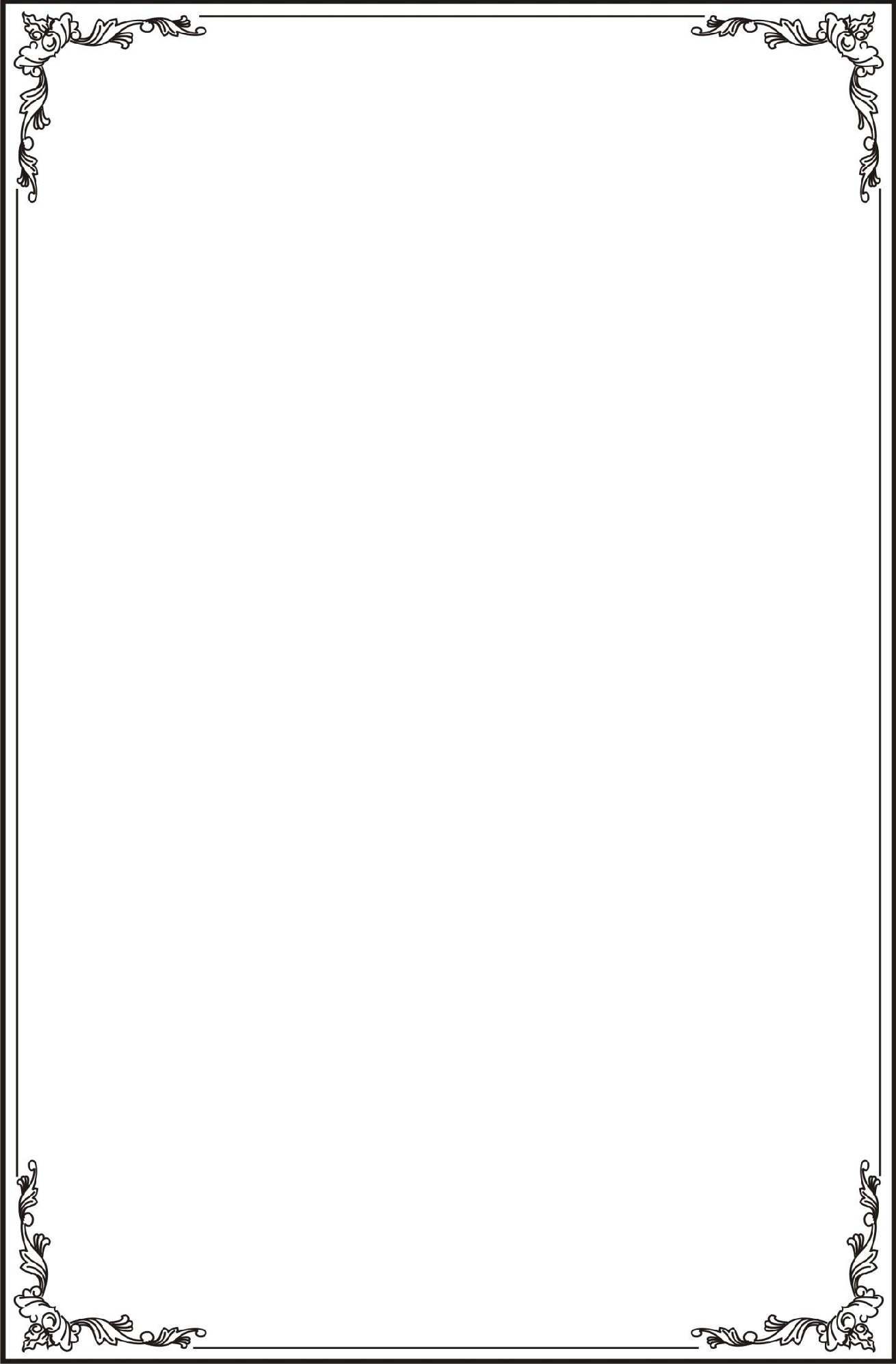
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC THĂNG LONG**

A black background with a black square

AI-generated content may be incorrect.

**BÀI TẬP LỚN  
THỊ GIÁC MÁY TÍNH**

**CHẤM ĐIỂM THI THPTQG**

GIẢNG VIÊN: NGÔ MẠNH CƯỜNG

SINH VIÊN THỰC HIỆN: A43605 HOÀNG ANH TUẤN

A42572 KHỔNG TRỌNG HOÀNG ANH

**HÀ NỘI – 2025**

**LỜI CẢM ƠN**

Sau một học kỳ của môn “Thị giác máy tính” với sự dẫn dắt của thầy Ngô Mạnh Cường. Bài tập lớn là bài tập cuối kỳ đánh dấu sự kết thúc quá trình học tập và rèn luyện các kiến thức cơ bản, đồng thời mở ra con đường thực tế áp dụng vào các môn học trong kỳ học tới.

Quá trình làm bài tập lớn giúp chúng em thu thập, tổng hợp lại các kiến thức đã học trong kì học qua, qua đó rèn luyện khả năng tổng hợp và giải quyết các vấn đề. Trong quá trình làm bài tập lớn chúng em gặp khó khăn trở ngại do vốn kiến thức còn giới hạn. Dù bận rộn nhưng thầy Cường vẫn giành thời gian tâm huyết hướng dẫn nhóm em.

Thầy Cường luôn quan tâm, chỉ bảo và giúp đỡ chúng em những vấn đề quan trọng giúp chúng em định hướng đúng đắn, giúp chúng em có tinh thần và khối lượng kiến thức phong phú để đến ngày hôm nay bài tập lớn của nhóm em đã được hoàn thành.

Với tất cả tấm lòng biết ơn sâu sắc, nhóm em xin chân thành gửi lời cảm ơn đến thầy Cường đã chân tình hướng dẫn giúp đỡ nhóm em trong suốt quá trình học tập trong môn học.

*Chúng em xin chân thành cảm ơn!*

*Hà Nội, ngày tháng năm 2025*

|  |  |  |
| --- | --- | --- |
| SINH VIÊN |  | SINH VIÊN |
|  |  |  |
| Hoàng Anh Tuấn |  | Khổng Trọng Hoàng Anh |

|  |
| --- |
|  |
|  |
|  |

**BẢNG PHÂN CÔNG VIỆC**

|  |  |
| --- | --- |
| Thành viên | Công việc |
| A43605  Hoàng Anh Tuấn | * Viết báo cáo * Xử lý ảnh * Mô hình hóa bài toán |
| A42572  Khổng Trọng Hoàng Anh | * Thiết kế giao diện * Đóng gói ứng dụng * Tạo slide thuyết trình |

# MỤC LỤC

[MỤC LỤC iv](#_Toc194458608)

[DANH MỤC HÌNH ẢNH vi](#_Toc194458609)

[DANH MỤC TỪ VIẾT TẮT xii](#_Toc194458610)

[DANH MỤC THUẬT NGỮ xiii](#_Toc194458611)

[LỜI MỞ ĐẦU 1](#_Toc194458612)

[A. Động lực 1](#_Toc194458613)

[B. Bố cục 1](#_Toc194458614)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc194458615)

[1.1. Các khái niệm cơ bản 2](#_Toc194458616)

[1.2. Các công cụ và thư viện 2](#_Toc194458617)

[1.2.1 Ngôn ngữ lập trình Python [1] 2](#_Toc194458618)

[1.2.2 Numpy [2] 4](#_Toc194458619)

[1.2.3 OpenCV [3] 5](#_Toc194458620)

[1.2.4 Tkinter [4] 7](#_Toc194458621)

[1.2.5 PIL [5] 8](#_Toc194458622)

[1.2.6 CSV [6] 10](#_Toc194458623)

[CHƯƠNG 2. MÔ TẢ DỮ LIỆU 11](#_Toc194458624)

[2.1. Dữ liệu ban đầu 11](#_Toc194458625)

[2.2 Xử lý ảnh 11](#_Toc194458626)

[2.3 Cắt hình ảnh 13](#_Toc194458627)

[CHƯƠNG 3. MÔ HÌNH HÓA BÀI TOÁN 16](#_Toc194458628)

[3.1. Mô tả bài toán 16](#_Toc194458629)

[3.2. Xây dựng mô hình 16](#_Toc194458630)

[3.2.1 Cắt vùng phiếu 16](#_Toc194458631)

[3.2.2 Chấm điểm 17](#_Toc194458632)

[CHƯƠNG 4 . CÀI ĐẶT, THỬ NGHIỆM 19](#_Toc194458633)

[4.1. Môi trường cài đặt 19](#_Toc194458634)

[4.2. Thử nghiệm 19](#_Toc194458635)

[4.2.1. Chức năng Chọn ảnh 19](#_Toc194458636)

[4.2.2 Chức năng Tải Đáp Án 19](#_Toc194458637)

[4.2.3 Chức năng Chấm Điểm 20](#_Toc194458638)

[KẾT LUẬN 22](#_Toc194458639)

[TÀI LIỆU THAM KHẢO 24](#_Toc194458640)

# DANH MỤC HÌNH ẢNH

[Hình 1.1. Hình ảnh minh họa Python 3](#_Toc194505621)

[Hình 1.2. Hình ảnh minh họa Numpy 4](#_Toc194505622)

[Hình 1.3. Hình ảnh minh họa OpenCV 6](#_Toc194505623)

[Hình 1.4. Hình ảnh minh họa Python 7](#_Toc194505624)

[Hình 1.5. Hình ảnh minh họa PIL 9](#_Toc194505625)

[Hình 2.1. Hình ảnh minh họa hàm extract\_part\_area 11](#_Toc194505626)

[Hình 2.2. Hình ảnh minh họa hàm get\_contours 12](#_Toc194505627)

[Hình 2.3. Hình ảnh minh họa hàm wrap\_image 14](#_Toc194505628)

[Hình 2.4. Hình ảnh minh họa hàm get\_4\_contour 14](#_Toc194505629)

[Hình 3.1. Hình ảnh minh họa lớp Detect 16](#_Toc194505630)

[Hình 3.2. Hình ảnh minh họa lớp Grading 17](#_Toc194505631)

[Hình 4.1. Hình ảnh minh họa chức năng Chọn Ảnh 19](#_Toc194505632)

[Hình 4.2. Hình ảnh minh họa chức năng Lưu Đáp Án 20](#_Toc194505633)

[Hình 4.3. Hình ảnh minh họa chức năng Chấm Điểm 21](#_Toc194505634)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Ý nghĩa |
| 1 | THPTQG | Trung học phổ thông quốc gia |
| 2 | PIL | Python Imaging Library |
| 3 | GUI | Graphical User Interface |
| 4 | TOEIC | Test of English for International Communication |
| 5 | CSV | Comma-Separated Values |
| 6 | RBG | Red, Green, Blue |

# DANH MỤC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ | Ý nghĩa |
| 1 | padding | Lề của ảnh |
| 2 | warp | Phép biến đổi |
| 3 | contours | Đường cong liên kết toàn bộ các điểm liên tục (dọc biên) cùng màu sắc hoặc giá trị cường độ |
| 4 | crop | Cắt ảnh |
| 5 | mean | Giá trị trung bình |
| 6 | Image Smoothing | Làm mịn hình ảnh |
| 7 | low-pass | Hạ bậc thấp |
| 8 | module | Mã nguồn |

# LỜI MỞ ĐẦU

## Động lực và mục tiêu

Bài thi THPTQG là một sự kiện quan trọng trong hệ thống giáo dục của một quốc gia. Hàng năm, hàng triệu học sinh tham gia vào bài thi này để đánh giá kiến thức và khả năng của họ.

Để đảm bảo tính công bằng và hiệu quả trong việc chấm điểm các bài thi, việc sử dụng thị giác máy tính để tự động hóa quá trình chấm điểm đã trở thành một ưu tiên.

Mục tiêu chính của đề tài này là phát triển một hệ thống chấm điểm tự động cho bài thi THPTQG bằng sử dụng thị giác máy tính. Hệ thống này sẽ có khả năng chấm điểm các bài thi viết và bài làm trên giấy dựa trên hình ảnh.

## Bố cục

Bài tập lớn môn Thị giác máy tính gồm có 5 chương:

* Chương mở đầu: Giới thiệu khái quát động lực và mục tiêu. Bên cạnh đó nêu rõ bố cục của bài tập lớn.
* Chương 1: Cơ sở lý thuyết: trình bày các khái niệm cơ bản của thị giác máy tính, cùng với các công cụ và thư viện được sử dụng để làm bài tập.
* Chương 2: Xử lý ảnh đầu vào: Cách xử lý ảnh đầu vào trước khi chấm điểm
* Chương 3:Mô hình hóa bài toán: Cách thuật toán nhận ảnh đã xử lý vào chấm thi và có giao diện ứng dụng.
* Chương 4: Cài đặt và thử nghiệm: Cách cài đặt và sử dụng ứng dụng chấm điểm thi.
* Kết luận: Đưa ra kết quả sau khi thực hiện bài tập, nêu lên các điểm mạnh kèm hạn chế và các mục tiêu phát triển mở rộng hơn cho hệ thống.

# CƠ SỞ LÝ THUYẾT

## 1.1. Các khái niệm cơ bản

* Ảnh gồm hai loại chính:
  + Ảnh màu: Mỗi điểm ảnh trong ảnh màu có thông tin về màu sắc, thường được biểu diễn bằng ba kênh màu RGB. Mỗi kênh chứa thông tin về màu tại mỗi điểm ảnh.
  + Ảnh xám: Ảnh xám chỉ chứa thông tin về mức xám (độ sáng) tại mỗi điểm ảnh. Nó không chứa thông tin về màu sắc và thường được biểu diễn bằng giá trị từ 0 (đen) đến 255 (trắng), hoặc trong một dải giá trị xám.
* Chuyển đổi Grayscale: giúp giảm số lượng thông tin mà bạn phải xử lý và tập trung vào các đặc trưng quan trọng của ảnh. Thông thường, việc chuyển đổi này được thực hiện bằng cách tính trung bình của các kênh màu (đỏ, xanh lá, và lam) tại mỗi điểm ảnh để tạo ra giá trị màu xám tương ứng.
* Image Smoothing: là kỹ thuật làm mịn hình ảnh bằng cách sử dụng một low-pass bộ lọc kernel. Điều này hữu ích để loại bỏ nhiễu. Vì vậy, các đường biên sẽ bị làm mờ một chút trong quá trình này.
* Xóa viền hình ảnh: được sử dụng để loại bỏ phần viền không mong muốn hoặc không cần thiết từ một hình ảnh. Điều này thường được thực hiện để làm sạch và tập trung vào phần chính của hình ảnh.
* Cắt ảnh: là quá trình loại bỏ một phần của hình ảnh gốc để chỉ giữ lại một phần cụ thể hoặc tạo ra một phần riêng biệt của hình ảnh. Quá trình này thường được thực hiện bằng cách xác định vùng cắt và sau đó trích xuất phần ảnh nằm trong vùng đó. Cắt ảnh là một phần quan trọng trong xử lý hình ảnh để tập trung vào các đối tượng hoặc phần cụ thể của hình ảnh và loại bỏ thông tin không cần thiết.

## 1.2. Các công cụ và thư viện

### 1.2.1 Ngôn ngữ lập trình Python [1]

#### 1.2.1.1. Định nghĩa

Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hoá các tác vụ và tiến hành phân tích dữ liệu. Python được Guido van Rossum tạo ra và lần đầu tiên phát hành vào năm 1991. Nó hỗ trợ nhiều mô hình lập trình như lập trình hướng đối tượng, lập trình thủ tục và lập trình hàm.

Python nổi bật với tính linh hoạt, thư viện phong phú, và cộng đồng phát triển lớn mạnh, cho phép nó được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển web, phân tích dữ liệu, trí tuệ nhân tạo, học máy, tự động hóa, và nhiều ứng dụng khác. Python là ngôn ngữ thông dịch, nghĩa là mã nguồn được thực thi trực tiếp từng dòng mà không cần biên dịch trước, giúp quá trình phát triển nhanh chóng và thuận tiện.



Hình 1.1. Hình ảnh minh họa Python

#### 1.2.1.2. Đặc trưng

Các đặc trưng nổi bật của ngôn ngữ lập trình Python:

* Cú pháp đơn giản và dễ đọc: Python có cú pháp rõ ràng, gần gũi với ngôn ngữ tự nhiên, giúp người mới bắt đầu dễ học và hiểu.
* Tính đa mô hình: Hỗ trợ lập trình hướng đối tượng, lập trình thủ tục, và lập trình hàm, mang lại sự linh hoạt trong cách viết mã.
* Đa nền tảng: Python hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux mà không cần thay đổi mã nguồn.
* Thư viện phong phú: Python đi kèm với một bộ thư viện chuẩn lớn, hỗ trợ nhiều tác vụ như xử lý tệp, kết nối mạng, toán học,....
* Mở rộng và nhúng dễ dàng: Có thể tích hợp với các ngôn ngữ khác như C/C++ và nhúng vào các ứng dụng để tăng hiệu suất.
* Quản lý bộ nhớ tự động: Python sử dụng cơ chế "garbage collection" để tự động thu hồi bộ nhớ không còn sử dụng, giảm gánh nặng cho lập trình viên.

#### 1.2.1.3. Ứng dụng

Python là ngôn ngữ lập trình đa nền tảng và dễ dàng mở rộng, dưới đây là các ứng dụng nổi bật của ngôn ngữ lập trình Python:

* Ứng dụng web: Python được sử dụng với các framework như Django, Flask để xây dựng các trang web động, hiệu quả và bảo mật.
* Trí tuệ nhân tạo (AI) và học máy (Machine Learning): Python là lựa chọn hàng đầu nhờ các thư viện như TensorFlow, PyTorch, Scikit-learn để phát triển mô hình AI, học sâu và phân tích dữ liệu.
* Phân tích dữ liệu (Data Analysis): Các thư viện như Pandas, NumPy và Matplotlib giúp xử lý, phân tích và trực quan hóa dữ liệu lớn một cách dễ dàng..
* Tự động hóa tác vụ: Python có thể viết script để tự động hóa các công việc lặp đi lặp lại như quản lý tệp, gửi email, hoặc crawl dữ liệu web (với thư viện như Selenium, BeautifulSoup).
* Xử lý hình ảnh và đồ họa: Các thư viện như OpenCV, Pillow hỗ trợ chỉnh sửa, phân tích hình ảnh và xử lý đồ họa..
* Kiểm thử phần mềm (Software Testing): Python hỗ trợ viết script kiểm thử tự động với các công cụ như PyTest, Unittest.

### 1.2.2 Numpy [2]

#### 1.2.2.1 Định nghĩa

NumPy là một thư viện mạnh mẽ trong Python được sử dụng để xử lý các mảng dữ liệu lớn và thực hiện các phép toán số học hiệu quả. Thư viện này cung cấp một đối tượng mảng đa chiều mạnh mẽ cùng với các công cụ để thao tác trên các mảng một cách hiệu quả.

NumPy được phát triển bởi Travis Oliphant và phát hành lần đầu vào năm 2005. Nó cung cấp một bộ hàm toán học đa dạng, hỗ trợ tính toán đại số tuyến tính, biến đổi Fourier, thống kê và nhiều lĩnh vực khác. NumPy là nền tảng cho nhiều thư viện khoa học dữ liệu khác như Pandas, SciPy, và TensorFlow.



Hình 1.2. Hình ảnh minh họa Numpy

#### 1.2.2.2. Đặc trưng

Các đặc trưng nổi bật của thư viện NumPy:

* Mảng đa chiều (ndarray): NumPy cung cấp cấu trúc dữ liệu mảng (array) có hiệu suất cao hơn so với danh sách (list) của Python.
* Hiệu suất cao: NumPy được viết bằng C, giúp tối ưu hóa tốc độ xử lý dữ liệu so với các vòng lặp Python thông thường.
* Tích hợp với các thư viện khác: NumPy là nền tảng của nhiều thư viện khoa học như Pandas, Matplotlib, SciPy, TensorFlow.
* Hỗ trợ đại số tuyến tính: Cung cấp các phép toán ma trận, phép nhân ma trận, phân rã giá trị suy biến (SVD), v.v.
* Hỗ trợ xử lý số ngẫu nhiên: NumPy cung cấp thư viện numpy.random để tạo số ngẫu nhiên phục vụ cho mô phỏng và học máy.

#### 1.2.2.3. Ứng dụng

NumPy là một thư viện cốt lõi trong lĩnh vực khoa học dữ liệu và tính toán số, được sử dụng rộng rãi trong nhiều lĩnh vực như:

* Phân tích dữ liệu: NumPy giúp lưu trữ và xử lý dữ liệu dạng mảng lớn, hỗ trợ các phép toán thống kê và tính toán số học nhanh chóng.
* Xử lý hình ảnh: NumPy hỗ trợ thao tác trên ảnh dưới dạng mảng số, giúp xử lý và phân tích ảnh dễ dàng hơn.
* Xây dựng đồ thị và trực quan hóa: NumPy kết hợp với Matplotlib để vẽ đồ thị, biểu đồ, hỗ trợ trực quan hóa dữ liệu hiệu quả.

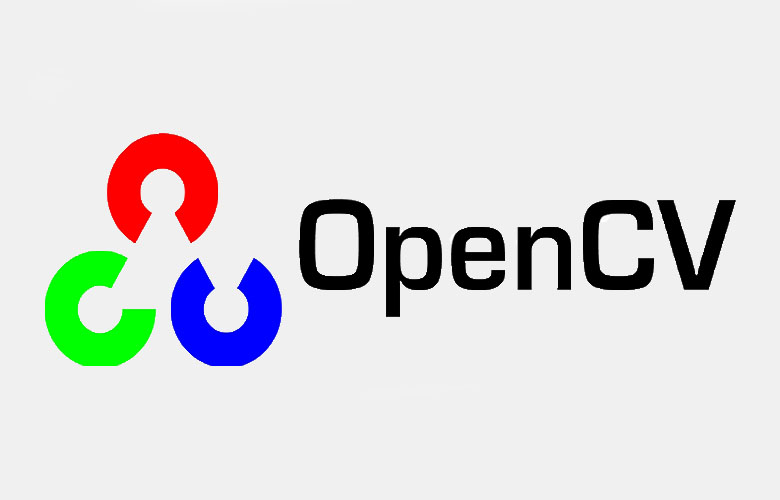
Thư viện NumPy là thư viện mạnh mẽ trong hệ sinh thái Python dành cho khoa học dữ liệu, giúp tối ưu hóa hiệu suất xử lý và tăng cường khả năng tính toán trong các dự án thực tế.

### 1.2.3 OpenCV [3]

#### 1.2.3.1 Định nghĩa

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở mạnh mẽ dành cho xử lý ảnh và thị giác máy tính. Thư viện này cung cấp nhiều công cụ giúp xử lý ảnh, nhận dạng mẫu, thị giác máy tính, và học sâu. OpenCV được viết chủ yếu bằng C++ nhưng có hỗ trợ các ngôn ngữ khác như Python, Java, và MATLAB.

OpenCV được phát triển bởi Intel vào năm 1999 và sau đó trở thành một dự án mã nguồn mở. Nó hỗ trợ nhiều nền tảng khác nhau như Windows, Linux, macOS và cả thiết bị di động. OpenCV cung cấp hàng trăm thuật toán tối ưu giúp xử lý ảnh và video một cách hiệu quả.



Hình 1.3. Hình ảnh minh họa OpenCV

#### 1.2.3.2 Đặc trưng

Các đặc trưng nổi bật của thư viện OpenCV:

* Xử lý ảnh và video mạnh mẽ: Hỗ trợ nhiều kỹ thuật xử lý ảnh như lọc ảnh, phát hiện biên, nhận diện khuôn mặt, theo dõi đối tượng, v.v.
* Hiệu suất cao: Được tối ưu hóa bằng ngôn ngữ C++, hỗ trợ tăng tốc phần cứng với OpenCL và GPU.
* Tích hợp với nhiều thư viện khác: Có thể kết hợp với NumPy, TensorFlow, Keras để xây dựng các hệ thống thị giác máy tính và học sâu.
* Hỗ trợ thị giác máy tính nâng cao: Bao gồm phát hiện vật thể, theo dõi chuyển động, phân đoạn ảnh, và nhận diện khuôn mặt.
* Hỗ trợ đa nền tảng: Chạy trên Windows, Linux, macOS và thiết bị di động như Android và iOS.

#### 1.2.3.3 Ứng dụng

OpenCV là một thư viện phổ biến trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo, được ứng dụng rộng rãi trong nhiều lĩnh vực như:

* Xử lý ảnh số: Lọc ảnh, làm mịn, phát hiện cạnh, tăng cường ảnh.
* Nhận diện khuôn mặt: Sử dụng trong bảo mật, giám sát và các ứng dụng tương tác người-máy.
* Thị giác máy tính trong robot: Giúp robot nhận biết và theo dõi đối tượng trong môi trường thực tế.
* Phân tích video: Ứng dụng trong camera an ninh, theo dõi chuyển động và phân tích hành vi.
* Ứng dụng y tế: Hỗ trợ phân tích ảnh y khoa, nhận diện bệnh lý từ hình ảnh.

Thư viện OpenCV đóng vai trò quan trọng trong các ứng dụng thị giác máy tính, giúp tối ưu hóa xử lý ảnh và hỗ trợ nhiều thuật toán tiên tiến để phân tích dữ liệu hình ảnh và video.

### 1.2.4 Tkinter [4]

#### 1.2.4.1 Định nghĩa

Tkinter là thư viện giao diện đồ họa mặc định của Python, cung cấp các công cụ mạnh mẽ để xây dựng ứng dụng có giao diện trực quan. Đây là một gói tích hợp sẵn trong Python, giúp lập trình viên dễ dàng thiết kế và triển khai các ứng dụng desktop mà không cần cài đặt thêm thư viện bên ngoài.

Tkinter dựa trên Tcl/Tk, một nền tảng GUI lâu đời và ổn định. Nhờ sự đơn giản và khả năng tùy chỉnh cao, Tkinter trở thành một trong những lựa chọn phổ biến nhất khi phát triển ứng dụng GUI bằng Python.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 1.4. Hình ảnh minh họa Python 1.4

#### 1.2.4.2 Đặc trưng

* Tích hợp sẵn trong Python: Không cần cài đặt bổ sung, giúp triển khai nhanh chóng.
* Giao diện dễ sử dụng: Cung cấp các widget như nút bấm, nhãn, ô nhập liệu, hộp thoại, giúp xây dựng giao diện dễ dàng.
* Đa nền tảng: Chạy trên Windows, macOS và Linux mà không cần thay đổi mã nguồn.
* Hỗ trợ sự kiện và tương tác: Cho phép xử lý sự kiện từ người dùng như nhấn chuột, nhập liệu, kéo thả.
* Tùy chỉnh linh hoạt: Cung cấp nhiều tùy chọn để thiết kế giao diện theo nhu cầu.

#### 1.2.4.3 Ứng dụng

Tkinter được sử dụng rộng rãi để tạo các ứng dụng có giao diện đồ họa, bao gồm:

* Phát triển công cụ hỗ trợ: Viết các ứng dụng desktop như trình soạn thảo văn bản, công cụ quản lý dữ liệu.
* Ứng dụng giáo dục: Dùng để tạo phần mềm học tập, trình giả lập hoặc giao diện điều khiển đơn giản.
* Tích hợp với các thư viện khác: Kết hợp với NumPy, Matplotlib để tạo công cụ trực quan hóa dữ liệu.

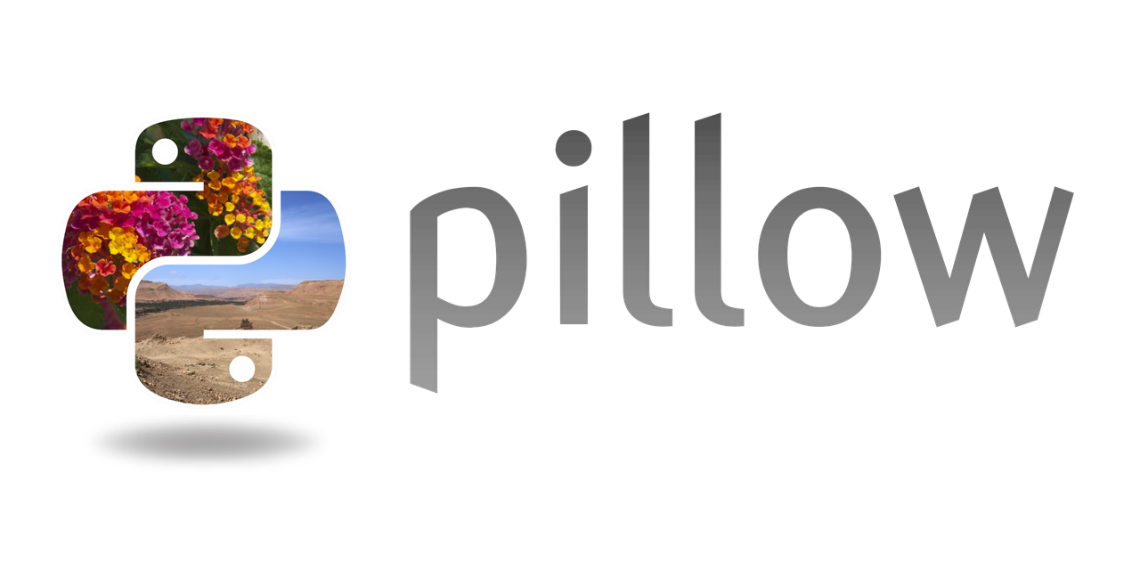
Với sự tiện lợi và tính linh hoạt, Tkinter là một lựa chọn lý tưởng cho những ai muốn xây dựng ứng dụng GUI với Python một cách nhanh chóng và hiệu quả.

### 1.2.5 PIL [5]

#### 1.2.5.1 Định nghĩa

PIL là một thư viện mạnh mẽ hỗ trợ xử lý ảnh trong Python. Tuy nhiên, dự án PIL gốc đã ngừng phát triển và được thay thế bởi Pillow, một nhánh nâng cấp của PIL, cung cấp nhiều tính năng hơn và vẫn được bảo trì đến nay.

Pillow cho phép thao tác với nhiều định dạng ảnh khác nhau như JPEG, PNG, BMP, GIF và TIFF. Với cú pháp đơn giản và khả năng mở rộng tốt, thư viện này là lựa chọn phổ biến trong các ứng dụng xử lý ảnh số, từ chỉnh sửa cơ bản đến các thao tác nâng cao như lọc ảnh và nhận dạng đối tượng.



Hình 1.5. Hình ảnh minh họa PIL

#### 1.2.5.2 Đặc trưng

* Hỗ trợ đa định dạng ảnh: Có thể đọc, ghi và chuyển đổi giữa nhiều định dạng tệp ảnh khác nhau.
* Chỉnh sửa ảnh dễ dàng: Cung cấp các công cụ để thay đổi kích thước, cắt, xoay, lật ảnh.
* Xử lý màu sắc: Hỗ trợ điều chỉnh độ sáng, độ tương phản, áp dụng bộ lọc màu, chuyển đổi ảnh sang grayscale.
* Thêm văn bản và vẽ hình lên ảnh: Cho phép vẽ các hình cơ bản, thêm văn bản với phông chữ tùy chỉnh.
* Tích hợp với các thư viện khác: Có thể sử dụng cùng NumPy, OpenCV để xử lý ảnh nâng cao và ứng dụng trong học máy.

#### 1.2.5.3 Ứng dụng

PIL/Pillow được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

* Chỉnh sửa ảnh tự động: Cắt, thay đổi kích thước, áp dụng bộ lọc cho hàng loạt ảnh.
* Nhận diện ký tự quang học (OCR): Hỗ trợ tiền xử lý ảnh trước khi sử dụng với thư viện nhận dạng chữ như Tesseract.
* Xử lý ảnh trong web: Ứng dụng trong các trang web để tạo ảnh thumbnail, nén ảnh.
* Thêm watermark: Đóng dấu bản quyền lên ảnh tự động.
* Thị giác máy tính: Kết hợp với OpenCV để nhận diện khuôn mặt, phát hiện đối tượng.

PIL/Pillow là một công cụ hữu ích cho lập trình viên khi làm việc với hình ảnh trong Python.

### 1.2.6 CSV [6]

#### 1.2.6.1 Định nghĩa

CSV là một định dạng tệp văn bản đơn giản dùng để lưu trữ dữ liệu dạng bảng. Trong Python, thư viện csv cung cấp các công cụ giúp thao tác với tệp CSV một cách dễ dàng, bao gồm đọc, ghi và xử lý dữ liệu.

Thư viện csv có sẵn trong Python mà không cần cài đặt bổ sung. Nó cho phép làm việc với dữ liệu có cấu trúc một cách hiệu quả, đặc biệt hữu ích trong phân tích dữ liệu, xử lý thông tin từ tệp log, hoặc trao đổi dữ liệu giữa các ứng dụng.

#### 1.2.6.2 Đặc trưng

* Đọc và ghi tệp CSV dễ dàng: Cung cấp các phương thức để đọc và ghi dữ liệu nhanh chóng.
* Hỗ trợ nhiều kiểu định dạng: Có thể xử lý CSV với các ký tự phân tách khác nhau như dấu phẩy, dấu chấm phẩy, tab.
* Tương thích với nhiều công cụ khác: Dễ dàng nhập và xuất dữ liệu giữa Python và các công cụ như Excel, Google Sheets, Pandas.
* Xử lý dữ liệu linh hoạt: Hỗ trợ đọc từng dòng, chuyển đổi dữ liệu thành danh sách hoặc từ điển.
* Được tích hợp sẵn trong Python: Không yêu cầu thư viện bên ngoài, dễ sử dụng và triển khai.

#### 1.2.6.3 Ứng dụng

* Phân tích dữ liệu: Đọc và xử lý dữ liệu CSV từ các tệp lớn để phân tích.
* Xuất báo cáo: Lưu kết quả phân tích thành tệp CSV để chia sẻ hoặc lưu trữ.
* Tích hợp hệ thống: Dùng để nhập và xuất dữ liệu giữa các hệ thống phần mềm.
* Lưu trữ nhật ký: Ghi dữ liệu log dưới dạng CSV để dễ dàng kiểm tra và theo dõi.
* Tương tác với cơ sở dữ liệu: Chuyển đổi dữ liệu từ tệp CSV vào cơ sở dữ liệu hoặc ngược lại.

Với khả năng đọc ghi linh hoạt và dễ sử dụng, csv là một công cụ hữu ích khi làm việc với dữ liệu dạng bảng trong Python.

### 1.2.7 PyInstaller [7]

#### 1.2.7.1 Định nghĩa

PyInstaller là một gói gồm nhiều ứng dụng của Python và các thư viện của Python.

PyInstaller đọc một tập lệnh Python do bạn viết. Nó phân tích mã để phát hiện mọi module và thư viện mà tập lệnh của bạn cần để thực thi. Sau đó, nó thu thập các bản sao của tất cả những tệp đó, bao gồm cả trình thông dịch Python đang hoạt động và đặt chúng cùng với tập lệnh của bạn trong một thư mục duy nhất, hoặc tùy chọn trong một tệp thực thi duy nhất.

#### 1.2.7.2 Đặc trưng

* Hỗ trợ tạo tệp thực thi cho Windows, Linux và macOS, nhưng không thể biên dịch chéo.

#### 1.2.7.3 Ứng dụng

* Đóng gói thành một ứng dụng duy nhất
* Có thể chạy ứng dụng mà không cần cài đặt thư viện hay môi trường

# CHƯƠNG 2. XỬ LÝ ẢNH ĐẦU VÀO

## Dữ liệu ban đầu

Dữ liệu đầu vào là một ảnh bài thi của bất kì một học sinh nào thi THPTQG. Ảnh gồm các bài thi:

* 50 câu
* 40 câu
* Các môn tổ hợp 120 câu

## 2.2 Lọc đối tượng

Tệp papercheck có hàm đầu tiên là display để hiển thị và gỡ lỗi với các tham số img, ratio, time\_sec. Dưới đây là mô tả chi tiết hoạt động của hàm:

* + Bước đầu thu nhỏ ảnh theo tỉ lệ.
  + Sau đó hiển thị với hàm cv2.imshow.
  + Tiếp tới hàm cv2.waitKey sẽ hiển thị cho đến khi có phím nào được nhấn (thích hợp cho việc hiển thị hình ảnh).
  + Cuối cùng đóng cửa sổ với cv2.destroyAllWindows.

Sau đó với hàm extract\_part\_area để trích xuất vùng cụ thể từ phiếu thi như mã học sinh, mã đề thi, phần câu trả lời với các tham số image, name. Dưới đây là mô tả chi tiết của hàm:

* + Lấy khoảng diện tích tối thiểu/tối đa cho vùng cần tìm.
  + Tìm các contour phù hợp với khoảng diện tích.
  + Với mỗi contour thỏa mãn, cắt vùng ảnh tương ứng.
  + Sắp xếp các vùng theo tọa độ x tăng dần.
  + Trả về danh sách các vùng đã cắt.

A computer screen shot of a code

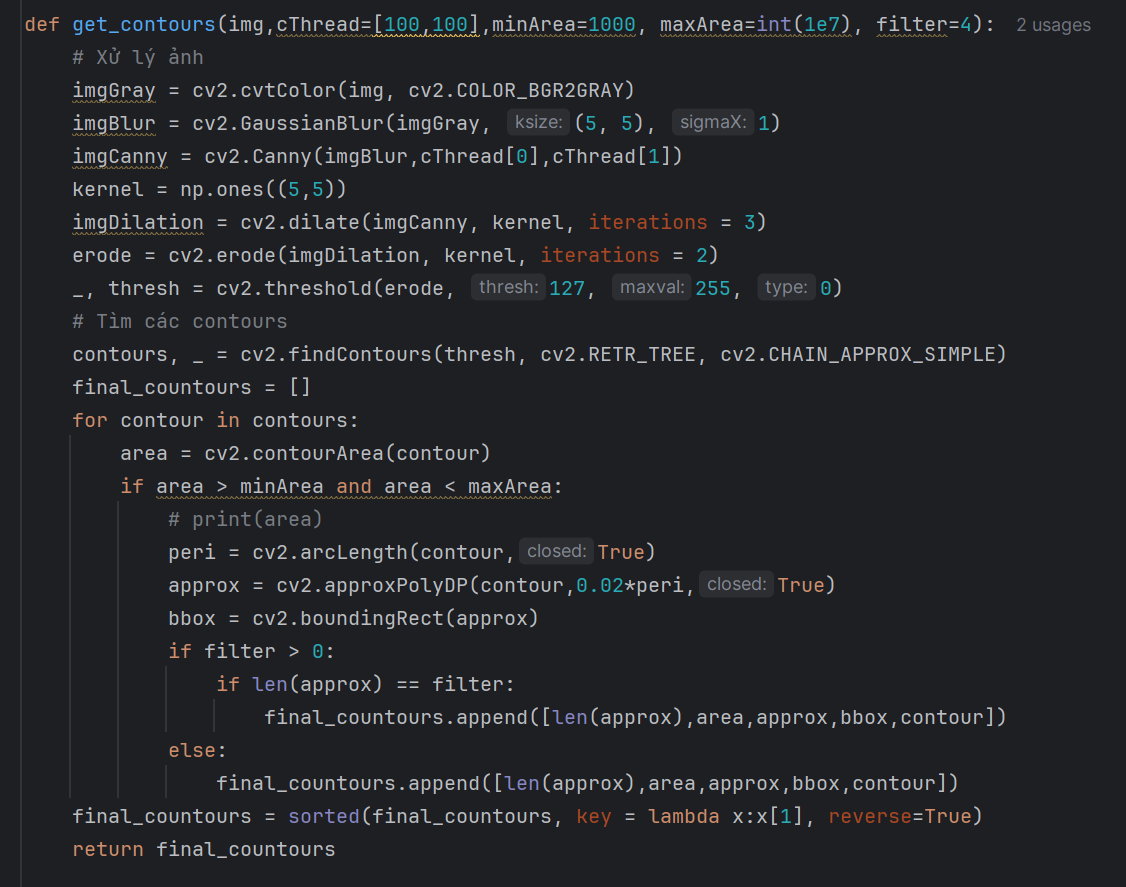
AI-generated content may be incorrect.

Hình 2.1. Hình ảnh minh họa hàm extract\_part\_area

Hàm get\_contours được sử dụng để xác định và lọc các đối tượng trong một hình ảnh. Dưới đây mô tả chi tiết cách hoạt động của hàm:

* Đầu tiên hình ảnh đước truyền vào sẽ được chuyển thành màu xám.
* Sau đó nó sẽ được làm mịn đồng thời giảm nhiễu với GaussianBlur.
* Sau đó biến đổi thành hình ảnh Canny để tìm biên của các đối tượng.
* Hàm cv2.dilate và cv2.erode được sử dụng để làm nổi bật và loại bỏ nhiễu trong hình ảnh Canny.
* Hàm cv2.findContours tìm các contours trong hình ảnh đã được xử lý và lưu chúng trong biến contours.
* Tiếp theo, hàm xác định các contours hợp lệ dựa trên diện tích và số cạnh.
* Danh sách contours hợp lệ được sắp xếp theo diện tích giảm dần.
* Hàm trả về hình ảnh gốc và danh sách contours hợp lệ.

Chức năng này có thể được sử dụng để xác định và lọc các đối tượng trong hình ảnh dựa trên diện tích và số cạnh của chúng. Đây là đoạn mã của hàm get\_contours:



Hình 2.2. Hình ảnh minh họa hàm get\_contours

## 2.3 Cắt hình ảnh

Hàm wrap\_image để thực hiện warp trên hình ảnh đầu vào img dựa trên contours trong danh sách points và kích thước đích (widthImg và heightImg). Hàm này được sử dụng để thực hiện crop và biến đổi hình ảnh để chỉ còn lại một phần cụ thể của hình ảnh gốc, và bỏ padding từ hình ảnh đích nếu được chỉ định. Hàm cụ thể như sau:

* img: Đây là hình ảnh đầu vào mà bạn muốn biến đổi.
* points: Đây là danh sách chứa contour sử dụng để xác định phép biến đổi. Mã yêu cầu rằng danh sách này phải chứa chính xác 4 điểm.
* widthImg và heightImg: Đây là kích thước mục tiêu(width và height) cho hình ảnh đầu ra sau biến đổi. Hình ảnh sẽ được biến đổi để có kích thước này.
* pad: Đây là một tham số tùy chọn để xác định phạm vi của padding được loại bỏ khỏi hình ảnh đầu ra. Giá trị mặc định là 0, tức là không có lề.
* Trước hết, hàm gọi get\_4\_contour(points) để đảm bảo rằng danh sách
* points chứa chính xác 4 điểm. Điều này đảm bảo rằng phép biến đổi sẽ được thực hiện với 4 điểm duy nhất.
* Tiếp theo, hàm tạo hai mảng numpy pts1 và pts2. pts1 chứa tọa độ của 4 điểm trong points, và pts2 chứa 4 điểm với các tọa độ được xác định bởi[[0, 0], [widthImg, 0], [0, heightImg], [widthImg, heightImg]]. Các điểm trong pts1 và pts2 sẽ được sử dụng để xác định phép biến đổi. Chức năng chính của hàm này là thực hiện phép biến đổi hình ảnh dựa trên các điểm xác định, và sau đó bỏ padding nếu được chỉ định.
* Hàm cv2.getPerspectiveTransform(pts1, pts2) sử dụng để tính toán ma trận biến đổi (perspective transformation matrix) dựa trên tọa độ của các điểm trong pts1 và pts2. Ma trận này sẽ được sử dụng để biến đổi hình ảnh.
* Hàm cv2.warpPerspective được sử dụng để thực hiện phép biến đổi của hình ảnh đầu vào img bằng cách sử dụng ma trận biến đổi matrix. Kết quả là hình ảnh đã biến đổi với kích thước mới.
* Sau đó, dòng mã wrap = wrap[pad:wrap.shape[0]-pad, pad:wrap.shape[1]- pad] được sử dụng để bỏ padding từ hình ảnh đầu ra, nếu pad lớn hơn 0.

Với hàm get\_4\_contour là hàm nhận vào một danh sách points chứa các contours và mục đích của hàm là tìm và trả về một mảng chứa 4 điểm đại diện cho 4 điểm góc cụ thể của hình ảnh.

Dưới đây là hình ảnh của hàm wrap\_image và get\_4\_contour:

A computer screen shot of a code

AI-generated content may be incorrect.

Hình 2.3. Hình ảnh minh họa hàm wrap\_image

A screen shot of a computer code

AI-generated content may be incorrect.

Hình 2.4. Hình ảnh minh họa hàm get\_4\_contour

# CHƯƠNG 3. MÔ HÌNH HÓA BÀI TOÁN

## Mô tả bài toán

Để giải quyết bài toán này ta sử dụng các phương pháp xử lý ảnh và ứng dụng thư viện OpenCV để nhận dạng các đáp án, số báo danh và mã đề thi của thí sinh đã tô. Sau đó so sánh với đáp án đã được nhập trước đó rồi trả ra điểm số của thí sinh, số báo danh và mã đề thi của thí sinh đó.

## 3.2. Xây dựng mô hình

Sau khi đã xây dựng các hàm liên quan đến việc xử lý ảnh thì chúng em bắt đầu xây dựng mô hình giải quyết bài toán chấm điểm thi THPTQG.

### 3.2.1 Cắt vùng phiếu

Lớp Detect sử dụng hàm xử lý ảnh chuẩn hóa kích thước ảnh và trích xuất các vùng quan trọng như mã đề thi, mã thí sinh, vùng phiếu trả lời từ đó cung cấp vùng ảnh đã cắt cho lớp Grade.

* Đầu tiên lớp Detect chuẩn hóa kích thước ảnh giúp thuật toán xử lý ảnh hoạt động nhất quán bất kể kích thước ảnh ban đầu. Gồm 2 thuộc tính là width, height.
* Tiếp theo phương thức \_\_init\_\_ với tham số đầu vào là path\_image, find\_exam. Có chức năng đọc ảnh từ đường dẫn với cv2.imread sau đó chuẩn hóa về kích thước chuẩn và nếu find\_exam là True sẽ cắt chính xác bài thi từ ảnh.

Hàm \_find\_exam giúp chấm ảnh tốt hơn khi duỗi thẳng ảnh, hạn chế phần dư thừa khi chấm bài. Cụ thể:

* + tìm contour lớn nhất trong ảnh với diện tích cho trước,
  + Sau đó với get\_4\_contour() từ papercheck.py để xác định 4 góc của bài thi.
  + Sử dụng perspective transform qua wrap\_image() để duỗi thẳng bài thi, loại bỏ biến dạng do góc quét và tạo hình ảnh vuông vắn.

Hàm get\_exam trả về hình ảnh đã xử lý để cung cấp cho lớp Grade.

Hàm get\_sheet\_ans trích xuất vùng câu trả lời trắc nghiệm.

Hàm get\_test\_code trích xuất cùng chứa mã đề.

Hàm get\_student\_code trích xuất vùng chứa mã thí sinh.

A computer screen shot of a program code

AI-generated content may be incorrect.

Hình 3.1. Hình ảnh minh họa lớp Detect

### 3.2.2 Chấm điểm

Lớp Grading là thành phần trung tâm của hệ thống chấm điểm tự động cho bài thi trắc nghiệm, có tác dụng phân tích hình ảnh bài thi đã quét, nhận diện câu trả lời, so sánh với đáp án, và tính điểm.

Lớp Grading nhận các tham số:

* path: Đường dẫn đến hình ảnh bài thi.
* answer\_keys: Mảng chứa đáp án đúng (dạng 'A', 'B', 'C', 'D').
* num\_questions: Số lượng câu hỏi.

A computer screen shot of a program code

AI-generated content may be incorrect.

Hình 3.2. Hình ảnh minh họa lớp Grading

Hàm get\_answers trích xuất tất cả câu trả lời từ phiếu trả lời. Cụ thể:

* Lấy vùng chứa câu trả lời từ ảnh qua detector.get\_sheet\_ans()
* Chia vùng trả lời thành 6 phần (mỗi phần có 5 câu)
* Xử lý từng phần và thêm câu trả lời vào mảng full\_answers
* Dừng khi đã có đủ số câu trả lời cần thiết

Hàm grading\_sheet phân tích một phần của phiếu trả lời. Cụ thể:

* Chia hình ảnh thành lưới 5x4 (5 hàng - câu hỏi, 4 cột - các lựa chọn A,B,C,D)
* Sử dụng HoughCircles để nhận diện các hình tròn (ô tròn đã tô)
* Xác định các ô được tô dựa trên giá trị màu trung bình
* Ánh xạ vị trí của ô được tô thành câu trả lời (A, B, C, D)
* Đánh dấu câu trả lời vào ảnh kết quả:
  + Màu xanh: câu trả lời đúng
  + Màu đỏ: câu trả lời sai

Hàm get\_score tính điểm bằng cách đếm số câu trả lời đúng, so sánh với đáp án, sau đó nhân với 10 để đưa về thang điểm 10.

Hàm extract\_code trích xuất mã thí sinh hoặc mã đề thi từ vùng tương ứng trên bài thi. Cụ thể:

* Xác định layout dựa trên loại mã (student\_code: 10x6, test\_code: 10x3)
* Sử dụng HoughCircles để nhận diện các ô tròn đã tô
* Xác định hàng và cột của các ô đã tô
* Đánh dấu các ô đã tô trên ảnh kết quả
* Ghi số tương ứng với hàng được tô

Hàm extract\_student\_code trích xuất mã thí sinh, extract\_test\_code trích mã đề.

Hàm get\_result\_image trả về ảnh đã được đánh dấu với kết quả.

# CHƯƠNG 4 . CÀI ĐẶT VÀ THỬ NGHIỆM

## Môi trường cài đặt

Cài đặt các thư viện cần thiết được lưu trong file requirements.txt. bằng lệnh:

pip install -r requirements.txt

## 4.2. Thử nghiệm

### 4.2.1. Chức năng Chọn ảnh

* + - * + Người dùng có thể chọn một hình ảnh từ máy tính thông qua nút Chọn Ảnh trong giao diện ứng dụng.
        + Hình ảnh sau khi được chọn sẽ được hiển thị trong cửa sổ ứng dụng.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.1. Hình ảnh minh họa chức năng Chọn Ảnh

### 4.2.2 Chức năng Tải Đáp Án

* Tự nhập đáp án thủ công với mỗi đáp án từ trái qua ứng với các câu từ số 1 trở đi trong phiếu thi.
* Tải lên file csv trong đã có sẵn các đáp án tương ứng với câu trong phiếu thi.

A screenshot of a table

AI-generated content may be incorrect.

* Cuối cùng lưu đáp án để chấm điểm thi.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2. Hình ảnh minh họa chức năng Lưu Đáp Án

### 4.2.3 Chức năng Chấm Điểm

* Khi người dùng nhấn vào nút Chấm Điểm, ứng dụng thực hiện các bước sau:
* Xử lý hình ảnh đầu vào để trích xuất thông tin bài thi.
* Tính điểm dựa trên thông tin đã trích xuất và so sánh với đáp án được cung cấp (answer\_key).
* Hiển thị điểm số, mã số sinh viên, và mã đề thi lên hình ảnh.
* Hiển thị hình ảnh đã được chấm điểm trong ứng dụng giao diện.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.3. Hình ảnh minh họa chức năng Chấm Điểm

# KẾT LUẬN

Báo cáo này trình bày toàn bộ quá trình xây dựng và triển khai hệ thống chấm điểm tự động cho bài thi THPTQG, từ khâu thiết kế, cài đặt đến thử nghiệm:

* + - * Ưu điểm
        + Độ chính xác cao trong quá trình nhận diện và chấm điểm.
        + Dễ dàng cài đặt và sử dụng.
      * Nhược điểm
        + Chưa được tự động hóa hoàn toàn và vẫn cần nhập đáp án thủ công.
        + Bắt buộc ảnh phải được quét chính xác và đã xuất ra file.

Sau khi hoàn thành dự án hiện tại, chúng em nhận thấy có thể nâng cấp và mở rộng ứng dụng này để cho phép người dùng chấm điểm cho bất kỳ bài thi nào, chẳng hạn như bài thi TOEIC hoặc bất kỳ bài trắc nghiệm nào khác. Điều này đòi hỏi việc thay đổi và bổ sung chức năng để hỗ trợ việc chấm điểm cho các loại bài thi khác nhau.

Các tính năng có thể cải tiến trong tương lai:

* Hỗ trợ nhiều định dạng bài thi:
* Mở rộng hỗ trợ chấm điểm cho các kỳ thi khác như TOEIC, IELTS, hoặc các bài kiểm tra nội bộ.
* Thống kê và lưu trữ kết quả:
* Lưu lịch sử chấm điểm theo từng thí sinh.
* Xuất kết quả dưới dạng file Excel hoặc PDF.
* Tích hợp chia sẻ kết quả:
* Gửi kết quả qua email.
* Chia sẻ trực tiếp lên các nền tảng như Google Drive, mạng xã hội.

Các cải tiến này sẽ giúp ứng dụng trở thành một công cụ mạnh mẽ để chấm điểm cho nhiều loại bài thi và làm cho quá trình chấm điểm trở nên hiệu quả hơn.

# TÀI LIỆU THAM KHẢO

1. [Welcome to Python.org](https://www.python.org/)
2. [NumPy](https://numpy.org/)
3. [OpenCV - Open Computer Vision Library](https://opencv.org/)
4. [tkinter — Python interface to Tcl/Tk — Python 3.13.2 documentation](https://docs.python.org/3/library/tkinter.html)
5. [pillow · PyPI](https://pypi.org/project/pillow/)
6. [Comma-separated values - Wikipedia](https://en.wikipedia.org/wiki/Comma-separated_values)
7. [PyInstaller Manual — PyInstaller 6.12.0 documentation](https://pyinstaller.org/en/stable/)
8. [Python: Sử dụng PyInstaller để dễ dàng phân phối các ứng dụng Python | V1Study](https://v1study.com/python-bai-hoc-core-su-dung-pyinstaller-de-de-dang-phan-phoi-cac-ung-dung-python.html)
9. [Line, Circle and Blob Detection. HoughLineTransform: | by Riwaj Neupane | Medium](https://medium.com/@RiwajNeupane/line-circle-and-blob-detection-e5a3487dcb1a)
10. [ChatGPT](https://chatgpt.com/)