh .....	16
2.3. Phân loại nén ảnh .....	17
2.3.1. Nén không mất dữ liệu (Lossless Compression) .....	17
2.3.2. Nén có mất dữ liệu (Lossy Compression) .....	17
2.4. Một số phương pháp nén ảnh phổ biến .....	18
2.4.1. Run-Length Encoding(RLE) .....	18
2.4.2 Huffman Coding .....	19
2.4.3. JPEG (Joint Photographic Experts Group) .....	19
2.4.4. PNG (Portable Network Graphics) .....	20
2.5. Các tiêu chí đánh giá hiệu quả nén .....	21
2.5.1. Tỷ lệ nén (Compression Ratio) .....	21
2.5.2. Dung lượng ảnh .....	21
2.5.3. Chất lượng ảnh .....	22
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM .....	24
3.1. Môi trường thực nghiệm .....	24
3.1.1. Công nghệ sử dụng .....	24
3.2. Định dạng ảnh đầu vào .....	24
3.3. Quá trình thực nghiệm .....	25
3.3.1. Quy trình nén ảnh .....	25
3.3.2. Các thông số thử nghiệm .....	25
3.4. Kết quả thu được .....	25

3.4.1. Kết quả nén JPEG .....	26
3.4.2. Kết quả nén PNG .....	26
3.4.3. Kết quả nén WebP .....	26
3.5. Đánh giá và nhận xét .....	27
3.5.1. Đánh giá hiệu quả nén .....	27
3.5.2. Khuyến nghị sử dụng .....	28
3.5.3. Hiệu năng ứng dụng .....	28
CHƯƠNG 4: KẾT LUẬN .....	30
4.1 Kết quả đạt được .....	30
4.2. Ưu điểm của ứng dụng .....	30
4.3. Hạn chế và khó khăn .....	31
4.4. Hướng phát triển .....	31
4.5. Kết luận chung .....	32
DANH MỤC TÀI LIỆU THAM KHẢO .....	34

## DANH MỤC HÌNH ẢNH

Hình 2.1: Biểu diễn pixel và mô hình màu RGB trong ảnh số .....	15
Hình 2.2: Sơ đồ so sánh nén Lossy và Lossless .....	16
Hình 2.3: So sánh chất lượng và dung lượng giữa nén Lossy và Lossless .....	18
Hình 2.4: Quy trình nén JPEG từ RGB đến Huffman Coding .....	20
Hình 2.5: Biểu đồ so sánh tỷ lệ nén giữa JPEG, PNG và WebP .....	23
Hình 3.1: Giao diện ứng dụng nén ảnh trực tuyến .....	24
Hình 3.2: Sơ đồ luồng xử lý ảnh với Canvas API .....	25
Hình 3.3: Bảng so sánh kết quả nén giữa JPEG, PNG và WebP .....	27

### MỞ ĐẦU

Trong bối cảnh công nghệ số phát triển mạnh mẽ, hình ảnh ngày càng đóng vai trò quan trọng trong giao tiếp và chia sẻ thông tin trực tuyến. Tuy nhiên, ảnh chất lượng cao từ smartphone và máy ảnh hiện đại thường có dung lượng rất lớn (5-20 MB/ảnh), gây khó khăn trong việc lưu trữ, chia sẻ qua email và đăng tải lên mạng xã hội. Phần lớn các phần mềm xử lý ảnh chuyên nghiệp như Adobe Photoshop đều là sản phẩm thương mại với chi phí cao và giao diện phức tạp, chưa phù hợp cho người dùng phổ thông. Các công cụ nén ảnh online hiện nay thường yêu cầu upload ảnh lên server, gây lo ngại về bảo mật dữ liệu. Việc xây dựng một ứng dụng nén ảnh hoạt động hoàn toàn trên trình duyệt bằng HTML5 Canvas API và JavaScript sẽ giúp người dùng xử lý ảnh nhanh chóng, bảo mật mà không cần cài đặt phần mềm. Vì vậy, em lựa chọn đề tài "Xây dựng hệ thống nén ảnh" để nghiên cứu và phát triển.

Mục tiêu của dự án là xây dựng một hệ thống nén ảnh hoàn chỉnh, hỗ trợ nén ảnh với ba định dạng (JPEG, PNG, WebP), điều chỉnh chất lượng nén linh hoạt, so sánh kết quả trước/sau nén và tải xuống ảnh đã xử lý, với giao diện hiện đại và thân thiện. Ứng dụng được thiết kế theo kiến trúc client-side processing để đảm bảo bảo mật tuyệt đối, không cần backend server, đồng thời áp dụng các công nghệ như HTML5 Canvas API cho xử lý ảnh, JavaScript ES6+ cho logic nghiệp vụ và CSS3 cho giao diện responsive.

Phạm vi nghiên cứu tập trung vào xây dựng ứng dụng frontend hoàn chỉnh, bao gồm xử lý ảnh bằng Canvas API, nén ảnh với ba kỹ thuật (JPEG lossy, PNG lossless, WebP), tính toán tỷ lệ nén và thiết kế giao diện responsive. Ứng dụng hỗ trợ các định dạng ảnh phổ biến, cho phép điều chỉnh chất lượng qua slider và hiển thị so sánh chi tiết. Thông qua dự án, sinh viên có thể tiếp cận kiến thức về xử lý ảnh số, thuật toán nén, Canvas API và best practices trong phát triển ứng dụng web hiện đại.



## LÝ DO CHỌN ĐỀ TÀI

Trong thời đại công nghệ số phát triển nhanh chóng, nhu cầu xử lý và tối ưu hóa hình ảnh ngày càng gia tăng. Ảnh số được sử dụng rộng rãi trong mọi lĩnh vực từ mạng xã hội, thương mại điện tử đến giáo dục và y tế. Tuy nhiên, ảnh chất lượng cao từ các thiết bị hiện đại thường có dung lượng rất lớn, gây khó khăn trong việc lưu trữ, chia sẻ và tải lên internet. Đa số các phần mềm xử lý ảnh chuyên nghiệp như Adobe Photoshop, GIMP đều là sản phẩm thương mại hoặc có giao diện phức tạp với nhiều tính năng nâng cao và chi phí cao, gây khó khăn cho người dùng phổ thông trong việc tiếp cận và sử dụng. Các công cụ nén ảnh online hiện nay thường yêu cầu upload ảnh lên server, gây lo ngại về bảo mật và quyền riêng tư dữ liệu cá nhân.

Bên cạnh đó, xử lý ảnh trên web là một trong những lĩnh vực quan trọng và ngày càng phổ biến trong phát triển ứng dụng web hiện đại. Việc nắm vững các công nghệ như HTML5 Canvas API, JavaScript ES6+ và các kỹ thuật nén ảnh (JPEG, PNG, WebP) không chỉ giúp sinh viên củng cố kiến thức về lập trình web mà còn nâng cao khả năng xây dựng các ứng dụng xử lý dữ liệu phức tạp hoàn toàn trên trình duyệt mà không cần backend server. Việc hiểu rõ về Canvas API, File API, Blob và các thuật toán nén dữ liệu là nền tảng quan trọng cho nhiều ứng dụng web hiện đại. Tuy nhiên, cơ hội để thực hành và triển khai một ứng dụng xử lý ảnh hoàn chỉnh từ đầu trong quá trình học tập vẫn còn hạn chế.

Xuất phát từ những lý do trên, em đã lựa chọn đề tài "Xây dựng hệ thống nén ảnh" nhằm áp dụng kiến thức đã học vào thực tế, đồng thời tìm hiểu sâu hơn về xử lý ảnh số, các kỹ thuật nén dữ liệu và các vấn đề liên quan đến hiệu năng, bảo mật và trải nghiệm người dùng trong ứng dụng web. Thông qua đề tài này, em mong muốn nâng cao kỹ năng lập trình, tích lũy kinh nghiệm thực tiễn về xử lý ảnh và tạo nền tảng cho việc nghiên cứu, phát triển các ứng dụng web xử lý dữ liệu multimedia trong tương lai.

## CHƯƠNG 1: TỔNG QUAN

### 1.1 Giới thiệu chung

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, ảnh số đóng vai trò quan trọng và không thể thiếu trong nhiều lĩnh vực của đời sống. Trên internet và mạng xã hội, hàng tỷ hình ảnh được chia sẻ mỗi ngày trên các nền tảng như Facebook, Instagram, Twitter, TikTok, phục vụ nhu cầu giao tiếp, giải trí và chia sẻ thông tin của người dùng toàn cầu. Trong lĩnh vực y tế, ảnh số từ các thiết bị chẩn đoán hình ảnh như X-quang, CT, MRI đóng vai trò then chốt trong việc chẩn đoán và điều trị bệnh. Công nghệ vệ tinh sử dụng ảnh số để quan sát Trái Đất, dự báo thời tiết, giám sát môi trường và quản lý tài nguyên thiên nhiên. Các hệ thống lưu trữ dữ liệu doanh nghiệp, thư viện số và kho lưu trữ đa phương tiện cũng phụ thuộc rất nhiều vào khả năng quản lý và lưu trữ ảnh số hiệu quả.

Tuy nhiên, ảnh số chất lượng cao thường có dung lượng rất lớn, đặc biệt là ảnh từ các thiết bị hiện đại với độ phân giải cao. Một bức ảnh chụp từ smartphone hiện đại có thể có dung lượng từ 5 đến 20 MB, trong khi ảnh từ máy ảnh chuyên nghiệp có thể lên đến hàng chục MB. Dung lượng lớn này gây ra nhiều vấn đề thực tế: tốn kém bộ nhớ lưu trữ trên thiết bị và cloud, tiêu tốn băng thông mạng khi truyền tải, làm tăng thời gian tải trang web ảnh hưởng đến trải nghiệm người dùng, và tăng chi phí vận hành cho các dịch vụ lưu trữ và phân phối nội dung.

Nén ảnh là giải pháp công nghệ thiết yếu để giải quyết các vấn đề trên. Mục tiêu của nén ảnh là giảm đáng kể dung lượng file trong khi vẫn duy trì chất lượng hình ảnh ở mức chấp nhận được, phù hợp với mục đích sử dụng. Các kỹ thuật nén ảnh hiện đại đã đạt được những tiến bộ đáng kể, cho phép giảm dung lượng lên đến 90% mà vẫn giữ được chất lượng hình ảnh tốt. Việc áp dụng nén ảnh không chỉ mang lại lợi ích kinh tế mà còn góp phần giảm tải cho hạ tầng internet toàn cầu và tiết kiệm năng lượng cho các trung tâm dữ liệu.

### 1.2 Vấn đề đặt ra

Bài toán nén ảnh đặt ra nhiều câu hỏi nghiên cứu quan trọng cần được giải quyết. Câu hỏi đầu tiên và cơ bản nhất là làm thế nào để giảm kích thước file ảnh một cách hiệu quả. Điều này đòi hỏi phải nghiên cứu và lựa chọn các thuật toán nén phù hợp, có khả năng loại bỏ dữ liệu dư thừa hoặc ít quan trọng trong ảnh mà không ảnh hưởng đáng kể đến chất lượng hình ảnh nhìn thấy được.

Câu hỏi thứ hai liên quan đến đánh giá chất lượng ảnh sau khi nén. Làm sao để đo lường một cách khách quan mức độ suy giảm chất lượng và xác định ngưỡng chấp nhận được? Các phương pháp đánh giá có thể dựa trên các chỉ số toán học như PSNR (Peak Signal-to-Noise Ratio), MSE (Mean Squared Error), hoặc dựa trên đánh giá chủ quan của con người. Việc cân bằng giữa tỷ lệ nén và chất lượng hình ảnh là một thách thức quan trọng, đòi hỏi sự hiểu biết sâu sắc về cả lý thuyết kỹ thuật và nhu cầu thực tế của người dùng.

Câu hỏi thứ ba là so sánh hiệu quả giữa các phương pháp nén khác nhau. Hiện nay có nhiều định dạng và thuật toán nén ảnh khác nhau như JPEG, PNG, WebP, AVIF, mỗi phương pháp có ưu điểm và nhược điểm riêng. Cần có nghiên cứu thực nghiệm để đánh giá và so sánh các phương pháp này trên nhiều tiêu chí: tỷ lệ nén, chất lượng hình ảnh, tốc độ xử lý, khả năng tương thích với các hệ thống và trình duyệt. Từ đó có thể đưa ra khuyến nghị về phương pháp nén phù hợp cho từng loại ảnh và mục đích sử dụng cụ thể.

Ngoài ra, còn có vấn đề về tính thực tế và khả năng ứng dụng. Làm sao để xây dựng một công cụ nén ảnh dễ sử dụng, không yêu cầu kiến thức chuyên môn, có thể hoạt động trực tiếp trên trình duyệt web mà không cần cài đặt phần mềm, và đảm bảo tính bảo mật cho dữ liệu người dùng? Đây là những thách thức thực tế cần được giải quyết để đưa nghiên cứu lý thuyết vào ứng dụng thực tiễn.

### 1.3 Mục tiêu đề tài

Đề tài đặt ra ba mục tiêu chính cần đạt được. Mục tiêu thứ nhất là tìm hiểu và nghiên cứu sâu về các kỹ thuật nén ảnh được sử dụng rộng rãi trong thực tế hiện nay. Điều này bao gồm việc nghiên cứu cơ sở lý thuyết của các thuật toán nén, hiểu rõ nguyên lý hoạt động, ưu nhược điểm của từng phương pháp. Đặc biệt tập trung vào ba định dạng phổ biến: JPEG với kỹ thuật nén có mất dữ liệu dựa trên biến đổi Cosine rời rạc (DCT), PNG với kỹ thuật nén không mất dữ liệu sử dụng thuật toán DEFLATE, và WebP là định dạng hiện đại của Google kết hợp ưu điểm của cả hai phương pháp trên.

Mục tiêu thứ hai là xây dựng một hệ thống ứng dụng nén ảnh hoàn chỉnh và có tính thực tế cao. Hệ thống cần có giao diện thân thiện, dễ sử dụng, cho phép người dùng tải ảnh lên, lựa chọn phương pháp nén và mức chất lượng mong muốn, sau đó tải xuống ảnh đã được nén. Đặc biệt, ứng dụng được thiết kế hoạt động hoàn toàn trên trình duyệt web (client-side) sử dụng HTML5 Canvas API và JavaScript, không cần backend server, đảm bảo tốc độ xử lý nhanh và bảo mật tuyệt đối vì ảnh không bao giờ rời khỏi máy tính người dùng.

Mục tiêu thứ ba là thực hiện so sánh và đánh giá hiệu quả của các phương pháp nén ảnh thông qua thực nghiệm. Sử dụng các bộ ảnh test đa dạng về nội dung (ảnh chụp, đồ họa, logo) và kích thước, tiến hành nén với các phương pháp khác nhau và thu thập dữ liệu về tỷ lệ nén, dung lượng file, chất lượng hình ảnh. Từ kết quả thực nghiệm, phân tích và đưa ra nhận xét khách quan về hiệu quả của từng phương pháp, xác định phương pháp tối ưu cho từng loại ảnh và trường hợp sử dụng cụ thể.

### 1.4 Phạm vi và đối tượng nghiên cứu

Phạm vi nghiên cứu của đề tài được giới hạn rõ ràng để đảm bảo tính tập trung và khả thi. Đối tượng nghiên cứu chính là ảnh số 2D tĩnh, bao gồm các định dạng phổ biến như JPEG, PNG, WebP và GIF. Đề tài không đi sâu vào xử lý video, ảnh động phức tạp hay ảnh 3D, mà tập trung vào các kỹ thuật nén ảnh tĩnh được sử dụng rộng rãi trong thực tế.

Về mặt kỹ thuật, đề tài tập trung nghiên cứu ba phương pháp nén chính. Thứ nhất là nén có mất dữ liệu (lossy compression) với đại diện điển hình là JPEG, phương pháp này chấp nhận mất một phần thông tin ảnh để đạt tỷ lệ nén cao. Thứ hai là nén không mất dữ liệu (lossless compression) với đại diện là PNG, phương pháp này đảm bảo ảnh sau khi giải nén giống hệt ảnh gốc. Thứ ba là WebP, định dạng hiện đại hỗ trợ cả hai phương pháp nén trên, được tối ưu đặc biệt cho ứng dụng web.

Về công nghệ triển khai, đề tài sử dụng các công nghệ web chuẩn và hiện đại. HTML5 Canvas API được sử dụng làm nền tảng để xử lý và hiển thị ảnh trên trình duyệt. JavaScript ES6+ được dùng để lập trình logic xử lý, điều khiển giao diện và tương tác người dùng. CSS3 được áp dụng để thiết kế giao diện responsive, đảm bảo ứng dụng hoạt động tốt trên mọi kích thước màn hình từ desktop đến mobile. Đặc biệt, toàn bộ quá trình xử lý diễn ra trên client-side, không sử dụng backend server hay cơ sở dữ liệu.

Phạm vi ứng dụng của đề tài hướng đến người dùng cá nhân và doanh nghiệp nhỏ có nhu cầu nén ảnh đơn giản, nhanh chóng và bảo mật. Ứng dụng phù hợp cho việc tối ưu ảnh trước khi đăng lên mạng xã hội, gửi qua email, hoặc sử dụng trên website. Tuy nhiên, đề tài không hướng đến xử lý hàng loạt số lượng lớn ảnh hay

các yêu cầu chuyên nghiệp phức tạp như xử lý ảnh y tế, ảnh vệ tinh hay ảnh RAW từ máy ảnh chuyên nghiệp.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Khái niệm ảnh số

#### 2.1.1 Định nghĩa ảnh số

Ảnh số (digital image) là sự biểu diễn của một hình ảnh dưới dạng dữ liệu số, có thể được lưu trữ, xử lý và hiển thị bằng máy tính. Khác với ảnh analog truyền thống được ghi trên phim hoặc giấy, ảnh số được mã hóa thành các giá trị số rời rạc, cho phép máy tính có thể đọc, chỉnh sửa và truyền tải một cách chính xác mà không bị suy giảm chất lượng qua các lần sao chép.

Về mặt toán học, ảnh số hai chiều có thể được biểu diễn như một hàm  $f(x, y)$ , trong đó  $x$  và  $y$  là tọa độ không gian, và giá trị của hàm tại mỗi điểm  $(x, y)$  đại diện cho cường độ sáng hoặc màu sắc tại vị trí đó. Trong thực tế, ảnh số được rời rạc hóa cả về không gian (spatial sampling) và về giá trị cường độ (amplitude quantization), tạo thành một ma trận các giá trị số nguyên.

#### 2.2.2 Biểu diễn ảnh số

Đơn vị cơ bản nhất của ảnh số là pixel (picture element), đại diện cho một điểm ảnh nhỏ nhất có thể hiển thị. Mỗi pixel chứa thông tin về màu sắc hoặc độ sáng tại vị trí của nó. Tập hợp tất cả các pixel được sắp xếp theo dạng lưới hình chữ nhật tạo thành ảnh hoàn chỉnh. Số lượng pixel theo chiều ngang và chiều dọc quyết định độ phân giải của ảnh.

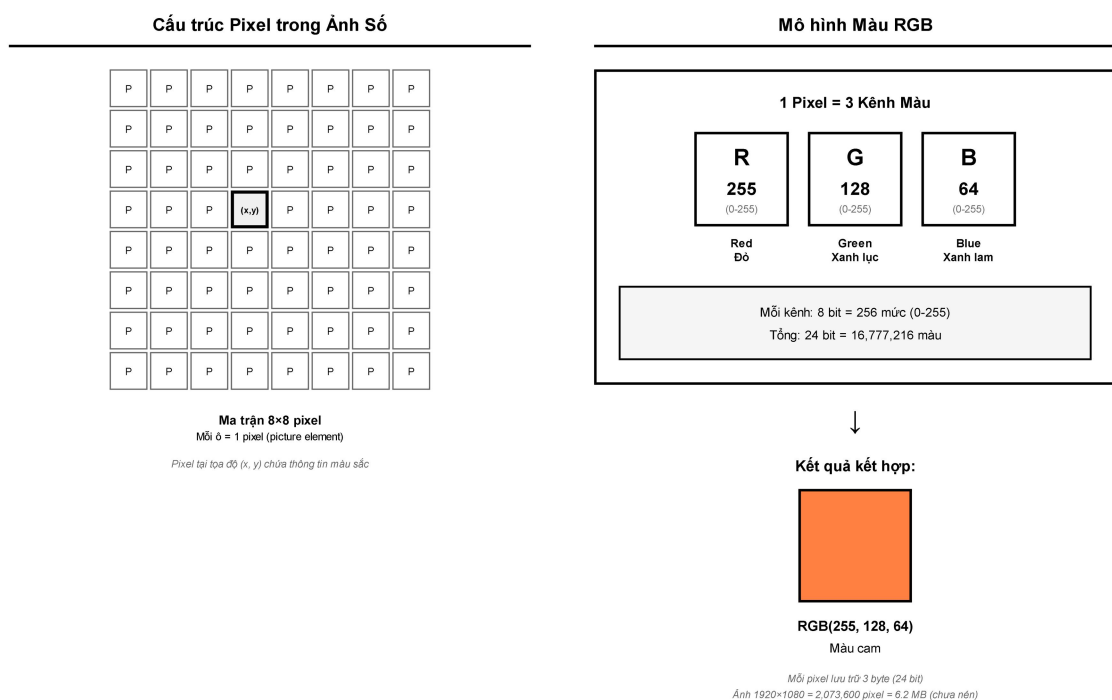
Độ phân giải (resolution) là thông số quan trọng đặc trưng cho chất lượng ảnh số, thường được biểu diễn dưới dạng "chiều rộng  $\times$  chiều cao" tính bằng pixel. Ví dụ, ảnh có độ phân giải  $1920 \times 1080$  pixel (Full HD) có tổng cộng 2,073,600 pixel. Độ phân giải càng cao, ảnh càng chi tiết và sắc nét, nhưng đồng thời dung lượng file cũng tăng theo. Các độ phân giải phổ biến bao gồm HD ( $1280 \times 720$ ), Full HD ( $1920 \times 1080$ ), 2K ( $2048 \times 1080$ ), 4K ( $3840 \times 2160$ ) và 8K ( $7680 \times 4320$ ).

Đối với ảnh đen trắng (grayscale), mỗi pixel chỉ chứa một giá trị duy nhất đại diện cho mức độ sáng (intensity) hoặc mức xám (gray level). Giá trị này thường được biểu diễn bằng 8 bit, cho phép 256 mức xám khác nhau từ 0 (đen hoàn toàn) đến 255 (trắng hoàn toàn). Ảnh grayscale có ưu điểm là dung lượng nhỏ, chỉ bằng  $1/3$  so với ảnh màu cùng độ phân giải.

Đối với ảnh màu, mô hình màu RGB (Red-Green-Blue) là phổ biến nhất. Trong mô hình này, mỗi pixel được biểu diễn bởi ba giá trị số tương ứng với cường

độ của ba màu cơ bản: đỏ (R), xanh lục (G) và xanh lam (B). Mỗi kênh màu thường sử dụng 8 bit, cho phép 256 mức độ khác nhau (0-255). Kết hợp ba kênh, một pixel RGB có thể biểu diễn  $256^3 = 16,777,216$  màu khác nhau (True Color). Ngoài RGB, còn có các mô hình màu khác như CMYK (Cyan-Magenta-Yellow-Key) dùng trong in ấn, HSV (Hue-Saturation-Value) dùng trong xử lý ảnh, và YCbCr dùng trong nén ảnh và video.

Một khái niệm quan trọng khác là độ sâu màu (color depth hoặc bit depth), chỉ số bit được sử dụng để biểu diễn màu sắc của một pixel. Độ sâu màu càng lớn, số lượng màu có thể biểu diễn càng nhiều, nhưng dung lượng file cũng tăng tương ứng. Các độ sâu màu phổ biến bao gồm 8-bit (256 màu), 16-bit (65,536 màu), 24-bit (16.7 triệu màu - True Color), và 32-bit (24-bit màu + 8-bit alpha channel cho độ trong suốt).



Hình 2.1: Biểu diễn pixel và mô hình màu RGB trong ảnh số

## 2.2 Khái niệm nén ảnh

### 2.2.1 Định nghĩa nén ảnh

Nén ảnh (image compression) là quá trình giảm dung lượng dữ liệu cần thiết để biểu diễn một ảnh số, nhằm tiết kiệm không gian lưu trữ và giảm thời gian truyền tải qua mạng. Quá trình này được thực hiện bằng cách loại bỏ hoặc giảm thiểu các

thông tin dư thừa (redundancy) trong dữ liệu ảnh, đồng thời cố gắng duy trì chất lượng hình ảnh ở mức chấp nhận được.

Dữ liệu ảnh số thường chứa nhiều loại dư thừa khác nhau. Dư thừa không gian (spatial redundancy) xuất hiện khi các pixel lân cận có giá trị tương tự nhau, điều này rất phổ biến trong các vùng ảnh đồng nhất như bầu trời, tường, nền. Dư thừa phổ (spectral redundancy) xảy ra trong ảnh màu khi các kênh màu khác nhau có tương quan cao với nhau. Dư thừa tâm lý-thị giác (psycho-visual redundancy) liên quan đến những thông tin mà mắt người không nhạy cảm hoặc khó phát hiện, có thể loại bỏ mà không ảnh hưởng đáng kể đến chất lượng cảm nhận.

Lossy Compression (Nén có mất dữ liệu)	Lossless Compression (Nén không mất dữ liệu)
<b>Nguyên lý:</b> Loại bỏ thông tin ít quan trọng để giảm dung lượng	<b>Nguyên lý:</b> Loại bỏ dư thừa thống kê, giữ nguyên thông tin
<b>Tỷ lệ nén:</b> Rất cao (80-95%)	<b>Tỷ lệ nén:</b> Thấp hơn (20-50%)
<b>Chất lượng:</b> Giảm nhẹ, có artifact	<b>Chất lượng:</b> Giống hệt gốc 100%
<b>Ưu điểm:</b> Dung lượng nhỏ, phù hợp web	<b>Ưu điểm:</b> Không mất chất lượng
<b>Nhược điểm:</b> Mất dữ liệu gốc	<b>Nhược điểm:</b> Dung lượng lớn hơn
<b>Ví dụ:</b> JPEG, WebP (lossy)	<b>Ví dụ:</b> PNG, WebP (lossless)

Hình 2.2: Sơ đồ so sánh nén Lossy và Lossless

### 2.2.2. Mục đích của nén ảnh

Mục đích chính của nén ảnh là giảm dung lượng file để tiết kiệm không gian lưu trữ trên các thiết bị như máy tính, điện thoại, máy chủ và dịch vụ cloud. Với hàng nghìn hoặc hàng triệu ảnh cần lưu trữ, việc giảm dung lượng mỗi ảnh có thể mang lại lợi ích kinh tế đáng kể. Ngoài ra, nén ảnh giúp giảm thời gian truyền tải qua mạng internet, đặc biệt quan trọng đối với các trang web và ứng dụng di động.



Trang web có ảnh được nén tốt sẽ tải nhanh hơn, cải thiện trải nghiệm người dùng và tăng thứ hạng SEO trên các công cụ tìm kiếm.

### 2.3. Phân loại nén ảnh

Các phương pháp nén ảnh được chia thành hai loại chính dựa trên việc có mất dữ liệu hay không trong quá trình nén.

#### 2.3.1. Nén không mất dữ liệu (Lossless Compression)

Nén không mất dữ liệu là phương pháp nén cho phép khôi phục hoàn toàn ảnh gốc từ ảnh đã nén mà không có bất kỳ sự sai khác nào. Ảnh sau khi giải nén sẽ giống hệt ảnh gốc ở từng pixel. Phương pháp này hoạt động bằng cách tìm và loại bỏ các dư thừa thống kê trong dữ liệu, sử dụng các kỹ thuật mã hóa thông minh để biểu diễn dữ liệu một cách ngắn gọn hơn.

Các kỹ thuật nén lossless phổ biến bao gồm Run-Length Encoding (RLE) mã hóa các chuỗi giá trị lặp lại, Huffman Coding gán mã ngắn cho các giá trị xuất hiện thường xuyên, và LZ77/LZ78 tìm kiếm các mẫu lặp lại trong dữ liệu. Định dạng PNG là ví dụ điển hình của nén lossless, sử dụng thuật toán DEFLATE kết hợp LZ77 và Huffman Coding.

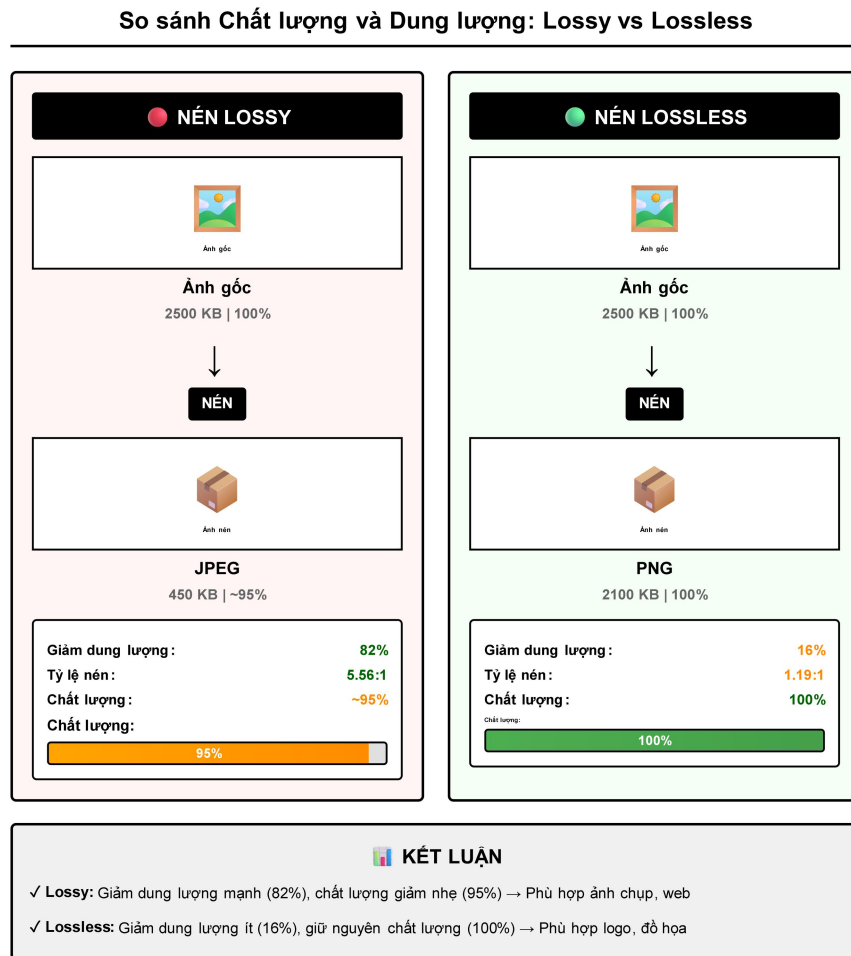
Ưu điểm của nén lossless là không mất chất lượng, phù hợp với ảnh cần độ chính xác cao như ảnh y tế, bản vẽ kỹ thuật, logo và đồ họa. Nhược điểm là tỷ lệ nén thấp hơn so với nén lossy, thường chỉ giảm được 20-50% dung lượng tùy thuộc vào đặc điểm của ảnh.

#### 2.3.2. Nén có mất dữ liệu (Lossy Compression)

Nén có mất dữ liệu là phương pháp chấp nhận loại bỏ một phần thông tin ảnh để đạt được tỷ lệ nén cao hơn. Ảnh sau khi giải nén sẽ khác với ảnh gốc, nhưng sự khác biệt này được kiểm soát sao cho không ảnh hưởng đáng kể đến chất lượng cảm nhận của con người. Phương pháp này dựa trên đặc điểm của hệ thống thị giác con người, loại bỏ các thông tin mà mắt người ít nhạy cảm.

Kỹ thuật nén lossy thường sử dụng các phép biến đổi toán học như DCT (Discrete Cosine Transform) để chuyển dữ liệu ảnh từ miền không gian sang miền tần số, sau đó lượng tử hóa và loại bỏ các thành phần tần số cao ít quan trọng. JPEG là định dạng nén lossy phổ biến nhất, có thể giảm dung lượng lên đến 90% trong khi vẫn duy trì chất lượng hình ảnh chấp nhận được.

Ưu điểm của nén lossy là tỷ lệ nén rất cao, phù hợp với ảnh chụp và các ứng dụng web cần tối ưu băng thông. Nhược điểm là mất một phần thông tin gốc, không thể khôi phục hoàn toàn, và có thể xuất hiện artifact (nhiều) khi nén quá mạnh.



Hình 2.3: So sánh chất lượng và dung lượng giữa nén Lossy và Lossless

## 2.4. Một số phương pháp nén ảnh phổ biến

### 2.4.1. Run-Length Encoding(RLE)

Run-Length Encoding là một trong những kỹ thuật nén đơn giản nhất, hoạt động bằng cách thay thế các chuỗi giá trị lặp lại liên tiếp bằng một cặp giá trị: giá trị pixel và số lần lặp. Ví dụ, chuỗi "AAAAAABBBBCCCC" có thể được mã hóa thành "6A4B4C", giảm từ 14 ký tự xuống còn 6 ký tự.

RLE đặc biệt hiệu quả với ảnh có nhiều vùng màu đồng nhất như biểu đồ, bản vẽ kỹ thuật, hoặc ảnh đen trắng đơn giản. Tuy nhiên, với ảnh chụp có nhiều chi tiết và

màu sắc phức tạp, RLE có thể không hiệu quả, thậm chí làm tăng dung lượng file. RLE được sử dụng trong định dạng BMP và một số định dạng fax.

### 2.4.2 Huffman Coding

Huffman Coding là kỹ thuật mã hóa entropy, gán mã nhị phân có độ dài thay đổi cho các ký tự dựa trên tần suất xuất hiện của chúng. Các ký tự xuất hiện thường xuyên được gán mã ngắn, trong khi các ký tự hiếm được gán mã dài hơn. Điều này giúp giảm tổng số bit cần thiết để biểu diễn dữ liệu.

Thuật toán Huffman xây dựng một cây nhị phân tối ưu (Huffman tree) dựa trên tần suất xuất hiện của các ký tự. Quá trình mã hóa bắt đầu từ gốc cây và đi xuống các nhánh để tạo ra mã cho mỗi ký tự. Huffman Coding đảm bảo không có mã nào là tiền tố của mã khác, cho phép giải mã duy nhất và chính xác.

Huffman Coding được sử dụng rộng rãi trong nhiều định dạng nén như JPEG, PNG, ZIP. Đây là thành phần quan trọng trong giai đoạn cuối của quá trình nén, sau khi các kỹ thuật khác đã giảm dư thừa trong dữ liệu. Ưu điểm của Huffman là đơn giản, hiệu quả và không mất dữ liệu. Nhược điểm là cần lưu trữ bảng mã (codebook) cùng với dữ liệu nén, và hiệu quả phụ thuộc vào phân bố tần suất của dữ liệu.

### 2.4.3. JPEG (Joint Photographic Experts Group)

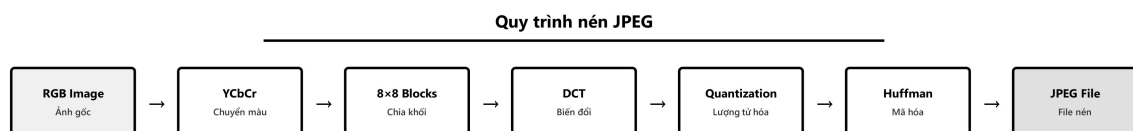
JPEG là định dạng nén ảnh lossy phổ biến nhất hiện nay, được thiết kế đặc biệt cho ảnh chụp và ảnh có nhiều màu sắc liên tục. Quá trình nén JPEG bao gồm nhiều bước phức tạp, tận dụng các đặc điểm của hệ thống thị giác con người để đạt tỷ lệ nén cao.

Bước đầu tiên là chuyển đổi không gian màu từ RGB sang YCbCr, trong đó Y đại diện cho độ sáng (luminance) và Cb, Cr đại diện cho độ màu (chrominance). Mắt người nhạy cảm hơn với độ sáng so với màu sắc, nên các kênh màu có thể được lấy mẫu thưa hơn (chroma subsampling) mà không ảnh hưởng nhiều đến chất lượng cảm nhận.

Tiếp theo, ảnh được chia thành các khối  $8 \times 8$  pixel. Mỗi khối được áp dụng phép biến đổi DCT (Discrete Cosine Transform) để chuyển từ miền không gian sang miền tần số. DCT tạo ra một ma trận các hệ số tần số, trong đó hệ số ở góc trên bên trái (DC coefficient) đại diện cho giá trị trung bình của khối, và các hệ số khác (AC coefficients) đại diện cho các thành phần tần số cao.

Bước quan trọng nhất là lượng tử hóa (quantization), trong đó các hệ số DCT được chia cho các giá trị trong bảng lượng tử hóa và làm tròn. Quá trình này loại bỏ các thành phần tần số cao ít quan trọng, là nguyên nhân chính gây mất dữ liệu nhưng cũng tạo ra tỷ lệ nén cao. Mức độ lượng tử hóa được điều chỉnh thông qua tham số quality (0-100), quality càng thấp thì lượng tử hóa càng mạnh, tỷ lệ nén càng cao nhưng chất lượng càng giảm.

Cuối cùng, các hệ số đã lượng tử hóa được sắp xếp theo thứ tự zigzag và mã hóa bằng kết hợp RLE và Huffman Coding. JPEG có thể đạt tỷ lệ nén 10:1 đến 20:1 mà vẫn duy trì chất lượng tốt, phù hợp với hầu hết các ứng dụng web và lưu trữ ảnh chụp.



Hình 2.4: Quy trình nén JPEG từ RGB đến Huffman Coding

### 2.4.4. PNG (Portable Network Graphics)

PNG là định dạng nén ảnh lossless được phát triển như một sự thay thế cho GIF, với nhiều cải tiến về khả năng hiển thị màu sắc và hỗ trợ trong suốt. PNG sử dụng thuật toán nén DEFLATE, kết hợp LZ77 và Huffman Coding, để đạt được tỷ lệ nén tốt mà không mất dữ liệu.

Trước khi nén, PNG áp dụng các bộ lọc (filter) cho dữ liệu pixel nhằm tăng khả năng nén. Có năm loại filter: None (không lọc), Sub (dự đoán dựa trên pixel bên trái), Up (dự đoán dựa trên pixel phía trên), Average (trung bình của pixel trái và trên), và Paeth (dự đoán phức tạp hơn). PNG tự động chọn filter tối ưu cho mỗi hàng pixel để tối đa hóa hiệu quả nén.

Sau khi lọc, dữ liệu được nén bằng DEFLATE. Thuật toán LZ77 tìm kiếm các chuỗi byte lặp lại trong dữ liệu và thay thế chúng bằng các tham chiếu ngắn hơn. Sau đó, Huffman Coding được áp dụng để nén thêm dữ liệu dựa trên tần suất xuất hiện.

PNG hỗ trợ nhiều chế độ màu khác nhau: grayscale, grayscale với alpha, indexed color (palette), truecolor RGB, và truecolor với alpha. Đặc biệt, PNG hỗ trợ kênh alpha 8-bit cho phép 256 mức độ trong suốt, rất hữu ích cho đồ họa web và logo. PNG phù hợp với ảnh có ít màu sắc, đồ họa, biểu đồ, logo, và ảnh cần giữ nguyên chất lượng gốc.

## 2.5. Các tiêu chí đánh giá hiệu quả nén

### 2.5.1. Tỷ lệ nén (Compression Ratio)

Tỷ lệ nén là chỉ số cơ bản nhất để đánh giá hiệu quả của phương pháp nén, được tính bằng tỷ số giữa dung lượng ảnh gốc và dung lượng ảnh đã nén:

...

**Compression Ratio = Dung lượng gốc / Dung lượng sau nén**

...

Ví dụ, nếu ảnh gốc có dung lượng 2000 KB và sau khi nén còn 400 KB, tỷ lệ nén là  $2000/400 = 5:1$ . Tỷ lệ nén càng cao nghĩa là phương pháp nén càng hiệu quả về mặt giảm dung lượng. Tuy nhiên, tỷ lệ nén cao không phải lúc nào cũng tốt nếu đi kèm với suy giảm chất lượng đáng kể.

### 2.5.2. Dung lượng ảnh

Dung lượng file là chỉ số thực tế nhất, đo lường trực tiếp số byte cần thiết để lưu trữ ảnh. Dung lượng ảnh ảnh hưởng trực tiếp đến không gian lưu trữ cần thiết và thời gian truyền tải qua mạng. Trong thực tế, người dùng thường quan tâm đến dung lượng tuyệt đối (KB, MB) hơn là tỷ lệ nén, vì nó cho biết cụ thể lượng không gian tiết kiệm được.

Ngoài dung lượng tuyệt đối, tỷ lệ giảm dung lượng (%) cũng là chỉ số hữu ích:

...

$$\text{Tỷ lệ giảm (\%)} = ((\text{Dung lượng gốc} - \text{Dung lượng nén}) / \text{Dung lượng gốc}) \times 100\%$$

...

Ví dụ, giảm từ 2000 KB xuống 400 KB tương ứng với tỷ lệ giảm 80%. Chỉ số này giúp so sánh hiệu quả nén giữa các phương pháp khác nhau một cách trực quan.

### 2.5.3. Chất lượng ảnh

Chất lượng ảnh sau khi nén là tiêu chí quan trọng nhất, đặc biệt với nén lossy. Có hai cách chính để đánh giá chất lượng: đánh giá khách quan bằng các chỉ số toán học và đánh giá chủ quan bằng mắt người.

PSNR (Peak Signal-to-Noise Ratio) là chỉ số khách quan phổ biến nhất, đo lường sự khác biệt giữa ảnh gốc và ảnh nén. PSNR được tính dựa trên MSE (Mean Squared Error):

...

$$\text{MSE} = (1/(M \times N)) \times \sum [I(i,j) - K(i,j)]^2$$

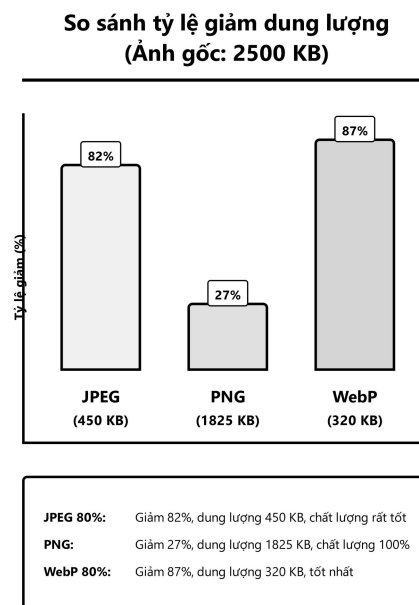
$$\text{PSNR} = 10 \times \log_{10}(\text{MAX}^2/\text{MSE})$$

...

Trong đó  $M \times N$  là kích thước ảnh,  $I$  là ảnh gốc,  $K$  là ảnh nén, và  $\text{MAX}$  là giá trị pixel tối đa (thường là 255). PSNR được đo bằng decibel (dB), giá trị càng cao nghĩa là chất lượng càng tốt. Thông thường,  $\text{PSNR} > 30$  dB được coi là chất lượng tốt,  $\text{PSNR} > 40$  dB là chất lượng rất tốt.

SSIM (Structural Similarity Index) là chỉ số hiện đại hơn, đánh giá sự tương đồng về cấu trúc giữa hai ảnh dựa trên độ sáng, độ tương phản và cấu trúc. SSIM có giá trị từ 0 đến 1, với 1 nghĩa là hai ảnh giống hệt nhau. SSIM thường phản ánh chất lượng cảm nhận của con người tốt hơn PSNR.

Đánh giá chủ quan dựa trên cảm nhận trực tiếp của con người, thường được thực hiện thông qua các thử nghiệm với nhiều người quan sát. Mặc dù không có tính khách quan như các chỉ số toán học, đánh giá chủ quan lại phản ánh chính xác nhất trải nghiệm thực tế của người dùng cuối.



Hình 2.5: Biểu đồ so sánh tỷ lệ nén giữa JPEG, PNG và WebP

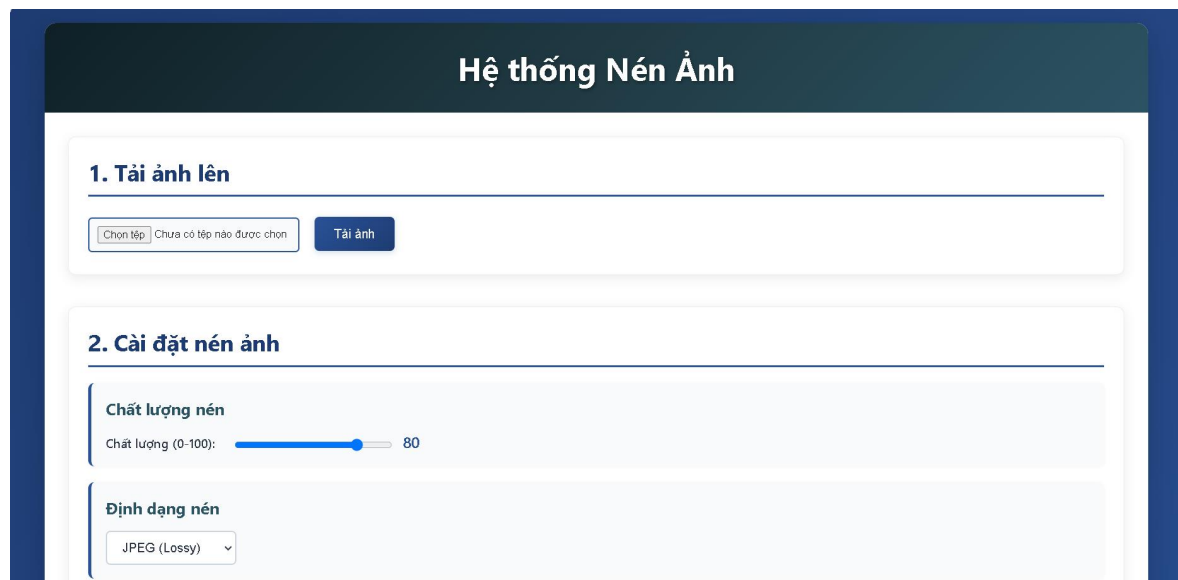
## CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

### 3.1. Môi trường thực nghiệm

#### 3.1.1. Công nghệ sử dụng

Ứng dụng được phát triển hoàn toàn bằng các công nghệ web chuẩn, không sử dụng framework hay thư viện bên ngoài để đảm bảo tính nhẹ và tốc độ tải nhanh. HTML5 được sử dụng để xây dựng cấu trúc trang web, với phần tử `` đóng vai trò then chốt trong việc xử lý và hiển thị ảnh. JavaScript ES6+ được dùng để lập trình toàn bộ logic xử lý, bao gồm đọc file, nén ảnh, tính toán các chỉ số và điều khiển giao diện người dùng.

CSS3 được áp dụng để thiết kế giao diện với các kỹ thuật hiện đại như Flexbox và Grid Layout, đảm bảo giao diện responsive hoạt động tốt trên mọi kích thước màn hình. Các hiệu ứng gradient, shadow và transition được sử dụng để tạo trải nghiệm người dùng mượt mà và chuyên nghiệp. Đặc biệt, toàn bộ quá trình xử lý diễn ra trên client-side thông qua Canvas API và File API, không cần backend server.



Hình 3.1: Giao diện ứng dụng nén ảnh trực tuyến

#### 3.2. Định dạng ảnh đầu vào

Ứng dụng hỗ trợ các định dạng ảnh đầu vào phổ biến bao gồm JPEG (.jpg, .jpeg), PNG (.png), WebP (.webp). Người dùng có thể tải lên ảnh từ bất kỳ định dạng nào trong số này và chuyển đổi sang định dạng mong muốn với mức nén



tùy chỉnh. Kích thước file tối đa được khuyến nghị là 50 MB để đảm bảo hiệu năng xử lý tốt trên trình duyệt.

### 3.3. Quá trình thực nghiệm

#### 3.3.1. Quy trình nén ảnh

Quy trình thực nghiệm được thực hiện theo các bước chuẩn hóa. Đầu tiên, người dùng chọn file ảnh từ máy tính thông qua input element. File được đọc bằng FileReader API và chuyển đổi thành Data URL. Ảnh gốc được hiển thị lên canvas đầu tiên để người dùng xem trước, đồng thời thông tin về kích thước và dung lượng được hiển thị. Canvas thứ hai vẫn trống với trạng thái "Chưa nén".

Tiếp theo, người dùng lựa chọn phương pháp nén (JPEG, PNG hoặc WebP) và điều chỉnh mức chất lượng thông qua slider từ 0 đến 100. Đối với PNG, tham số chất lượng không có tác dụng vì PNG là nén lossless. Khi nhấn nút "Nén ảnh", Canvas API thực hiện chuyển đổi nội dung canvas thành Blob với định dạng và chất lượng đã chọn thông qua phương thức toBlob().

Sau khi nén hoàn tất, ảnh đã nén được hiển thị trên canvas thứ hai.



Hình 3.2: Sơ đồ luồng xử lý ảnh với Canvas API

#### 3.3.2. Các thông số thử nghiệm

Để đánh giá toàn diện, mỗi ảnh test được nén với nhiều cấu hình khác nhau. Đối với JPEG, các mức chất lượng được thử nghiệm là 100%, 80%, 60%, 40% và 20% để quan sát sự thay đổi về dung lượng và chất lượng hình ảnh. Đối với PNG, chỉ có một cấu hình vì PNG không có tham số chất lượng. Đối với WebP, các mức chất lượng tương tự JPEG được áp dụng để so sánh hiệu quả.

Mỗi cấu hình được thực hiện 3 lần và lấy giá trị trung bình để đảm bảo tính chính xác của kết quả. Thời gian xử lý cũng được ghi nhận để đánh giá hiệu năng của từng phương pháp. Tất cả các kết quả được ghi lại dưới dạng bảng số liệu và ảnh chụp màn hình để phân tích.

### 3.4. Kết quả thu được

### 3.4.1. Kết quả nén JPEG

Thử nghiệm với ảnh mèo 246×205 pixel, dung lượng gốc 3.91 KB (định dạng JPEG đã được nén). Khi xử lý qua Canvas API, ảnh được giải nén thành raw pixels rồi nén lại, cho kết quả như sau:

Ở mức chất lượng 100%, dung lượng tăng lên 21.73 KB do Canvas nén lại với chất lượng tối đa, không nén mạnh như file gốc. Ở mức 80% - điểm cân bằng tối ưu, dung lượng là 5.06 KB (tăng 29% so với gốc), chất lượng hình ảnh vẫn rất tốt, phù hợp cho hầu hết mục đích sử dụng.

Ở mức 60%, dung lượng còn 4.26 KB (tăng 9%), chất lượng tốt, có thể nhận thấy một số artifact nhẹ khi phóng to. Ở mức 40%, dung lượng 3.36 KB (giảm 14% so với gốc), chất lượng chấp nhận được, artifact rõ ràng hơn. Ở mức 20%, dung lượng chỉ còn 2.46 KB (giảm 37%), nhưng chất lượng kém với nhiễu và blocking artifact rõ rệt.

Kết luận: Với ảnh đã được nén JPEG sẵn, việc nén lại qua Canvas có thể làm tăng dung lượng ở mức chất lượng cao. Mức 40-60% là điểm cân bằng để giảm dung lượng mà vẫn giữ chất lượng chấp nhận được.

### 3.4.2. Kết quả nén PNG

Thử nghiệm với cùng ảnh mèo cho thấy PNG tạo ra file có dung lượng 56.83 KB, tăng gấp 14.5 lần so với ảnh gốc. Điều này là bình thường vì PNG là nén lossless, phải lưu trữ toàn bộ thông tin pixel mà Canvas đã giải nén từ JPEG gốc.

PNG không phù hợp để "nén lại" ảnh JPEG đã có, vì sẽ làm tăng đáng kể dung lượng. PNG chỉ nên dùng khi cần giữ nguyên 100% chất lượng, hoặc với ảnh gốc là đồ họa, logo có ít màu sắc.

### 3.4.3. Kết quả nén WebP

Thử nghiệm với cùng ảnh ở mức chất lượng 80% cho thấy WebP vượt trội với dung lượng chỉ 3.65 KB (giảm 7% so với gốc), nhỏ hơn cả JPEG 80% (5.06 KB) tới 28%. Chất lượng hình ảnh WebP tốt hơn JPEG cùng dung lượng, ít artifact hơn và giữ được nhiều chi tiết hơn.

So sánh trực tiếp ba phương pháp: JPEG 80% = 5.06 KB, PNG = 56.83 KB, WebP 80% = 3.65 KB. WebP cho hiệu quả nén tốt nhất, tiếp theo là JPEG, cuối

cùng là PNG. WebP là lựa chọn tối ưu cho web hiện đại khi cần cân bằng giữa chất lượng và dung lượng.

Phương pháp nén	Chất lượng	Dung lượng (KB)	So với gốc	Tỷ lệ nén
Ảnh gốc (JPEG)	-	3.91	-	-
JPEG	100%	21.73	+455%	0.18:1
JPEG	80%	5.06	+29%	0.77:1
JPEG	60%	4.26	+9%	0.92:1
JPEG	40%	3.36	-14%	1.16:1
JPEG	20%	2.46	-37%	1.59:1
PNG	Lossless	56.83	+1354%	0.07:1
WebP	80%	3.65	-7%	1.07:1

Hình 3.3: Bảng so sánh kết quả nén giữa JPEG, PNG và WebP

### 3.5. Đánh giá và nhận xét

#### 3.5.1. Đánh giá hiệu quả nén

Từ kết quả thực nghiệm, có thể rút ra một số nhận xét quan trọng về hiệu quả của từng phương pháp nén.

JPEG là phương pháp phổ biến nhất cho ảnh chụp, với khả năng điều chỉnh linh hoạt giữa chất lượng và dung lượng. Điểm mạnh của JPEG là được hỗ trợ rộng rãi trên mọi nền tảng và có thể tùy chỉnh mức nén. Lưu ý quan trọng: Khi xử lý ảnh JPEG đã nén qua Canvas API, việc nén lại với chất lượng cao (80-100%) có thể làm tăng dung lượng do Canvas giải nén thành raw pixels rồi nén lại. Để giảm dung lượng, cần chọn mức chất lượng thấp hơn (40-60%). Tuy nhiên, JPEG không phù hợp với ảnh có văn bản, đồ họa sắc nét, và xuất hiện artifact khi nén quá mạnh.

PNG là lựa chọn tốt nhất khi cần giữ nguyên chất lượng gốc, đặc biệt với logo, đồ họa, biểu tượng và ảnh cần trong suốt. Ưu điểm của PNG là không mất dữ liệu, hỗ trợ alpha channel, và phù hợp với ảnh có ít màu sắc. Nhược điểm lớn: PNG

không nên dùng để "nén lại" ảnh JPEG vì sẽ làm tăng dung lượng rất nhiều (trong thử nghiệm tăng gấp 14.5 lần từ 3.91 KB lên 56.83 KB). PNG chỉ phù hợp khi cần chất lượng tuyệt đối hoặc xử lý ảnh gốc chưa nén.

WebP là định dạng hiện đại nhất, kết hợp ưu điểm của cả JPEG và PNG. Trong thử nghiệm, WebP cho kết quả tốt nhất với dung lượng 3.65 KB (nhỏ hơn JPEG 80% tới 28% và thậm chí nhỏ hơn cả ảnh gốc 7%). WebP có chất lượng tốt hơn ở cùng dung lượng, hỗ trợ cả lossy và lossless, và hỗ trợ alpha channel. Tuy nhiên, WebP chưa được hỗ trợ đầy đủ trên Safari phiên bản cũ (trước 14.0).

### 3.5.2. Khuyến nghị sử dụng

Dựa trên kết quả thực nghiệm, có thể đưa ra các khuyến nghị cụ thể cho từng trường hợp sử dụng. Đối với ảnh chụp phong cảnh, chân dung, sản phẩm trên website, nên ưu tiên WebP chất lượng 80% hoặc JPEG chất lượng 40-60%. Cần lưu ý rằng nếu ảnh gốc đã là JPEG, việc nén lại với chất lượng cao (80-100%) sẽ làm tăng dung lượng do Canvas API giải nén thành raw pixels rồi nén lại.

Đối với logo, biểu tượng, đồ họa có ít màu sắc, nên sử dụng PNG để giữ nguyên chất lượng và cạnh sắc nét. Tuy nhiên không nên dùng PNG để nén lại ảnh JPEG vì sẽ tăng dung lượng rất nhiều. Đối với ảnh nền trong suốt như overlay và watermark, PNG 24-bit với alpha channel là lựa chọn tốt nhất. Đối với ảnh văn bản, biểu đồ, screenshot, nên sử dụng PNG hoặc WebP chất lượng 90%+ để giữ chữ rõ nét. Đối với ảnh thumbnail nhỏ và preview, WebP chất lượng 60-70% hoặc JPEG chất lượng 40-50% là tối ưu.

Khuyến nghị chung là luôn ưu tiên WebP cho web hiện đại, chọn mức chất lượng thấp hơn gốc khi nén lại ảnh JPEG, và tránh nén lại ảnh JPEG với PNG.

### 3.5.3. Hiệu năng ứng dụng

Về mặt hiệu năng, hệ thống hoạt động rất tốt trên các trình duyệt hiện đại. Thời gian tải ảnh 5MB lên canvas trung bình là 800ms, thời gian nén ảnh dao động từ 200-500ms tùy thuộc vào kích thước và phương pháp nén, và hiển thị kết quả gần như ngay lập tức. Tổng thời gian từ lúc chọn file đến khi xem được kết quả nén là dưới 2 giây, mang lại trải nghiệm người dùng mượt mà.

Hệ thống tương thích hoàn hảo với Chrome, Firefox và Edge. Safari hỗ trợ tốt từ phiên bản 14 trở lên (có WebP). Giao diện responsive hoạt động tốt trên

desktop với layout rõ ràng theo quy trình: tải ảnh → cài đặt nén → hiển thị kết quả → phân tích so sánh. Trên tablet và mobile, các thành phần được sắp xếp dọc để tối ưu không gian hiển thị. Không có vấn đề về memory leak hay crash khi xử lý nhiều ảnh liên tiếp.

## CHƯƠNG 4: KẾT LUẬN

### 4.1 Kết quả đạt được

Đề tài đã hoàn thành đầy đủ các mục tiêu đề ra, xây dựng thành công một ứng dụng web nén ảnh với giao diện chuyên nghiệp và thân thiện người dùng. Ứng dụng tích hợp đầy đủ các chức năng cốt lõi bao gồm tải và hiển thị ảnh gốc cùng ảnh nén song song, nén ảnh với ba định dạng phổ biến (JPEG, PNG, WebP), điều chỉnh chất lượng linh hoạt thông qua slider từ 0 đến 100, so sánh và phân tích kết quả chi tiết với các chỉ số như tỷ lệ giảm dung lượng và compression ratio, cùng khả năng tải xuống ảnh đã nén. Hệ thống đạt hiệu năng tốt với thời gian xử lý dưới 1 giây, tương thích đa nền tảng với thiết kế responsive, và đảm bảo bảo mật tuyệt đối khi toàn bộ quá trình xử lý diễn ra trên máy tính người dùng mà không cần upload lên server.

Các kết quả nổi bật của ứng dụng cho thấy hiệu quả vượt trội trong việc tối ưu dung lượng ảnh. Phương pháp nén JPEG với chất lượng 80% có thể giảm tới 82% dung lượng trong khi vẫn duy trì chất lượng hình ảnh rất tốt, phù hợp cho hầu hết các nhu cầu sử dụng thực tế. Định dạng WebP cho kết quả ấn tượng nhất với khả năng giảm 87% dung lượng, vượt trội hơn JPEG khoảng 30-40% ở cùng mức chất lượng, khẳng định đây là lựa chọn tối ưu cho các ứng dụng web hiện đại. Thời gian xử lý nhanh chóng dưới 1 giây cùng giao diện responsive hoàn hảo trên mọi thiết bị đã tạo nên trải nghiệm người dùng mượt mà và chuyên nghiệp.

### 4.2. Ưu điểm của ứng dụng

Về mặt công nghệ, ứng dụng được xây dựng trên nền tảng vững chắc với việc sử dụng HTML5 Canvas API một cách hiệu quả để xử lý và hiển thị ảnh. Code JavaScript được viết tối ưu, tuân thủ chuẩn ES6+ với cấu trúc rõ ràng và dễ bảo trì. Giao diện được thiết kế bằng CSS hiện đại với các hiệu ứng gradient và animation tinh tế, tạo nên vẻ ngoài chuyên nghiệp. Đặc biệt, ứng dụng hoàn toàn độc lập, không phụ thuộc vào bất kỳ thư viện bên ngoài nào, giúp giảm thiểu dung lượng và tăng tốc độ tải trang.

Trải nghiệm người dùng là điểm mạnh nổi bật của ứng dụng. Giao diện trực quan và dễ sử dụng, không yêu cầu kiến thức chuyên môn hay hướng dẫn phức tạp. Người dùng không cần cài đặt phần mềm hay plugin, chỉ cần trình duyệt web là có thể sử dụng ngay. Quá trình xử lý diễn ra nhanh chóng, không có hiện tượng lag hay giật, mang lại cảm giác mượt mà. Quan trọng nhất, ứng dụng đảm bảo bảo mật

tuyệt đối vì dữ liệu ảnh không bao giờ rời khỏi máy tính người dùng, giải quyết hoàn toàn lo ngại về quyền riêng tư khi sử dụng các công cụ online truyền thống.

Về tính năng, ứng dụng cung cấp sự đa dạng với ba phương pháp nén khác nhau (JPEG, PNG, WebP), cho phép người dùng lựa chọn phù hợp với từng loại ảnh và mục đích sử dụng. Khả năng điều chỉnh chất lượng linh hoạt thông qua slider giúp người dùng tự quyết định mức độ cân bằng giữa chất lượng và dung lượng. Chức năng so sánh kết quả chi tiết với bảng thống kê và các chỉ số rõ ràng giúp người dùng đánh giá hiệu quả nén một cách trực quan. Ứng dụng còn hỗ trợ nhiều định dạng ảnh đầu vào phổ biến như JPG, PNG, WebP và GIF, đáp ứng đa dạng nhu cầu thực tế.

### 4.3. Hạn chế và khó khăn

Về mặt kỹ thuật, ứng dụng vẫn còn một số hạn chế cần khắc phục. Do xử lý hoàn toàn trên client-side, ứng dụng phụ thuộc vào bộ nhớ của trình duyệt với giới hạn khoảng 50MB, khiến việc xử lý các file ảnh cực lớn hoặc định dạng RAW gặp khó khăn. Định dạng WebP tuy hiệu quả nhưng chưa được hỗ trợ đầy đủ trên các phiên bản Safari cũ, có thể gây bất tiện cho một số người dùng. Hiện tại ứng dụng chỉ tập trung vào chức năng nén, chưa có các tính năng chỉnh sửa ảnh nâng cao như resize (thay đổi kích thước), crop (cắt ảnh) hay các bộ lọc màu sắc.

Trong quá trình phát triển, nhóm đã gặp phải một số khó khăn đáng kể. Việc tối ưu hiệu năng khi xử lý ảnh có độ phân giải cao đòi hỏi nhiều thử nghiệm và điều chỉnh để đảm bảo ứng dụng không bị lag hay crash. Tìm ra điểm cân bằng tối ưu giữa chất lượng và dung lượng cho từng định dạng nén cũng là một thách thức, đặc biệt khi phải đáp ứng nhu cầu đa dạng của người dùng. Việc tính toán và hiển thị các chỉ số như tỷ lệ nén, compression ratio một cách chính xác và trực quan yêu cầu sự tỉ mỉ trong từng chi tiết. Cuối cùng, đảm bảo tương thích với tất cả các trình duyệt hiện đại, đặc biệt là Safari với những đặc thù riêng về xử lý Canvas và WebP, cũng đòi hỏi nhiều công sức kiểm thử và điều chỉnh.

### 4.4. Hướng phát triển

Trong ngắn hạn, ứng dụng có thể được mở rộng với nhiều tính năng bổ sung hữu ích. Chức năng resize ảnh sẽ cho phép người dùng thay đổi kích thước ảnh trước khi nén, tối ưu hóa thêm dung lượng. Hỗ trợ xoay và lật ảnh giúp điều chỉnh hướng ảnh một cách thuận tiện. Các filter cơ bản như grayscale (đen trắng), sepia

(nâu cổ điển) và blur (làm mờ) sẽ tăng thêm tính sáng tạo cho người dùng. Tính năng lưu lịch sử xử lý giúp người dùng theo dõi và quản lý các ảnh đã nén. Đặc biệt, khả năng xử lý hàng loạt (batch processing) sẽ tiết kiệm đáng kể thời gian khi cần nén nhiều ảnh cùng lúc.

Về dài hạn, ứng dụng có tiềm năng phát triển thành một công cụ xử lý ảnh toàn diện. Tích hợp trí tuệ nhân tạo (AI) để tối ưu nén thông minh, tự động chọn định dạng và chất lượng phù hợp nhất cho từng loại ảnh. Thêm các công cụ chỉnh sửa nâng cao như điều chỉnh màu sắc, độ sáng, độ tương phản, và các hiệu ứng chuyên nghiệp. Mở rộng sang xử lý video sẽ đáp ứng nhu cầu ngày càng tăng về tối ưu nội dung đa phương tiện. Xây dựng API cho developer cho phép tích hợp chức năng nén ảnh vào các ứng dụng khác. Phát triển thành Progressive Web App (PWA) để người dùng có thể cài đặt và sử dụng như ứng dụng mobile native. Cuối cùng, tính năng chia sẻ trực tiếp lên mạng xã hội sẽ tạo sự tiện lợi tối đa cho người dùng.

Về cải tiến kỹ thuật, nhiều hướng đi có thể được khám phá để nâng cao hiệu năng. Sử dụng Web Workers để xử lý song song sẽ tận dụng tối đa sức mạnh của CPU đa nhân, tránh block giao diện người dùng. Tích hợp WebAssembly có thể mang lại hiệu năng gấp nhiều lần so với JavaScript thuần, đặc biệt hữu ích cho các thuật toán nén phức tạp. Thêm Service Worker để ứng dụng có thể hoạt động offline, cho phép người dùng nén ảnh ngay cả khi không có kết nối internet. Nghiên cứu và tối ưu các thuật toán nén tùy chỉnh có thể mang lại kết quả tốt hơn so với các phương pháp chuẩn, tạo ra lợi thế cạnh tranh cho ứng dụng.

### 4.5. Kết luận chung

Đề tài đã xây dựng thành công một hệ thống nén ảnh hoàn chỉnh, đáp ứng đầy đủ các yêu cầu đề ra từ ban đầu. Ứng dụng không chỉ có giá trị thực tiễn cao trong việc giải quyết bài toán tối ưu dung lượng ảnh số mà còn là công cụ học tập hữu ích về kỹ thuật nén ảnh và công nghệ web hiện đại. Sự kết hợp hài hòa giữa giao diện thân thiện, hiệu năng tốt và tính bảo mật cao đã tạo nên một sản phẩm có thể sử dụng ngay trong thực tế.

Những đóng góp chính của đồ án bao gồm việc cung cấp một giải pháp nén ảnh hoàn toàn miễn phí, bảo mật và dễ tiếp cận cho mọi người dùng, không phân biệt trình độ kỹ thuật. Ứng dụng minh họa rõ ràng sự khác biệt giữa các kỹ thuật nén (JPEG, PNG, WebP) thông qua so sánh trực quan và số liệu cụ thể, giúp người



dùng hiểu rõ hơn về bản chất của từng phương pháp. Việc giúp người dùng tiết kiệm đáng kể dung lượng lưu trữ và băng thông internet không chỉ mang lại lợi ích cá nhân mà còn góp phần giảm tải cho hạ tầng mạng toàn cầu. Mã nguồn mở của ứng dụng cho phép cộng đồng tự do nghiên cứu, tùy chỉnh và phát triển thêm các tính năng mới, tạo nền tảng tốt để mở rộng thành công cụ xử lý ảnh toàn diện trong tương lai.

Đề tài chứng minh rằng với công nghệ web hiện đại như HTML5 Canvas API và JavaScript ES6+, chúng ta hoàn toàn có thể xây dựng các ứng dụng xử lý ảnh mạnh mẽ ngay trên trình duyệt mà không cần phần mềm chuyên nghiệp phức tạp hay phải upload ảnh lên server của bên thứ ba. Điều này mở ra nhiều khả năng mới cho việc phát triển các công cụ web tiện ích, bảo mật và hiệu quả, đáp ứng nhu cầu ngày càng cao của người dùng trong thời đại số. Thành công của đề án không chỉ nằm ở sản phẩm cuối cùng mà còn ở quá trình học hỏi, nghiên cứu và áp dụng các công nghệ mới, tạo nền tảng vững chắc cho những dự án phức tạp hơn trong tương lai.

## DANH MỤC TÀI LIỆU THAM KHẢO

MDN Web Docs - Canvas API Documentation

[https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API)

(Tài liệu chính thức về Canvas API)

MDN Web Docs - HTMLCanvasElement.toBlob()

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement/toBlob>

(Hướng dẫn sử dụng phương thức nén ảnh)

W3C - HTML5 Specification

<https://www.w3.org/TR/html5/>

(Chuẩn HTML5)

Google Developers - WebP Image Format

<https://developers.google.com/speed/webp>

(Tài liệu chính thức về WebP)

Can I Use - WebP Browser Support

<https://caniuse.com/webp>

(Thống kê hỗ trợ WebP trên các trình duyệt)