|  |
| --- |
| **BỘ XÂY DỰNG**  **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **Xử lý ảnh và thị giác máy tính**  ***Chương trình tạo ảnh động***  **Giảng viên hướng dẫn: TS. Trần Nguyên Bảo**  **Nhóm sinh viên thực hiện: Nhóm 04**  **Lớp: 010100086903**  **TP.Hồ Chí Minh, tháng 11/2025** |

|  |
| --- |
| **BỘ XÂY DỰNG**  **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **Xử lý ảnh và thị giác máy tính**  ***Chương trình tạo ảnh động***    **Giảng viên hướng dẫn: TS. Trần Nguyên Bảo**  **Nhóm sinh viên thực hiện: Nhóm 04**  **Lớp:**  **010100086903**  **TP.Hồ Chí Minh, tháng 11/2025** |

**Danh sách Nhóm:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Ghi chú** |
| 1 | **Lê Quang Nguyên** | **2331540284** | **23ĐHTT01** | Thành viên |
| 2 | **Ao Trường Giang** | **2331540143** | **23ĐHTT03** | Thành viên |
| 3 | **Trần Phạm Minh Đức** | **2331540141** | **23ĐHTT03** | Nhóm Trưởng |
| 4 | **Hoàng Quốc Cường** | **2331540219** | **23ĐHTT04** | Thành viên |
| 5 | **Nguyễn Hữu Tín** | **2331540057** | **23ĐHTT02** | Thành viên |

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1**  *(ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(ký và ghi rõ họ tên)* |
| **Cán bộ chấm thi phúc khảo 1**  *(ký và ghi rõ họ tên)* | **Cán bộ chấm thi phúc khảo 2**  *(ký và ghi rõ họ tên)* |

**DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

[Hình 3.1: Biều đồ lớp Error! Bookmark not defined.](#_Toc211772005)

[Hình 3.2: Biều đồ usecse Người dùng Error! Bookmark not defined.](#_Toc211772006)

[Hình 3.3: Các collection của database Error! Bookmark not defined.](#_Toc211772007)

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc211771982)

[1.1. Lý do chọn đề tài 1](#_Toc211771983)

[1.2. Mục tiêu đề tài 1](#_Toc211771984)

[1.3. Phạm vi đề tài 2](#_Toc211771985)

[1.4 Đối tượng nghiên cứu 2](#_Toc211771986)

[1.5. Phương pháp nghiên cứu 2](#_Toc211771987)

[1.6 Bố cục 3](#_Toc211771988)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 4](#_Toc211771989)

[2.1. Cơ sở lý thuyết A (ví dụ: 2.1. Framework Laravel) 4](#_Toc211771990)

[2.2. Cơ sở lý thuyết B 5](#_Toc211771991)

[2.3. Cơ sở lý thuyết C 5](#_Toc211771992)

[CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM 7](#_Toc211771993)

[3.1. Phân tích hệ thống 7](#_Toc211771994)

[3.1.1 Tác nhân 7](#_Toc211771995)

[3.1.2 Use case 7](#_Toc211771996)

[3.2. Thiết kế cơ sở dữ liệu **Error! Bookmark not defined.**](#_Toc211771997)

[3.3. Xây dựng giao diện các chức năng sản phẩm 8](#_Toc211771998)

[3.3.1 Chức năng chi tiết sản phẩm **Error! Bookmark not defined.**](#_Toc211771999)

[KẾT LUẬN 9](#_Toc211772000)

[DANH MỤC TÀI LIỆU THAM KHẢO 10](#_Toc211772001)

[PHỤ LỤC 11](#_Toc211772002)

[Link GitHub 11](#_Toc211772003)

[Link Kế hoạch làm việc nhóm **Error! Bookmark not defined.**](#_Toc211772004)

**MỞ ĐẦU**

Trong thời đại công nghệ số phát triển mạnh mẽ, hình ảnh và video trở thành phương tiện truyền tải thông tin quan trọng trong mọi lĩnh vực như quảng cáo, giáo dục, truyền thông và giải trí. Các ứng dụng về xử lý ảnh và thị giác máy tính ngày càng được áp dụng rộng rãi, từ việc nhận diện khuôn mặt, phân loại vật thể đến tạo hiệu ứng hình ảnh động (animation).

Đề tài **“Xây dựng chương trình tạo ảnh động”** được thực hiện với mục tiêu giúp sinh viên nắm vững các kỹ thuật xử lý ảnh cơ bản, đồng thời vận dụng lập trình Python để tạo ra một công cụ có khả năng chuyển đổi chuỗi ảnh tĩnh thành ảnh động (GIF hoặc video ngắn).

Chương trình cho phép người dùng nhập vào các ảnh riêng lẻ, điều chỉnh tốc độ hiển thị, áp dụng các hiệu ứng chuyển cảnh (transition) và xuất ra tệp ảnh động hoàn chỉnh. Qua đó, người học có thể hiểu rõ hơn về quy trình xử lý chuỗi ảnh, biểu diễn khung hình và thao tác trên không gian thời gian trong lĩnh vực thị giác máy tính.

Mục tiêu của đề tài là xây dựng một công cụ đơn giản, dễ sử dụng, hỗ trợ người dùng (đặc biệt là sinh viên ngành công nghệ thông tin và thiết kế đồ họa) có thể tạo ra ảnh động phục vụ học tập, minh họa hoặc trình bày sản phẩm một cách trực quan, sinh động.

# CHƯƠNG 1. GIỚI THIỆU

## 1.1. Lý do chọn đề tài

Trong lĩnh vực **xử lý ảnh và thị giác máy tính**, việc phân tích và biểu diễn chuỗi hình ảnh theo thời gian đóng vai trò quan trọng trong nhiều ứng dụng như tạo hoạt ảnh, mô phỏng chuyển động, phân tích video hoặc hiển thị kết quả trực quan. Tuy nhiên, đa số người học mới chỉ dừng lại ở mức thao tác trên ảnh tĩnh, chưa tiếp cận sâu đến kỹ thuật tổng hợp chuỗi ảnh để tạo chuyển động.

Bên cạnh đó, các công cụ tạo ảnh động hiện nay (như Photoshop, After Effects hoặc các website online) thường phức tạp, yêu cầu bản quyền hoặc không cho phép tùy chỉnh sâu các thông số kỹ thuật. Do đó, việc **tự xây dựng một chương trình tạo ảnh động bằng Python** không chỉ giúp hiểu rõ nguyên lý hoạt động mà còn rèn luyện kỹ năng lập trình xử lý ảnh, sử dụng thư viện khoa học và thiết kế giao diện người dùng.

Xuất phát từ những nhu cầu thực tế và mong muốn vận dụng kiến thức đã học, nhóm chúng em quyết định thực hiện đề tài **“Xây dựng chương trình tạo ảnh động”**.

## 1.2. Mục tiêu đề tài

Mục tiêu chính của đề tài là **xây dựng một chương trình có khả năng tạo ảnh động (animation)** từ các ảnh tĩnh, thông qua các thao tác xử lý ảnh cơ bản và thao tác trên chuỗi khung hình.

Cụ thể, đề tài hướng đến:

* Phát triển **ứng dụng chạy trên nền Python**, có giao diện hoặc sử dụng dòng lệnh dễ thao tác.
* Cho phép **người dùng tải và sắp xếp nhiều ảnh đầu vào** theo thứ tự mong muốn.
* Hỗ trợ **điều chỉnh tốc độ phát (frame per second - FPS)** và thời lượng từng ảnh.
* Thêm **hiệu ứng chuyển tiếp (fade, slide, zoom, dissolve)** giữa các ảnh.
* Xuất kết quả ra **định dạng GIF hoặc video (.mp4)**.
* (Tùy chọn nâng cao) Áp dụng **các thuật toán xử lý ảnh** như làm mờ, tăng tương phản hoặc nhận diện vật thể để tạo hiệu ứng tự động.

## 1.3. Phạm vi đề tài

* **Phạm vi không gian:** Ứng dụng chạy trên môi trường **máy tính cá nhân (desktop)**, có thể thực thi trên các hệ điều hành Windows, Linux hoặc macOS.
* **Phạm vi thời gian:** Thực hiện trong một học kỳ của môn học *Xử lý ảnh và Thị giác máy tính*.
* **Phạm vi kỹ thuật:**
  + Ngôn ngữ lập trình: **Python**
  + Thư viện chính: **OpenCV**, **Pillow (PIL)**, **imageio**, **tkinter**
  + Xuất đầu ra: **GIF** hoặc **video MP4**
  + Không tập trung vào xử lý video phức tạp, deep learning hay mô hình 3D.

## 1.4 Đối tượng nghiên cứu

* Kỹ thuật biểu diễn và xử lý chuỗi ảnh trong thị giác máy tính.
* Nguyên lý tạo ảnh động (animation) và cách lưu trữ khung hình.
* Cấu trúc dữ liệu hình ảnh và cách thao tác bằng thư viện OpenCV/Pillow.
* Các định dạng ảnh động (GIF, MP4) và phương pháp mã hóa.
* Giao diện và trải nghiệm người dùng (UI/UX) trong chương trình xử lý ảnh.

## 1.5. Phương pháp nghiên cứu

* **Phương pháp thu thập thông tin:**  
  Tìm hiểu tài liệu về xử lý ảnh động, tham khảo các thư viện Python chuyên dụng như OpenCV, Pillow, imageio, moviepy.
* **Phương pháp xử lý thông tin:**  
  Phân tích nguyên lý hoạt động của các công cụ tạo ảnh động; so sánh hiệu suất giữa các phương pháp đọc/ghi ảnh; lựa chọn cấu trúc chương trình phù hợp với mục tiêu đề tài.
* **Phương pháp thực nghiệm:**  
  Tiến hành lập trình thử nghiệm, kiểm thử các hàm xử lý chuỗi ảnh, tối ưu thời gian xuất ảnh động; khảo sát người dùng thử nghiệm (sinh viên) về mức độ dễ sử dụng và chất lượng kết quả.

## 1.6 Bố cục

Phần còn lại của báo cáo tiểu luận môn học này được tổ chức như sau:

**Chương 2**: *Cơ sở lý thuyết* — trình bày các khái niệm và công nghệ sử dụng, bao gồm xử lý ảnh bằng OpenCV/Pillow, nguyên lý hoạt động của ảnh động GIF, khái niệm khung hình (frame) và tốc độ phát (FPS), cùng tổng quan các định dạng video/ảnh động phổ biến.

**Chương 3**: *Phân tích và thiết kế hệ thống* — mô tả chức năng chương trình, sơ đồ use case, thiết kế giao diện, và luồng xử lý dữ liệu giữa các mô-đun.

**Chương 4**: *Cài đặt và kiểm thử hệ thống* — trình bày mã nguồn chính, kết quả thực nghiệm, các thông số kỹ thuật, hình ảnh minh họa đầu ra, và đánh giá hiệu quả.

**Chương 5**: *Kết luận và hướng phát triển* — tổng kết kết quả đạt được, hạn chế còn tồn tại, và đề xuất các hướng mở rộng như thêm hiệu ứng nâng cao, nhận dạng chuyển động tự động hoặc kết hợp AI để tạo animation thông minh.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Đề tài “Xây dựng chương trình tạo ảnh động” thuộc lĩnh vực xử lý ảnh số và thị giác máy tính, do đó cần ứng dụng nhiều kiến thức liên quan đến việc biểu diễn, thao tác và tổng hợp ảnh.  
Để hiện thực hóa hệ thống, nhóm lựa chọn sử dụng ngôn ngữ Python cùng các thư viện mã nguồn mở chuyên dụng cho xử lý ảnh và video như OpenCV, Pillow (PIL) và ImageIO.

Lý do lựa chọn các cơ sở lý thuyết này:

* Python là ngôn ngữ dễ học, có cộng đồng lớn và hỗ trợ mạnh mẽ trong lĩnh vực xử lý ảnh, AI, và thị giác máy tính.
* OpenCV cung cấp các công cụ xử lý ảnh nhanh, mạnh và hiệu quả.
* Pillow hỗ trợ thao tác ảnh tĩnh, đọc/ghi nhiều định dạng và vẽ lên ảnh dễ dàng.
* ImageIO hỗ trợ đọc/ghi ảnh động (GIF) và video, rất phù hợp cho phần xuất sản phẩm của đề tài.

## 2.1. Cơ sở lý thuyết Ngôn ngữ Python

Python được phát triển bởi **Guido van Rossum** vào cuối những năm 1980 và chính thức phát hành năm 1991. Python nổi tiếng nhờ cú pháp đơn giản, dễ đọc và thư viện phong phú, trở thành lựa chọn hàng đầu cho các lĩnh vực như **xử lý ảnh, thị giác máy tính, trí tuệ nhân tạo và khoa học dữ liệu**.

**Đặc điểm nổi bật:**

* Cú pháp dễ hiểu, phù hợp cho người mới bắt đầu.
* Cộng đồng người dùng và tài nguyên học tập phong phú.
* Có thư viện mạnh cho xử lý ảnh/video như OpenCV, Pillow, Numpy, ImageIO, Matplotlib,…
* Dễ tích hợp với giao diện đồ họa (Tkinter, PyQt) hoặc môi trường web (Flask, Django).

## 2.2. Cơ sở lý thuyết Thư viện OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở ra đời vào năm 1999 bởi Intel, sau đó được hỗ trợ bởi Willow Garage và Itseez (sau này thuộc Intel). Thư viện này cung cấp hàng trăm hàm phục vụ xử lý ảnh, nhận dạng vật thể, thị giác máy tính và học sâu.

**Các chức năng chính của OpenCV:**

* Đọc, ghi và hiển thị ảnh/video.
* Thay đổi kích thước, cắt, xoay, lật ảnh.
* Áp dụng các phép biến đổi hình học và màu sắc.
* Lọc ảnh (Gaussian, Median, Bilateral).
* Xử lý biên, phát hiện vật thể, trích chọn đặc trưng.

## 2.3. Cơ sở lý thuyết Thư viện Pillow (PIL)

**Pillow** là phiên bản nâng cấp của **Python Imaging Library (PIL)** – thư viện xử lý ảnh ra đời từ năm 1995. Pillow hỗ trợ đọc, ghi, chỉnh sửa và hiển thị nhiều định dạng ảnh khác nhau (JPEG, PNG, BMP, GIF,…).

**Các chức năng chính:**

* Đọc và ghi ảnh (Image.open(), Image.save()).
* Chuyển đổi định dạng ảnh.
* Thao tác pixel (lấy giá trị, thay đổi màu sắc).
* Vẽ văn bản hoặc hình dạng lên ảnh (ImageDraw).
* Hỗ trợ tạo và lưu ảnh GIF.

## 2.4. Cơ sở lý thuyết Thư viện ImageIO

**ImageIO** là thư viện Python chuyên dùng để đọc và ghi nhiều định dạng ảnh, video và ảnh động (GIF, MP4, AVI, TIFF, v.v.). Thư viện này rất gọn nhẹ, dễ dùng và thường được sử dụng trong các ứng dụng xử lý ảnh nhanh.

**Các chức năng chính :**

* Đọc và ghi ảnh, video, ảnh động.
* Hỗ trợ ghi ảnh động từ danh sách ảnh (imageio.mimsave()).
* Tương thích với NumPy để thao tác ma trận ảnh.
* Hỗ trợ xuất video nhiều định dạng thông qua FFMPEG.

## 2.5. Cơ sở lý thuyết Nguyên lý tạo ảnh động (Animation)

**Ảnh động** (animated image) là chuỗi các ảnh tĩnh (frame) được hiển thị liên tiếp với tốc độ nhất định, tạo cảm giác chuyển động liên tục. Mỗi khung hình có thể có thời gian hiển thị riêng (frame duration).

**Nguyên lý hoạt động**

* Chuỗi ảnh được sắp xếp theo thứ tự thời gian.
* Mỗi ảnh được hiển thị trong một khoảng thời gian ngắn (ví dụ: 0.1 giây).
* Khi lặp lại liên tục, mắt người cảm nhận như vật thể đang chuyển động.
* Tốc độ hiển thị (FPS – frame per second) quyết định độ mượt của ảnh động.

# CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM

(Tùy theo các dạng đề tài của MÔN HỌC mà phần chương 3 sẽ có cách trình bày khác nhau. Đây là mẫu dành cho các Đề tài dạng Xây dựng các hệ thống ứng dụng. Sinh viên cần tham khảo sự hướng dẫn của giảng viên hướng dẫn Đề tài. Tuy nhiên, ít nhất phải có 2 phần: Phân tích hệ thống và Xây dựng sản phẩm)

## 3.1. Phân tích hệ thống

### 3.1.1 Mục tiêu và yêu cầu của hệ thống

Hệ thống được xây dựng nhằm **xử lý ảnh và tạo ảnh động (GIF)** từ các chuỗi ảnh tĩnh, phục vụ cho các mục đích như trình diễn, minh họa hoặc thị giác máy tính.  
Ứng dụng tập trung vào việc cho phép người dùng **nhập nhiều ảnh đầu vào**, **xử lý chúng** (như thay đổi kích thước, áp dụng bộ lọc, chuyển đổi định dạng, v.v.), và cuối cùng **ghép lại thành một ảnh động (GIF)** có thể lưu hoặc xem trực tiếp.

### 3.1.2

### 3.1.3 Mô hình hoạt động của hệ thống

**Luồng hoạt động (Data Flow):**

**Người dùng Hệ thống**

|  |
| --- |
|  |
| **------------------------------------ Chọn các ảnh đầu vào ---------------------------->** |
|  |
| **<--------------------------------- Hiển thị ảnh -----------------------------------------** |
|  |
| **<--------------------------------------Áp dụng xử lý ảnh------------------------------** |
|  |
| **--- ------------------------------------Nhấn "Generate GIF" ---------------------->** |
|  |
| **<--------------------------------------- Tạo GIF -----------------------------------------** |
|  |

## 3.2. Xây dựng giao diện các chức năng sản phẩm

**(Phần Xây dựng giao diện cho các chức năng trong hệ thống tùy theo mỗi đề tài sẽ mà phân ra từng mục nhỏ khác nhau theo tác nhân hệ thống)**

**Lưu ý cách trình bày trong phần trình bày thiết kế giao diện:**

**Phải có tên chức năng của giao diện**

**Chức năng giúp người dùng thực hiện công việc gì, giải thích chức năng giao diện**

**Đặt tên hình cho chức năng, để làm danh mục hình, đặt đúng cứu pháp theo quy định.**

**….**

**3.2.1 ….**

**….**

**3.2.2 ….**

**….**

# KẾT LUẬN

## Kết luận

Đề tài **“Xây dựng chương trình tạo ảnh động”** đã hoàn thành mục tiêu đề ra là xây dựng được một ứng dụng xử lý ảnh cơ bản, cho phép người dùng:

Tải lên nhiều ảnh từ máy tính.

Áp dụng các thao tác xử lý ảnh cơ bản như chuyển ảnh sang thang xám, lọc nhiễu, thay đổi kích thước, v.v.

Tự động ghép chuỗi ảnh đã xử lý thành một ảnh động định dạng **GIF**.

Hiển thị kết quả trực quan ngay trên giao diện đồ họa thân thiện sử dụng **Tkinter**.

Chương trình được xây dựng hoàn toàn bằng **Python**, tận dụng sức mạnh của các thư viện **OpenCV**, **Pillow** và **ImageIO**, giúp xử lý và tạo ảnh động một cách nhanh chóng và chính xác.  
Qua quá trình triển khai, nhóm (hoặc sinh viên) đã:

Nắm vững quy trình đọc – xử lý – hiển thị ảnh bằng Python.

Hiểu rõ cách chuyển đổi giữa định dạng ảnh OpenCV (NumPy array) và ảnh hiển thị giao diện (PIL Image).

Thiết kế giao diện người dùng trực quan, dễ thao tác cho cả người không chuyên về kỹ thuật.

So với các công cụ tạo ảnh động phổ biến hiện nay, sản phẩm của đề tài tuy đơn giản nhưng có ưu điểm ở chỗ: mã nguồn mở, dễ mở rộng, và cho phép tích hợp thêm nhiều kỹ thuật xử lý ảnh khác nhau tùy nhu cầu học tập và nghiên cứu.

Trong quá trình thực hiện, đề tài đã đạt được hầu hết các mục tiêu ban đầu, tuy nhiên vẫn còn một số hạn chế như:

* + Chưa hỗ trợ nhiều hiệu ứng chuyển cảnh giữa các ảnh.
  + Chưa cho phép xem trước ảnh động trước khi lưu.
  + Giao diện còn đơn giản, chưa tối ưu cho nhiều độ phân giải màn hình.

Thông qua quá trình nghiên cứu và lập trình, sinh viên đã tích lũy được nhiều kinh nghiệm thực tiễn về xử lý ảnh, lập trình Python, tư duy thiết kế phần mềm, cũng như khả năng tự học, tìm hiểu tài liệu kỹ thuật.

## Hướng phát triển

Trong tương lai, đề tài có thể được mở rộng và hoàn thiện theo các hướng sau:

1. **Bổ sung hiệu ứng xử lý ảnh nâng cao:**
   * Thêm các bộ lọc làm mịn, làm nét, phát hiện biên, biến đổi màu, v.v.
   * Cho phép người dùng tự chọn loại hiệu ứng trước khi tạo ảnh động.
2. **Thêm các hiệu ứng chuyển động động học:**
   * Hỗ trợ hiệu ứng **fade-in / fade-out**, **zoom-in**, **slide**, hoặc **morphing** giữa các khung hình để tạo ảnh động mượt mà và chuyên nghiệp hơn.
3. **Xem trước ảnh động (Preview GIF):**
   * Trước khi lưu, người dùng có thể xem trước ảnh GIF trong cửa sổ giao diện, giúp dễ điều chỉnh tốc độ hoặc thứ tự ảnh.
4. **Tùy chỉnh tốc độ khung hình (Frame Duration):**
   * Thêm thanh điều chỉnh tốc độ phát hoặc lựa chọn khoảng thời gian giữa các ảnh.
5. **Nâng cấp giao diện người dùng (UI/UX):**
   * Thiết kế lại bằng thư viện **Tkinter nâng cao (ttk)** hoặc **PyQt5 / PySide6** để có giao diện hiện đại hơn, hỗ trợ nhiều kích cỡ màn hình.
6. **Tích hợp công nghệ trí tuệ nhân tạo (AI):**
   * Ứng dụng AI để tự động chọn ảnh phù hợp theo nội dung, nhận diện khuôn mặt hoặc tạo hoạt ảnh dựa trên chuyển động.
   * Dùng mô hình học sâu (Deep Learning) để sinh ảnh động từ video ngắn hoặc chuỗi ảnh chụp liên tiếp.
7. **Triển khai ứng dụng lên Web hoặc Mobile:**
   * Chuyển chương trình sang nền tảng web bằng **Flask / Django** hoặc tích hợp vào ứng dụng di động với **Kivy** hoặc **React Native + Python API**, giúp người dùng thao tác mọi lúc mọi nơi.

[1] “OpenCV – Open Source Computer Vision Library,” *OpenCV Documentation*, [Online]. Available: https://opencv.org/. [Accessed: 16-Oct-2025].

[2] “Pillow – The friendly PIL fork (Python Imaging Library),” *Pillow Documentation*, [Online]. Available: https://pillow.readthedocs.io/. [Accessed: 16-Oct-2025].

# PHỤ LỤC

## Link GitHub

[**Duckkeip/tieuluanxulyanh: abc**](https://github.com/Duckkeip/tieuluanxulyanh)