TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA ĐIỆN-ĐIỆN TỬ**



**FINAL PROJECT**

**DIGITAL IMAGE PROCESSING**

**TS. Le Anh Vu**

Class:N02

Goup 4:

Lê Qúy Hữu \_422H0075

Hoàng Tấn Hưng\_419H0143

Nguyễn Ngọc Luân\_422H0097

**INDEX**

**I/INTRODUCTION:**

**A/PROBLEM:**

Trích xuất thông tin trên giấy khai sinh, giấy kết hôn bằng xử lý ảnh.

- Giấy khai sinh và giấy kết hôn là những tài liệu quan trọng trong việc xác nhận các thông tin cá nhân và gia đình. Những giấy tờ này chứa đựng các dữ liệu quan trọng như tên tuổi, ngày tháng năm sinh, địa chỉ, ngày kết hôn và nhiều thông tin khác. Việc trích xuất thông tin từ các giấy tờ này đóng vai trò quan trọng trong việc giảm bớt khối lượng công việc thủ công của các cơ quan hành chính, đồng thời tiết kiệm thời gian và nâng cao độ chính xác trong công việc. Tuy nhiên, hiện nay, việc trích xuất thông tin từ các giấy tờ hành chính này vẫn chủ yếu được thực hiện thủ công, dẫn đến những vấn đề như sai sót thông tin và mất thời gian.

- Để giải quyết vấn đề này, việc áp dụng công nghệ xử lý ảnh và nhận diện ký tự quang học (OCR) sẽ giúp tự động hóa quá trình trích xuất thông tin, từ đó nâng cao hiệu quả công việc và giảm thiểu sai sót.

**B/MOTIVATION:**

*\* Tiết kiệm thời gian và chi phí:*

- Việc tự động trích xuất thông tin từ giấy khai sinh và giấy kết hôn sẽ giúp giảm thiểu đáng kể thời gian nhập liệu thủ công và chi phí nhân sự

- Thay vì phải mất nhiều giờ để nhập liệu thủ công, nhân viên chỉ cần tải ảnh lên hệ thống và nhận kết quả nhanh chóng

*\* Giảm thiểu sai sót*:

- Việc nhập liệu thủ công dễ dẫn đến các sai sót như nhầm lẫn tên, ngày tháng năm sinh hoặc các lỗi chính tả

- Việc sử dụng công nghệ xử lý ảnh và OCR, hệ thống sẽ giúp giảm thiểu những sai sót này và đảm bảo độ chính xác cao hơn.

*\* Tăng cường khả năng quản lý và tìm kiếm thông tin:* Thông tin sau khi được trích xuất sẽ được số hóa, giúp dễ dàng lưu trữ và tìm kiếm trong các hệ thống cơ sở dữ liệu.

**C/GENERAL IDEAL:**

**1. Tiền xử lý ảnh:**

- Trước khi bắt đầu quá trình nhận diện ký tự, tiền xử lý ảnh đóng vai trò quan trọng trong việc cải thiện chất lượng của hình ảnh đầu vào. Các bước tiền xử lý này sẽ giúp tăng độ chính xác của thuật toán nhận diện ký tự (OCR), đặc biệt là khi hình ảnh có chất lượng thấp, bị mờ hoặc bị nhiễu

- Chuyển ảnh sang ảnh xám: Bước này giúp đơn giản hóa hình ảnh, loại bỏ các yếu tố màu sắc không cần thiết, chỉ giữ lại thông tin về độ sáng tối của các điểm ảnh. Điều này giúp thuật toán dễ dàng nhận diện văn bản hơn.

**2. Nhận diện ký tự (OCR):**

- OCR (Optical Character Recognition) là công nghệ chủ chốt trong việc chuyển đổi các ký tự quang học từ ảnh thành dữ liệu văn bản có thể chỉnh sửa. Hệ thống sử dụng Tesseract OCR, một công cụ mã nguồn mở mạnh mẽ được phát triển bởi Google, để nhận diện các ký tự từ ảnh

- OCR với Tesseract: Tesseract sẽ phân tích các vùng có văn bản trong ảnh và chuyển đổi chúng thành chuỗi ký tự văn bản. Hệ thống sẽ hỗ trợ nhiều ngôn ngữ và có khả năng nhận diện các font chữ khác nhau, từ đó tạo ra kết quả chính xác

1. **Trích xuất thông tin cần thiết:**

**-** Lựa chọn tọa độ có văn bản cần thiết để cắt tùy vào loại giấy tờ được đưa vào

- Sau khi văn bản được nhận diện và xử lý, bước tiếp theo là trích xuất thông tin cần thiết từ văn bản: Tên của người khai sinh hoặc người kết hôn, ngày tháng năm sinh, số căn cước, ngày kết hôn, địa chỉ,….

**D/OBJECTIVES:**

**1/General:** *Mục tiêu chính của dự án là xây dựng một hệ thống tự động trích xuất thông tin từ giấy khai sinh và giấy kết hôn nhằm tăng cường hiệu quả công việc trong các cơ quan hành chính, tiết kiệm thời gian và chi phí, đồng thời nâng cao độ chính xác và giảm thiểu sai sót trong quá trình nhập liệu.*

**2/Specific:**

**a/Technical:**

- Xây dựng hệ thống có thể nhận diện và trích xuất thông tin từ các giấy khai sinh và giấy kết hôn có chất lượng hình ảnh khác nhau, bao gồm cả các ảnh mờ hoặc có nhiễu

- Áp dụng các thuật toán xử lý ảnh và OCR như Tesseract để chuyển đổi văn bản từ hình ảnh thành dạng dữ liệu có thể chỉnh sửa và sử dụng.

**b/Sepecs:**

- Hệ thống phải có độ chính xác cao trong việc nhận diện các ký tự từ ảnh, đặc biệt là các loại văn bản phức tạp hoặc bị mờ

- Hệ thống cần có khả năng nhận diện các loại giấy tờ với các cấu trúc và font chữ khác nhau.

**c/GUI/UX:** *Hệ thống sẽ cung cấp một giao diện người dùng đơn giản và dễ sử dụng, nơi người dùng có thể tải ảnh của các giấy tờ như giấy khai sinh và giấy kết hôn lên hệ thống. Sau khi tải ảnh, người dùng sẽ nhận được kết quả trích xuất thông tin một cách nhanh chóng:*

- Giao diện đơn giản: Giao diện người dùng cần dễ hiểu và dễ thao tác, cho phép người dùng thực hiện các thao tác một cách mượt mà và hiệu quả.

- Hiển thị kết quả: Sau khi quá trình nhận diện hoàn thành, thông tin trích xuất sẽ được hiển thị dưới dạng bảng hoặc báo cáo, cho phép người dùng kiểm tra kết quả và lưu trữ thông tin.

**d/Tools:**

\*Language: *Python*

\*Library:

- Opencv

- Tesseract OCR

\*Open source: OpenCV và Tesseract OCR

**II/THEORY:**

**1. Xử lý ảnh:**

- Xử lý ảnh là một lĩnh vực trong khoa học máy tính liên quan đến việc áp dụng các thuật toán để thay đổi, phân tích, và xử lý các hình ảnh. Các mục tiêu của xử lý ảnh có thể bao gồm làm mượt ảnh, phát hiện biên, phân đoạn, và cải thiện chất lượng ảnh để làm dễ dàng các tác vụ nhận diện và phân tích

- Một trong những bước đầu tiên trong xử lý ảnh là chuyển đổi ảnh màu (RGB) thành ảnh xám. Quá trình này giúp giảm độ phức tạp trong việc xử lý, vì ảnh xám chỉ có thông tin độ sáng (intensity) mà không có thông tin màu sắc. Điều này giúp các thuật toán nhận diện và phân tích dễ dàng hơn.

1. **Lọc Gaussian:**

- Lọc Gaussian là một kỹ thuật dùng để làm mượt ảnh, giúp giảm nhiễu và làm giảm độ sắc nét của các chi tiết trong ảnh. Lọc này áp dụng một bộ lọc trung bình có dạng phân phối chuẩn (Gaussian) lên mỗi pixel trong ảnh

- Ứng dụng: Trong bài toán nhận diện văn bản, việc làm mượt ảnh giúp giảm bớt các nhiễu không cần thiết, giúp các ký tự trở nên rõ ràng hơn và dễ dàng nhận diện hơn.

1. **Phát hiện biên**:

- Phát hiện biên là một kỹ thuật trong xử lý ảnh để xác định các biên trong ảnh, giúp phân biệt các đối tượng hoặc các khu vực có sự thay đổi lớn về độ sáng.

- Ứng dụng: Việc phát hiện các biên của ký tự hoặc văn bản giúp tách rời văn bản khỏi các phần nền trong ảnh, tạo thuận lợi cho việc nhận diện ký tự.

**4. Nhận diện ký tự quang học (OCR):**

- OCR (Optical Character Recognition) là công nghệ chuyển đổi văn bản trong hình ảnh thành dạng văn bản có thể chỉnh sửa được. Nó được sử dụng rộng rãi trong việc chuyển đổi tài liệu văn bản quét hoặc ảnh chứa văn bản thành các tài liệu số có thể tìm kiếm và chỉnh sửa

- Tesseract OCR là một công cụ mã nguồn mở mạnh mẽ được sử dụng trong dự án này để nhận diện văn bản trong ảnh. Nó hỗ trợ nhiều ngôn ngữ và có thể nhận diện văn bản trong các định dạng hình ảnh khác nhau

- Cách hoạt động của OCR: sử dụng các thuật toán để phân tích các hình dạng của các ký tự trong ảnh, sau đó so khớp chúng với một cơ sở dữ liệu các ký tự đã biết

- Tiền xử lý ảnh: Trước khi thực hiện OCR, ảnh thường phải được xử lý để làm sạch và tăng cường chất lượng nhằm tăng độ chính xác của OCR

- Chuyển đổi hình ảnh thành văn bản: Sau khi tiền xử lý, OCR sẽ phân tích hình ảnh để nhận diện và trích xuất văn bản từ đó.

**5. Trích xuất thông tin từ văn bản:**

- Sau khi văn bản được nhận diện từ hình ảnh, bước tiếp theo là trích xuất thông tin từ văn bản đó. Đây là quá trình xử lý văn bản đã nhận diện để tìm ra các thông tin quan trọng như: tên người, ngày tháng năm sinh, số chứng minh nhân dân,…

**III/IMPLEMENTATION:**

**A/BLOCK DIAGRAM:**

1. Preprocessing
2. Input Image

**1.Input Image:** Người dùng tải lên ảnh giấy khai sinh hoặc giấy kết hôn

**2.Preprocessing:** Tiền xử lý ảnh (chuyển sang ảnh xám, làm mượt, thresholding)

**3.Text Detection:** Phát hiện các vùng chứa văn bản

**4.OCR (Text Recognition):** Nhận diện văn bản từ các vùng đã phát hiện

**5.Postprocessing:** Làm sạch văn bản và trích xuất thông tin cần thiết

**6.Output:** Kết quả trích xuất thông tin .

6. Out put

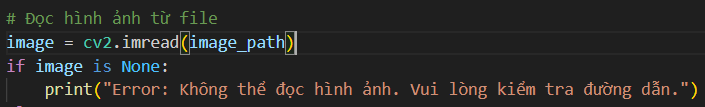
5. Post Processing

4. ORC(Text Recognition)

3.Text Detection

**B/AGORITHN:**

1. **Đọc hình ảnh và kiểm tra:**



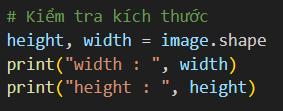
Sử dụng cv2.imread(image\_path) để đọc ảnh từ đường dẫn được cung cấp trong biến image\_path. Nếu không thể đọc ảnh chương trình sẽ in thông báo lỗi và dừng lại.

1. **Chuyển sang ảnh xám(Gray Scale):**



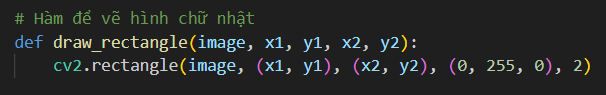
Sử dụng cv2.cvtColor() để chuyển đổi ảnh màu (mã màu BGR trong OpenCV) sang ảnh xám. Việc này giúp giảm độ phức tạp của hình ảnh và làm tăng hiệu quả nhận diện văn bản, vì ảnh xám chỉ có thông tin độ sáng, không cần phải xử lý các thông tin màu sắc.

1. **Kiểm tra kích thước ảnh:**



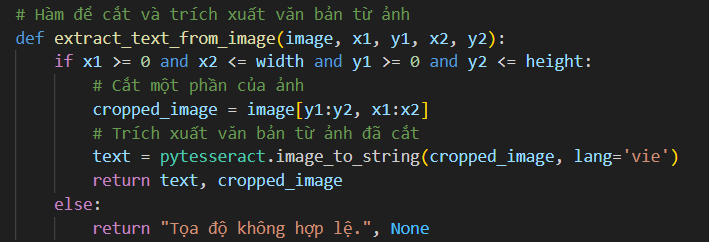
In ra kích thước của ảnh để giúp người dùng có thể kiểm tra thông tin về kích thước ảnh, phục vụ cho các bước xử lý tiếp theo.

**4.Vẽ hình chữ nhật**:



Vẽ hình chữ nhật lên ảnh tại vị trí có tọa độ góc trên trái (x1, y1) và góc dưới phải (x2, y2). Màu sắc của hình chữ nhật là màu xanh lá cây (0, 255, 0), với độ dày của đường viền là 2 pixel. Hàm này giúp xác định và làm rõ các vùng trong ảnh mà bạn muốn cắt và trích xuất văn bản.

5.Trích xuất văn bản:



- Đầu tiên, hàm kiểm tra các tọa độ cắt có hợp lệ không (x1, y1 là điểm góc trên trái và x2, y2 là điểm góc dưới phải của vùng cắt). Nếu các tọa độ này hợp lệ (chúng phải nằm trong phạm vi kích thước ảnh), ảnh sẽ được cắt ra bằng cách sử dụng image[y1:y2, x1:x2].

- Sau đó, hàm pytesseract.image\_to\_string() được gọi để trích xuất văn bản từ phần ảnh đã cắt. Tham số lang='vie' giúp xác định rằng văn bản cần nhận diện là tiếng Việt.

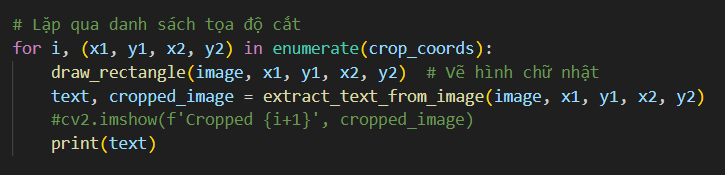
- Hàm trả về hai giá trị: văn bản đã trích xuất và ảnh đã cắt.

1. **Danh sách tọa độ cắt**:



Một danh sách các bộ tọa độ (x1, y1, x2, y2) chỉ định các vùng cần cắt trong ảnh. Các tọa độ này được tính toán dựa trên tỷ lệ phần trăm chiều rộng và chiều cao của ảnh (width \* 0.3, height \* 0.25, v.v.). Mỗi bộ tọa độ chỉ ra một phần của ảnh sẽ được cắt ra để trích xuất văn bản.

1. **Lặp qua danh sách các tọa độ và thực hiện vẽ hình chữ nhật và trích xuất văn bản:**

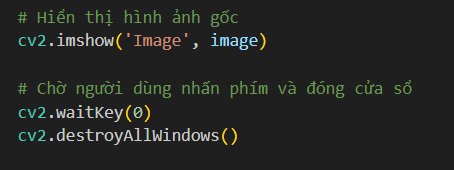


- Lặp qua tất cả các tọa độ trong danh sách crop\_coords. Mỗi tọa độ (x1, y1, x2, y2) chỉ định một phần của ảnh cần được cắt.

- Vẽ hình chữ nhật: draw\_rectangle() được gọi để vẽ hình chữ nhật bao quanh vùng cắt, giúp xác định rõ ràng khu vực cần trích xuất văn bản.

- Trích xuất văn bản: Hàm extract\_text\_from\_image() được gọi để cắt ảnh và trích xuất văn bản từ khu vực đó. Kết quả văn bản được in ra màn hình.

1. Hiển thị ảnh gốc với hình chữ nhật được vẽ:



Sau khi vẽ các hình chữ nhật và trích xuất văn bản từ các khu vực đã cắt, sử dụng cv2.imshow() để hiển thị ảnh gốc với các hình chữ nhật đã vẽ lên đó. cv2.waitKey(0) đảm bảo cửa sổ hiển thị ảnh sẽ không tự động đóng cho đến khi người dùng nhấn phím. Sau đó, cv2.destroyAllWindows() đóng tất cả các cửa sổ hiển thị ảnh.

**C/DELOPMENT:**

**IV/ EXPERIMANTAL RESULTS:(support)**

**A/EXPERIMENT SETUP**

**B/VISUAL DATA, NUMBER DATA**

**V/DISCUSSION RESULTS:**