

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KÌ:
SỐ HÓA TỬ SÁCH

SINH VIÊN THỰC HIỆN: NGUYỄN GIA BẢO NGỌC – 21520366 (33,3%)
NGUYỄN QUỐC TRƯỜNG AN – 21521810 (33,3%)
NGUYỄN ĐỨC TÚ – 21521612 (33,3%)

LỚP: CS114.O11

GIẢNG VIÊN HƯỚNG DẪN:

PhGS.TS Lê Đình Duy

ThS Phạm Nguyễn Trường An

TP. HỒ CHÍ MINH – Tháng 1 năm 2024

TÓM TẮT ĐỒ ÁN

TÊN ĐỒ ÁN TIẾNG VIỆT: SỐ HÓA TỦ SÁCH

TÊN ĐỒ ÁN TIẾNG ANH: DIGITALIZING BOOKSHELVES

Giảng viên hướng dẫn: PhGS.TS. Lê Đình Duy, ThS. Phạm Nguyễn Trường An.

Sinh viên thực hiện: Nguyễn Gia Bảo Ngọc – 21520366.

Nguyễn Quốc Trường An – 21521810.

Nguyễn Đức Tú – 21521612.

Ngôn ngữ lập trình: Python.

Môi trường lập trình: Google Colab.

Github: <https://github.com/ngbn111723/CS114.O11-21520366.git>

Tóm tắt đề tài:

- Mô tả bài toán: Đề tài của nhóm với mục tiêu là phát triển một hệ thống bao gồm mô hình Máy học, cho phép người dùng đưa vào ảnh là trang bìa một cuốn sách, hệ thống sẽ nhận dạng được và xuất ra tên của quyển sách đó.
 - Input: Ảnh bìa quyển sách.
 - Output: Tên của quyển sách.
- Ngữ cảnh ứng dụng: Sản phẩm của đề tài có thể được ứng dụng cho các tổ chức, cá nhân có nhu cầu lập danh sách quản lý sách, các loại văn bản gần giống sách.
- Lý do cần sử dụng mô hình Máy học: trong một bức ảnh bìa sách có thể xuất hiện nhiều vùng văn bản, nhưng không phải tên của sách, như tên nhà xuất bản, tên tác giả.... Do đó hệ thống cần áp dụng mô hình Máy học kết hợp các công cụ hỗ trợ khác để phân loại được chính xác và xuất ra tên sách.
- Số lượng phần tử trong tập dữ liệu (dùng để huấn luyện và kiểm tra mô hình Máy học) bao gồm **5046** phần tử được rút trích từ **700** ảnh bìa sách (tiếng Việt hoặc tiếng Anh). Ngoài ra, nhóm còn sử dụng **70** ảnh bìa sách khác để kiểm tra hệ thống. Ảnh bìa sách được chụp bằng camera, nguồn sách từ các thư viện, cá nhân, internet, các nguồn công khai khác.
- Mô tả hệ thống:
 - Giai đoạn thứ nhất (Text Detection): sử dụng công cụ PyTesseract để nhận diện vùng văn bản trên ảnh bìa sách.
 - Giai đoạn thứ hai (Classification): sử dụng mô hình Máy học để phân loại vùng văn bản trên ảnh bìa sách có phải tựa sách không (được nhóm huấn luyện từ tập dữ liệu thu thập được).
 - Giai đoạn thứ ba (Text Extraction): xuất ra văn bản từ vùng văn bản được phân loại là tựa sách.

MỤC LỤC:

Danh mục ảnh:III

Danh mục bảngIV

CHƯƠNG 0: UPDATE SAU KHI VẤN ĐÁP 5

1. Cập nhật thứ nhất:5

2. Cập nhật thứ hai:6

CHƯƠNG 1: TỔNG QUAN ĐỒ ÁN 7

1. Các bài viết tham khảo:7

1.1 Bài viết tham khảo số 1:7

1.2 Bài viết tham khảo số 2:8

2. Tổng quan đồ án8

2.1 Mô tả đề tài và các ngữ cảnh ứng dụng:8

2.2 Mô tả bộ dữ liệu:9

2.3 Mô tả thuật toán Máy học và các công cụ hỗ trợ:10

2.4 Mục tiêu của đề tài:11

CHƯƠNG 2: CHI TIẾT VÀ HIỆN THỰC ĐỒ ÁN..... 12

1. Machine learning pipeline12

1.1 Pha ‘Training’:12

1.2 Pha ‘Serving’:13

2. Xây dựng bộ dữ liệu16

2.1 Thu thập dữ liệu ban đầu (Raw Data):16

2.2 Kiểm định dữ liệu ban đầu (Data Validation):16

2.3 Rút trích đặc trưng và đánh nhãn dữ liệu (Data Preparation):17

3. Huấn luyện và đánh giá mô hình Binary Classification20

3.1 Các thuật toán được dùng để huấn luyện mô hình Binary Classification20

3.2 Huấn luyện mô hình Binary Classification20

3.3 Đánh giá mô hình Binary Classification21

CHƯƠNG 3: ĐÁNH GIÁ HỆ THỐNG VÀ KẾT LUẬN..... 22

1. Đánh giá hệ thống22

1.1 Đánh giá giai đoạn thứ nhất (Text Detection)22

1.2 Đánh giá giai đoạn thứ hai (Classification)23

1.3 Đánh giá giai đoạn thứ ba (Text Extraction)23

1.4	Đánh giá hệ thống.....	24
2.	Hướng phát triển tiếp theo và kết luận đề tài hiện tại	24
2.1	Hướng phát triển tiếp theo	24
2.2	Kết luận đề tài hiện tại	25
CHƯƠNG 4: CÁC NGUỒN THAM KHẢO		25

Danh mục ảnh:

Hình 1 - Ví dụ về file "select_by_size.txt"	5
Hình 2 - Mô tả kết quả thực hiện đề tài	9
Hình 4 - Mô tả bộ dữ liệu	10
Hình 5 - Mô tả cơ bản quá trình hoạt động của hệ thống	11
Hình 6 - Machine Learning Pipeline	12
Hình 7 - Ví dụ về một mẫu dữ liệu ban đầu	16
Hình 8 - Ví dụ về một mẫu dữ liệu ban đầu đã qua kiểm định.....	16
Hình 9 - Ví dụ về quá trình khoanh vùng văn bản và trích xuất dữ liệu	18
Hình 10 - file "map_final.txt"	19
Hình 11 - Dữ liệu thu được sau đánh nhãn.....	20
Hình 12 - Thông số các mô hình Binary Classification.....	22
Hình 13 - "Confusion matrix" của mô hình Binary Classification	22
Hình 14 - Ví dụ về cách đánh giá giai đoạn Text Detection.....	23
Hình 15 - Kết quả đánh giá giai đoạn Text Extraction	23
Hình 16 - Minh họa các tiêu chuẩn đánh giá hệ thống	24

Danh mục bảng

Bảng 1 - Hàm detect_text	13
Bảng 2 - Pha Serving trong Machine Learning Pipeline	14
Bảng 3 - Hàm "process_img"	15
Bảng 4 - Hàm "enhance_image_for_ocr"	15
Bảng 5 - Hàm khoanh vùng và trích xuất đặc trưng từ vùng văn bản	18
Bảng 6 - Khai báo các đối tượng mô hình Máy học.....	20
Bảng 7 - Tiến hành huấn luyện mô hình Binary Classification.....	21

CHƯƠNG 0: UPDATE SAU KHI VẤN ĐÁP

1. Cập nhật thứ nhất:

Sau buổi vấn đáp, nhóm xin trình bày cập nhật thứ nhất của đề tài. Theo đó, vấn đề đặt ra là ta có thể thay thế cách thức thực hiện giai đoạn thứ hai của hệ thống (phân loại vùng văn bản có là tựa của quyển sách không) bằng cách áp dụng phương pháp khác đơn giản hơn việc áp dụng mô hình Máy học hay không, cụ thể là phân loại vùng văn bản có diện tích lớn nhất là tựa sách. Để làm rõ vấn đề, nhóm thực hiện cách phân loại tựa sách dựa trên kích thước vùng văn bản, thay thế cho mô hình Máy học, trên một tập dữ liệu kiểm tra, để tính toán các thông số ‘precision’, ‘recall’, ‘f1-score’ và ‘accuracy’.

Các thông số ‘precision’, ‘recall’, ‘f1-score’ và ‘accuracy’ nêu trên được nhóm tính toán từ tập dữ liệu gồm **1262** phần tử (tương ứng với số lượng tập ‘test’ dùng để kiểm tra mô hình Máy học) lưu trong file “select_by_size.txt”. Để tạo ra được file dữ liệu “select_by_size.txt”, nhóm đã thực hiện lấy ra cột đặc trưng ‘size’ (đặc trưng nói lên diện tích của vùng văn bản) trong tập dữ liệu của nhóm (từ file “final_data_set.txt”), sau đó sử dụng file “map_final.txt” để biết được đặc trưng ‘size’ thuộc về ảnh nào (chức năng của file “map_final.txt” được trình bày trong phần “2.3 Rút trích đặc trưng và đánh nhãn dữ liệu (Data Preparation)”). Sau khi biết được đặc trưng ‘size’ của từng vùng văn bản trong các ảnh, bước tiếp theo là đánh nhãn cho từng mẫu là ‘1’ hoặc ‘0’ (là tựa sách hay không là tựa sách) dựa theo tiêu chí là vùng văn bản lớn nhất có trong ảnh sẽ là tựa sách và được đánh nhãn **1**. Ngoài ra, nhóm còn lấy ra cột cuối cùng từ tập dữ liệu của nhóm (cột cho biết vùng văn bản có thật sự là tựa sách hay không) làm cột cuối cùng trong file “select_by_size.txt”. Dưới đây là ví dụ về một đoạn của file “select_by_size.txt”.

1	377145	0	0
1	1144250	0	1
1	298660	0	1
1	7721125	1	0
2	756	0	0
2	2022720	0	0
2	1634	0	0
2	652534	0	0
2	245168	0	0
2	309379	0	0
2	26565	0	0
2	21	0	0
2	14490	0	0

Hình 1 - Ví dụ về file “select_by_size.txt”

- Mỗi hàng đại diện cho một vùng văn bản với đặc trưng ‘size’ ở cột thứ hai.
- Cột thứ nhất cho biết ảnh nào (img (1), img (2),...) rút trích ra đặc trưng ‘size’ ở cột thứ hai.
- Cột thứ hai là đặc trưng ‘size’ được lấy ra từ tập dữ liệu là đặc trưng thể hiện diện tích vùng văn bản.
- Cột thứ ba thể hiện vùng văn có phải là tựa sách hay không dựa trên tiêu chí: vùng có diện tích lớn nhất là tựa sách.

- Cột thứ tư thể hiện vùng văn bản có thật sự là tựa sách hay không.

Từ file “select_by_size.txt” được tạo thành, tiếp tục tiến hành tính toán các số liệu là:

- TP: số lượng các trường hợp trên cùng một hàng, cột thứ ba và cột thứ tư cùng bằng 1 (phân loại đúng vùng văn là tựa sách).
- FP: số lượng các trường hợp trên cùng một hàng, cột thứ ba bằng 1 và cột thứ tư bằng 0 (phân loại sai vùng văn là tựa sách).
- TN: số lượng các trường hợp trên cùng một hàng, cột thứ ba và cột thứ tư cùng bằng 0 (phân loại đúng vùng văn không là tựa sách).
- FN: số lượng các trường hợp trên cùng một hàng, cột thứ ba bằng 0 và cột thứ tư bằng 1 (phân loại sai vùng văn không là tựa sách).

Từ các số liệu trên ta có thể tính toán các thông số ‘precision’, ‘recall’, ‘f1-score’ và ‘accuracy’. Dựa trên công thức:

$$Precision = \frac{TP}{TP+FP};$$

$$Recall = \frac{TP}{TP+FN};$$

$$F1 - score = \frac{2TP}{2TP + FP + FN}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

Các thông số thu được lần lượt là:

- Precision : 0.2896174863387978.
- Recall: 0.18661971830985916.
- F1-score: 0.22698072805139186.
- Accuracy: 0.713946117274168.

Từ kết quả trên có thể thấy được độ đánh tin cậy của phương pháp phân loại cho vùng văn bản có diện tích lớn nhất là tựa sách tương đối thấp, dù cho cách thức thực hiện là khá đơn giản. Như vậy có thể kết luận cách thức này không thể thay thế mô hình Máy học.

2. Cập nhật thứ hai:

Sau buổi vấn đáp, nhóm xin trình bày cập nhật thứ hai của đề tài. Theo đó, vấn đề thứ hai là việc cần làm rõ giá trị trả về được gán cho biến ‘text_extracted’ trong đoạn mã sau:

```
“ reader = easyocr.Reader(['en','vi'])

text_extracted= reader.readtext(enhance_image_for_ocr(handled_image))”
```

Trong đoạn mã trên, ‘Reader’ là một đối tượng thuộc thư viện easyocr, đối tượng ‘Reader’ bao gồm phương thức ‘readtext’. Phương thức này sẽ trả về đầu ra ở dạng danh sách, mỗi mục tương ứng là một khung giới hạn, văn bản được phát hiện và mức độ tin cậy[1]. Như vậy, nhóm xin đính

chính lại thông số nhóm dùng là để đánh giá giai đoạn thứ ba của hệ thống là trung bình mức tin cậy. Thông số độ trung bình tin cậy được nhóm tính bằng thương số giữa tổng mức độ tin cậy của tất cả văn bản được phát hiện với số lượng văn bản được phát hiện. Tổng mức độ tin cậy của tất cả văn bản và số lượng văn bản được phát hiện được nhóm tính toán thông qua đoạn mã:

```
“ for text in text_extracted:
```

```
    sum_acc+= text[2]
```

```
    count+=1 ”
```

Sau đây là ví dụ về một kết quả trả về của phương thức ‘readtext’:

```
[([[0, 2], [480, 2], [480, 107], [0, 107]], 'HỒ CHÍ MINH', 0.6400966295081111), ([[37, 95], [449, 95], [449, 203], [37, 203]], 'MỘT NHÂN CÁCH', 0.8701970504017805), ([[116, 180], [380, 180], [380, 270], [116, 270]], 'HOÀN HẢO', 0.9770865478519624)]
```

CHƯƠNG 1: TỔNG QUAN ĐỒ ÁN

1. Các bài viết tham khảo:

1.1 Bài viết tham khảo số 1:

Title Extraction from Book Cover Images Using Histogram of Oriented Gradients and Color Information (Yen Do, Soo Hyung Kim, In Seop Na, School of Electronics & Computer Engineering, Chonnam National University Gwangju, 500-757 Korea)[2].

Hình ảnh bìa sách điển hình có thể chứa văn bản, hình ảnh, sơ đồ cũng như nền phức tạp và không đều. Ngoài ra, tính biến đổi cao của đặc điểm ký tự như độ dày, phong chữ, vị trí, nền và độ nghiêng của văn bản cũng khiến cho công việc trích xuất văn bản trở nên phức tạp hơn. Do đó, các tác giả đề xuất một phương pháp hiệu quả gồm hai bước sử dụng biểu đồ chuyển màu định hướng và thông tin màu sắc để tìm vùng tiêu đề. Đầu tiên, việc định dạng vị trí văn bản được thực hiện để tìm ra các vị trí có khả năng là tiêu đề. Cuối cùng, quá trình sàng lọc được thực hiện để tìm ra đủ các thành phần của vùng tiêu đề. Để có được kết quả tốt nhất, họ còn sử dụng các ràng buộc khác về kích thước, tỷ lệ giữa chiều dài và chiều rộng của tiêu đề. Các tác giả đã đạt được kết quả rất tốt trong việc trích xuất vùng tiêu đề từ ảnh bìa sách, chứng tỏ ưu điểm và hiệu quả của phương pháp đề xuất. Kết quả có thể ứng dụng trong việc quản lý và số hóa tủ sách, từ đó cho ra những thông tin cần thiết của cuốn sách cho người dùng.

1.2 Bài viết tham khảo số 2:

Vietnamese Text Extraction From Book Covers (Phan Thi Thanh Nga, Nguyen Thi Huyen Trang, Nguyen Van Phuc, Thai Duy Quy, Vo Phuong Binh, The Faculty of Information Technology, Dalat University, Lamdong, Vietnam, The Devsoft Company, Hochiminh City, Vietnam, The Research Management and International Cooperation Department, Dalat University, Lamdong, Vietnam)[3].

Trong bài báo này, các tác giả đã trình bày một phương pháp mới trong việc trích xuất văn bản tiếng Việt từ ảnh bìa sách được ‘scan’. Hệ thống được đề xuất chấp nhận ảnh chụp nhanh bìa sách, lọc hình ảnh đầu vào để nâng cao chất lượng, định vị các vùng có văn bản, sau đó sử dụng bộ nhận dạng ký tự quang học (OCR) để trích xuất văn bản. Bước cuối cùng là lọc văn bản được trích xuất kèm theo từ điển để có được kết quả văn bản cuối cùng. Việc thực hiện các thử nghiệm với hệ thống được đề xuất bằng bộ dữ liệu của các tác giả đã mang lại kết quả thử nghiệm rất tốt. Bằng cách triển khai VTEB từ đầu, các tác giả đã cho thấy và chỉ ra rằng dự án này có thể được coi là thử nghiệm ban đầu của hệ thống nhận dạng bìa sách. Họ đã cung cấp quy trình làm việc và cách triển khai để truy xuất văn bản rõ ràng từ hình ảnh bìa sách. Phép biến đổi Hough được triển khai để làm giảm độ lệch của hình ảnh. Đồng thời cũng sử dụng các bộ lọc khác nhau để giảm nhiễu, xóa nền và trích xuất các vùng văn bản. Imagemaker đã giúp đỡ rất nhiều trong việc làm sạch ảnh nguồn bằng các bộ lọc cơ bản. Việc sử dụng công cụ Tesseract giúp OCR văn bản từ hình ảnh đầu vào. Trong nguyên mẫu, các tác giả đã triển khai lồng lểo các bước xử lý hình ảnh, OCR và xử lý hậu kỳ hình ảnh. Tuy nhiên, mỗi bước này có thể được thay đổi độc lập để có kết quả phù hợp. Thời gian đáp ứng của thuật toán biến đổi Hough chậm đáng kể trong trường hợp chúng ta đặt giá trị lớn cho góc. Bằng cách giả định rằng hầu hết các hình ảnh bị lệch đều nhận được giá trị nhỏ hơn 15 độ, họ đã giảm độ phức tạp của thuật toán bằng cách giới hạn các góc ở $[-16, 16]$ độ và kích thước bước được đặt thành 0,2.

2. Tổng quan đồ án

Từ hai bài viết tham khảo vừa đề cập, đã giúp nhóm thực hiện tổng kết được những kiến thức cơ bản có liên quan, cũng như bước đầu tư duy về các bước thực hiện đề tài. Tuy nhiên hai bài viết trên tiếp cận bài toán với các hướng nghiên cứu đòi hỏi kỹ thuật cao, không phù hợp với hợp sinh viên có mức trình độ nhập môn (các thành viên nhóm thực hiện), sau đây nhóm thực hiện sẽ nghiên cứu, thực hiện đề tài với hướng tiếp cận khác, phù hợp với sinh viên nhập môn Máy học.

2.1 Mô tả đề tài và các ngữ cảnh ứng dụng:

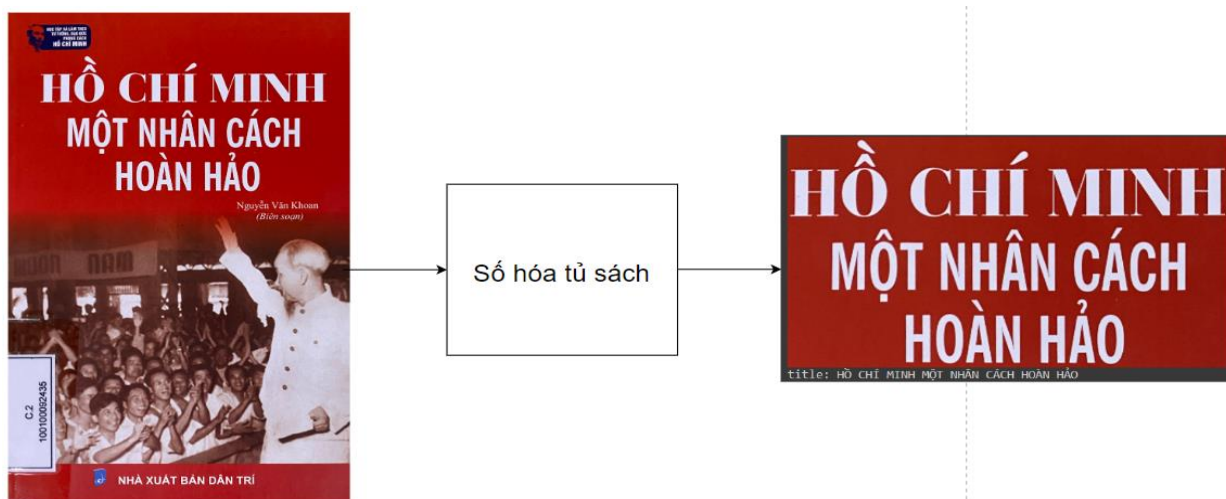
Đề tài đồ án của nhóm với mục tiêu là phát triển một hệ thống bao gồm mô hình Máy học cho phép người dùng đưa vào ảnh là trang bìa một cuốn sách, hệ thống sẽ nhận dạng được và xuất ra tên cuốn sách đó, hỗ trợ việc lập danh sách quản lý sách.

- Input: Ảnh bìa quyển sách.
- Output: Tên của quyển sách.

Ngữ cảnh ứng dụng: Sản phẩm của đề tài có thể ứng dụng cho các tổ chức, cá nhân có nhu cầu lập danh sách quản lý sách, văn bản có bìa như sách,... Ví dụ thực tế: Các tiệm sách cũ, các quán cà phê sách... có thể sở hữu hàng trăm thậm chí hàng nghìn tựa sách, từ nhiều nguồn như mua lại, được cho tặng,... Trong các trường hợp trên thường không có danh sách quản lý ngay từ ban đầu, việc

đó gây khó khăn cho việc quản lý sách, bởi lẽ việc lập danh sách lúc này sẽ rất mất thời gian do phải viết bằng tay hoặc đánh máy. Sản phẩm của đề tài có thể ứng dụng để hỗ trợ các hiệu sách cũ trong việc quản lý mua bán, các mô hình kinh doanh liên quan đến sách hay các cá nhân muốn quản lý tủ sách.

Lý do sử dụng mô hình máy học: Trên một bìa sách có thể bao gồm rất nhiều kí tự chữ viết, nhưng không phải tên sách, bao gồm: Tên nhà xuất bản, tên tác giả hoặc cụm từ chỉ để trang trí hay quảng bá như: “Best Seller”, “New Edition”,... .Không có một chương trình lập trình truyền thống cụ thể nào, thật sự có thể áp dụng cho việc phân loại các vùng văn bản trên bìa sách đâu thật sự là tên sách. Vì lý do trên, hệ thống cần áp dụng mô hình Máy học để phân loại được tên sách trong tất cả các vùng văn bản được nhận diện. Cụ thể hơn là mô hình Binary Classification để trả lời câu hỏi vùng văn bản vừa được nhận diện có phải là tên sách hay không (có hoặc không). Mô hình Binary Classification cũng chính là **nhiệm vụ chính và quan trọng nhất** mà nhóm đặt ra. Ngoài ra, để hỗ trợ cho việc phân loại các vùng văn bản cần có sự tham gia của các công cụ khác để nhận diện, khoanh vùng vùng văn bản trên ảnh bìa sách và trích xuất văn bản đó. Cụ thể, hai giai đoạn vừa nêu sẽ được hỗ trợ bởi công cụ ‘PyTesseract’- hỗ trợ việc nhận diện và khoanh vùng văn bản, công cụ ‘EasyOCR’- hỗ trợ việc trích xuất vùng văn bản. Hai công cụ trên sẽ được phối hợp với mô hình Máy học được nhóm phát triển, cho ra hệ thống có chức năng đúng như mục tiêu đề ra.



Hình 2 - Mô tả kết quả thực hiện đề tài

2.2 Mô tả bộ dữ liệu:

Tập dữ liệu được sử dụng cho việc huấn luyện mô hình Binary Classification bao gồm **5046** mẫu - tương ứng với các hàng, mỗi mẫu bao gồm các đặc trưng (feature) và mục tiêu (target) – tương ứng với các cột. Tập dữ liệu được xây dựng từ **700** ảnh bìa sách qua các bước xử lí, rút trích đặc trưng cho ra các phần tử là các đặc trưng của các vùng chứa văn bản trên ảnh bìa sách được đặt tên là: ‘left’, ‘right’, ‘length’, ‘height’, ‘size’, ‘center’ (sẽ được giải thích chi tiết trong phần sau). Từ các đặc trưng được rút trích nhóm sẽ thực hiện việc đánh nhãn cho dữ liệu bằng cách điền giá trị cho cột ‘title’ (cột tương ứng với cột mục tiêu – ‘target’) thể hiện cho việc mẫu dữ liệu được thu thập, đại diện cho vùng văn bản có phải là vùng văn bản chứa tựa sách hay không (nếu là tựa sách sẽ được đánh giá trị **1** ngược lại là **0**), lưu ý tựa sách có thể nằm trên nhiều vùng văn bản khác nhau. Tập dữ liệu thu được

cuối cùng là tập dữ liệu gồm **5046** hàng (5046 mẫu) và **7** cột theo thứ tự từ trái sang phải 6 cột đầu tiên là 6 cột đặc trưng: ‘left’, ‘right’, ‘length’, ‘height’, ‘size’, ‘center’; 1 cột cuối cùng là cột mục tiêu: ‘title’. Ảnh bìa sách mà nhóm thu thập là các ảnh thực tế được chụp từ camera, nguồn thu thập đến từ Thư viện Trường Đại học Công nghệ Thông Tin, Thư viện Trung tâm ĐHQG - TP.HCM, các hội nhóm internet,....Tập dữ liệu của nhóm sẽ được lưu trong file “final_data_set.txt”.

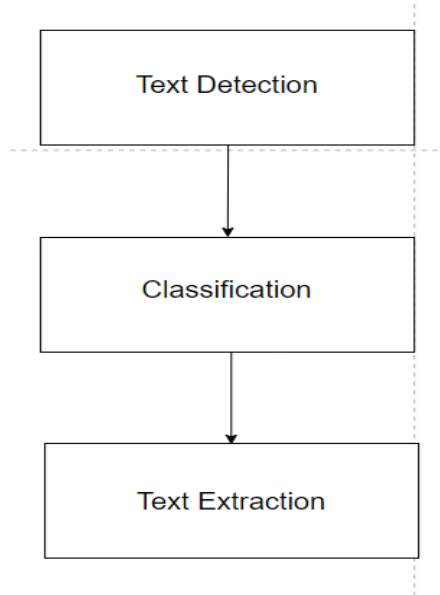
370	235	1479	255	377145	0	0
72	1598	1990	575	1144250	1	1
493	2095	1370	218	298660	1	1
0	0	2375	3251	7721125	0	0
1561	64	4	189	756	0	0
51	180	2064	980	2022720	0	0
1807	80	86	19	1634	0	0
51	70	1514	431	652534	0	0
594	86	1232	199	245168	0	0
512	221	1351	229	309379	0	0

Hình 3 - Mô tả bộ dữ liệu

2.3 Mô tả thuật toán Máy học và các công cụ hỗ trợ:

Như đã trình bày ở phần mô tả đề tài, để thực hiện được mục tiêu đặt ra, đề tài cần sự phối hợp của mô hình Máy học và các công cụ hỗ trợ để tạo thành một hệ thống hoàn chỉnh. Cụ thể, hệ thống bao gồm 3 giai đoạn được như sau: giai đoạn thứ nhất (Text Detection), giai đoạn thứ hai (Binary Classification), giai đoạn thứ ba (Text Extraction). Chức năng của các giai đoạn:

- Giai đoạn thứ nhất (Text Detection): Ảnh bìa sách đưa vào được xử lý để nhận diện ra vùng chứa văn bản. Giai đoạn này được thực hiện bằng cách áp dụng công cụ Pytesseract để phát hiện các vùng văn bản trên bìa sách, đồng thời cho biết các đặc trưng của những vùng đó.
- Giai đoạn thứ hai (Classification): Nhóm sẽ thực hiện huấn luyện mô hình để từ các vùng văn bản được nhận diện (kết quả của giai đoạn thứ nhất) phân loại đâu là vùng chứa tên sách (dựa vào các đặc trưng của vùng đó). Việc phân loại này có thể được liên tưởng đến việc trả lời câu hỏi “vùng văn bản có phải là tên của quyển sách hay không?”, câu trả lời là “có hoặc không” vùng văn bản sẽ được phân vào hai lớp: “có” hoặc “không” tương ứng với giá trị ‘1’ hoặc ‘0’ tại cột mục tiêu trong tập dữ liệu, từ những phân tích trên có thể thấy giai đoạn này sử dụng mô hình máy học “Binary Classification” để thực hiện là phù hợp nhất. Đây là giai đoạn mà nhóm sẽ tiến hành thu thập dữ liệu và tự huấn luyện mô hình sau đó cài đặt vào hệ thống.
- Giai đoạn thứ ba (Text Extraction): Sau khi đã biết được vùng văn bản nào là vùng chứa tên sách, giai đoạn này sẽ sử dụng công cụ hỗ trợ là Easy OCR để xuất văn bản từ vùng văn bản được phân loại là tựa sách, cho ra kết quả cuối cùng là tựa của quyển sách.



Hình 4 - Mô tả cơ bản quá trình hoạt động của hệ thống

2.4 Mục tiêu của đề tài:

Mục tiêu nhóm hướng đến trong đề tài lần này bao gồm hai điều quan trọng nhất:

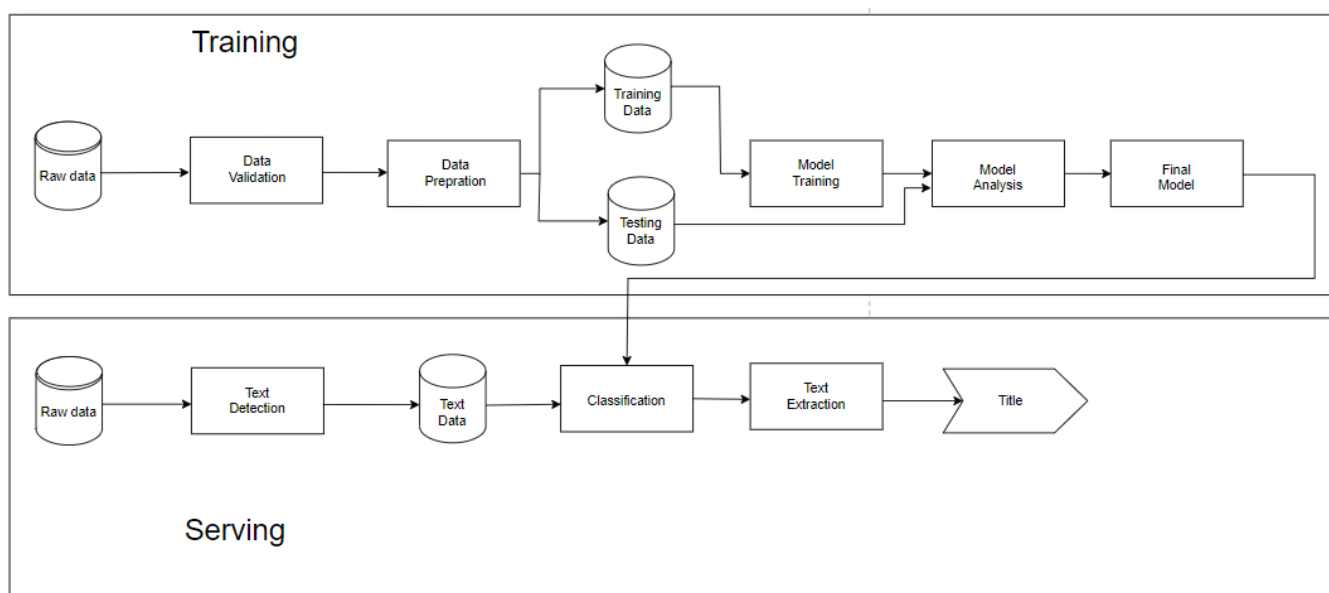
- Thực hiện thành công mô hình Binary Classification: thu thập được bộ dữ liệu, chọn ra được thuật toán phù hợp, thông số đánh giá nằm trên mức trung bình.
- Phối hợp giữa mô hình máy học và các công cụ để tạo ra hệ thống hoàn chỉnh: đạt được mô tả đồ án là đầu vào dữ liệu là ảnh của quyển sách sau đó cho ra tên sách.

Trong đồ án này, nhóm không đặt mục tiêu sẽ đem đề tài ứng dụng thực tế vì còn hạn chế nhiều về mặt kiến thức và thời gian để có thể phát triển một mô hình đủ tốt. Vì thế độ chính xác của hệ thống không phải mục tiêu mà nhóm hướng tới.

CHƯƠNG 2: CHI TIẾT VÀ HIỆN THỰC ĐỒ ÁN

1. Machine learning pipeline

Trong phần “2.3 Mô tả mô hình thuật toán máy học và các công cụ hỗ trợ” đã nêu lên những cơ bản nhất về hệ thống mà nhóm phát triển. Trong phần này sẽ đi sâu vào chi tiết những công đoạn mà nhóm đã nghiên cứu và thực hiện để cho ra kết quả cuối cùng, hình sau là Machine Learning Pipeline của nhóm, có thể xem là bản thiết kế của đề tài:



Hình 5 - Machine Learning Pipeline

1.1 Pha ‘Training’:

Trong phạm vi đề tài, đây là pha mô tả việc huấn luyện mô hình Binary Classification được sử dụng trong giai đoạn thứ hai của hệ thống, cũng là phần chiếm nhiều thời gian và công sức thực hiện nhất.

Trong đó “Raw data” là 700 ảnh bìa sách mà nhóm thu thập được, chưa qua bất cứ công đoạn xử lý nào, chỉ là ảnh vừa được chụp từ camera. Công đoạn kế tiếp là “Data Validation” đây là công đoạn xử lý dữ liệu đầu tiên, mang tính xác minh dữ liệu. Tại công đoạn “Data Validation” những phần thừa không liên quan đến ảnh bìa sách sẽ được cắt bỏ hay những ảnh mờ sẽ được loại bỏ. Công đoạn “Data Preparation” là công đoạn mà nhóm sẽ tiến hành rút trích đặc trưng và đánh nhãn cho dữ liệu. Những tấm ảnh bìa sách ban đầu, sau khi qua các công đoạn xử lý, đã được xác minh và rút trích đặc trưng của các vùng văn bản có chứa trên ảnh bìa sách, được đánh nhãn và tạo thành bộ dữ liệu phù hợp cho việc huấn luyện mô hình Máy học. Tập dữ liệu sau xử lý sẽ được chia thành 2 tập nhỏ hơn là tập “Training Data” và “Test Data” nhằm mục đích huấn luyện và đánh giá mô hình Máy học. Tới đây, mô hình Binary Classification đã sẵn sàng được huấn luyện. Các công đoạn vừa nêu trên là những công đoạn quan trọng hàng đầu của đề án, sẽ được trình bày cụ thể trong phần “2. Xây dựng bộ dữ liệu”.

Sau khi hoàn thành việc xây dựng bộ dữ liệu, tiếp tục đến với công đoạn huấn luyện để hoàn thiện mô hình Binary Classification. Nhóm sẽ tiến hành huấn luyện mô hình Binary Classification bằng cách huấn luyện cả 6 mô hình dựa trên 6 thuật toán bao gồm: Logistic Regression, Support Vector Machines, Decision Trees, Random Forest, Naive Bayes, K-Nearest Neighbor sau đó chọn ra mô hình tốt nhất dựa trên thông số F1-score để làm mô hình Binary Classification của hệ thống. Lưu ý 6 mô hình dựa trên các thuật toán: Logistic Regression, Support Vector Machines, Decision Trees, Random Forest, Naive Bayes, K-Nearest Neighbor là những mô hình Máy học đã được Scikit-learn hỗ trợ, nhóm chỉ tiến hành cài đặt và không xây dựng lại từ đầu. Các công đoạn liên quan đến huấn luyện mô hình máy học sẽ được nhóm trình bày rõ hơn trong phần “3. Huấn luyện và đánh giá mô hình Binary Classification”.

1.2 Pha ‘Serving’:

Pha ‘Serving’ là pha mô tả việc hệ thống thực hiện chức năng của mình như thế nào. Sau khi hoàn thành việc huấn luyện mô hình Binary Classification, các giai đoạn của hệ thống đã hoàn thiện, hệ thống đã sẵn sàng. ‘Raw data’ tại đây cũng là ảnh bìa sách chưa qua xử lý (không trùng lại với những ảnh đã dùng để huấn luyện mô hình Máy học). Sau đó sẽ đi đến công đoạn ‘Text Detection’ đồng thời cũng là giai đoạn thứ nhất của hệ thống xử lý để xác định các vùng có chứa văn bản, cụ thể hơn công việc đó sẽ được xử lý bằng cách áp dụng công cụ PyTesseract với câu lệnh cụ thể là:

`d=pytesseract.image_to_data(image,output_type=pytesseract.Output.DICT)`,

trong đó:

- image là ảnh bìa sách được truyền vào.
- output_type=pytesseract.Output.DICT để xác định dữ liệu đầu ra là kiểu từ điển.
- Biến ‘d’ mang thông tin của tất cả các vùng có chứa văn bản trong ảnh.

Biến ‘d’ mang giá trị kiểu từ điển lưu thông tin các vùng được nhận diện là có chứa văn bản. Tuy nhiên, nó bao gồm nhiều cấp độ văn bản như: kí tự, từ, đoạn văn bản,... Theo đó tựa sách sẽ nằm ở cấp độ là đoạn văn bản, vì thế ta cần tiếp câu lệnh: ‘if d[‘level’][i] == 2:’ để đảm bảo tránh các trường hợp dữ liệu xấu (trường hợp đã nhận diện đoạn văn bản và tiếp tục nhận diện các kí tự, các từ trong đoạn văn bản đó). Kết thúc giai đoạn này ta sẽ được kết quả là thông tin đặc trưng của những vùng văn bản được nhận diện (trong Pipeline Machine Learning là ‘Text Data’), là dữ liệu sẽ được sử dụng cho giai đoạn tiếp theo. Giai đoạn này được thực hiện trong ‘notebook’ bằng đoạn mã sau:

Bảng 1 - Hàm detect_text

```
def detect_text (image_path):
    image = cv2.imread(image_path)
    image = process_img(image)
    height, width = image.shape
    image_center = (width / 2, height / 2) #tâm ảnh
    min_distance = float('inf')
    center_box = None
    d = pytesseract.image_to_data(image, output_type=pytesseract.Output.DICT)
    box = []
    boxes_with_distances = []
    for i in range(len(d['level'])):
        if d['level'][i] == 2:
            (x, y, w, h) = (d['left'][i], d['top'][i], d['width'][i], d['height'][i])
            box.append((x, y, w, h, w*h))
```



```

box_center = (x + w / 2, y + h / 2)
distance = np.sqrt((box_center[0] - image_center[0])**2 + (box_center[1] - image_center[1])**2)
boxes_with_distances.append(((x, y, w, h), distance))

boxes_with_distances.sort(key=lambda x: x[1])
final_box = ()
# Lấy 3 hộp gần trung tâm nhất
closest_boxes = boxes_with_distances[:3]
# Lấy các tọa độ box từ closest_boxes để so sánh
closest_coords = [box[0] for box in closest_boxes]

# Thêm 1 nếu box nằm trong closest_boxes, ngược lại thêm 0 tạo feature center
final_boxes = [box + (1,) if box[:4] in closest_coords else box + (0,) for box in boxes]

return final_boxes

```

Các giá trị trả về sẽ là (x, y, w, h, w*h, 1 hoặc 0) các giá trị này sẽ tương ứng với các đặc trưng của vùng văn bản là: left, right, length, height, size, center trong bộ dữ liệu dùng để huấn luyện mô hình máy học. Các đặc trưng này sẽ được giải thích ở phần sau.

Sau công đoạn nhận diện vùng văn bản, các thông tin của vùng chứa văn bản sẽ được trích xuất và trở thành dữ liệu đầu vào cho giai đoạn thứ hai của hệ thống cũng là công đoạn ‘Classification’ trong Pipeline Machine Learning. Dựa vào những thông tin nhận được mô hình Binary Classification sẽ phân loại xem vùng văn bản có phải là tựa của quyển sách hay không. Công đoạn “Text Extract” (giai đoạn thứ 3 của hệ thống) sẽ dựa vào giá trị đầu vào là kết quả phân loại và đặc trưng về vùng văn bản để hoạt động. Cụ thể nếu mô hình Binary Classification cho ra kết quả phân loại là 1, nghĩa là vùng văn bản được phân loại là tựa sách thì văn bản trong vùng sẽ được tách ra. Tựa sách được tách ra cũng là kết quả cuối cùng của hệ thống, để có thể lấy được văn bản trên ảnh sẽ cần đến công cụ hỗ trợ EasyOCR với các câu lệnh cụ thể là:

```
“reader = easyocr.Reader(['en','vi'])
```

```
text_extracted= reader.readtext(enhance_image_for_ocr(handled_image))”, trong đó:
```

- “reader= easyocr.Reader(['en','vi'])”, là câu lệnh dùng để khởi tạo một đối tượng ‘Reader’ từ thư viện ‘easyocr’ với ngôn ngữ là tiếng Anh (‘en’) và tiếng Việt (‘vi’).
- “text_extracted= reader.readtext(enhance_image_for_ocr(handled_image))” là câu lệnh với mục đích đọc và lưu các thông tin về văn bản được đọc ra từ ảnh.

Pha ‘Serving’ có thể được mô tả trong đoạn mã Python sau:

Bảng 2 - Pha Serving trong Machine Learning Pipeline

```

target_size = (480, 270) #dài rộng

#-----GIAI ĐOẠN TEXT DETECTION-----
for i in range(1, 71):
    print('-----'+ ' book '+str(i)+' -----')
    data= detect_text('/content/img ('+str(i)+').JPG')
    image = cv2.imread('/content/img ('+str(i)+').JPG')
    img_height, img_width = image.shape[:2]
    #cv2.imshow(image)
    df = pd.DataFrame(data, columns=['left', 'top', 'width', 'height', 'size', 'center'])
    df_temp = df
#-----GIAI ĐOẠN CLASSIFICATION-----

```



```

ss_train = StandardScaler()
df = ss_train.fit_transform(df)
models[best_model_key].fit(X_train, y_train)
predictions = models[best_model_key].predict(df)

#-----GIAI ĐOẠN TEXT EXTRACTION-----
title = ""
for index, row in df_temp[predictions == 1].iterrows():
    # Các thông số để cắt ảnh
    left, top, width, height = row['left'], row['top'], row['width'], row['height']
    left = max(row['left'] - 10, 0) # Giảm left nhưng không để nó âm
    top = max(row['top'] - 10, 0)
    width = min(row['width'] + 20, img_width - left)
    height = min(row['height'] + 20, img_height - top)
    # Cắt ảnh
    handled_image = image[top:top+height, left:left+width]
    handled_image = cv2.resize(handled_image, target_size)
    # Đọc văn bản từ ảnh
    reader = easyocr.Reader(['en', 'vi'])
    text_extracted = reader.readtext(enhance_image_for_ocr(handled_image))

    for text in text_extracted:
        if text[2] > 0.1:
            title += text[1] + ' '
    # Hiển thị ảnh được dự đoán là tựa sách
    cv2.imshow(handled_image)
    print('title: ' + title)

```

Lưu ý: ảnh được truyền vào phải được đặt tên theo đúng cú pháp “img ([i]).JPG” trong đó ‘i’ là biến đếm thể hiện cho ảnh thứ i. Ngoài 3 giai đoạn là: Text Detection, Classification, Text Extraction là 3 nội dung chính, đoạn mã trên còn bao gồm một số bước phụ như tạo “data frame”, cắt ảnh,...

Để hỗ trợ cho việc phát hiện vùng văn bản và tách văn bản ra khỏi ảnh, cần một số bước tinh chỉnh ảnh trong đó hàm “process_img” được dùng cho việc tinh chỉnh ảnh cho giai đoạn “*Text Dectcion*”, hàm “enhance_image_for_ocr” được dùng cho giai đoạn “*Text Extraction*”. Đoạn mã cho hai hàm trên có nội dung như sau:

Bảng 3 - Hàm “process_img”

```

def process_img (image):
    image= cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image= cv2.bitwise_not(image)
    kernel = np.ones((2, 2), np.uint8)
    image = cv2.dilate(image, kernel, iterations=1)
    image = cv2.erode(image, kernel, iterations=1)
    return image

```

Bảng 4 - Hàm “enhance_image_for_ocr”

```

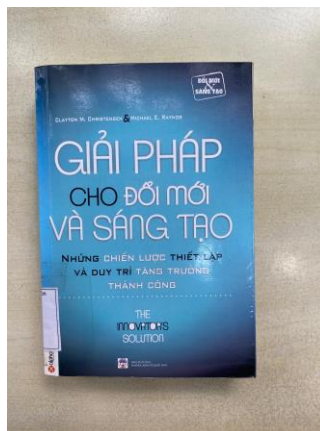
def enhance_image_for_ocr(image):
    # Chuyển đổi sang grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Áp dụng denoising
    denoised = cv2.medianBlur(gray, 1)
    # Tăng độ tương phản
    alpha = 1.5 # Hệ số tương phản
    beta = 0 # Độ sáng
    contrast = cv2.convertScaleAbs(denoised, alpha=alpha, beta=beta)
    return contrast

```

2. Xây dựng bộ dữ liệu

2.1 Thu thập dữ liệu ban đầu (Raw Data):

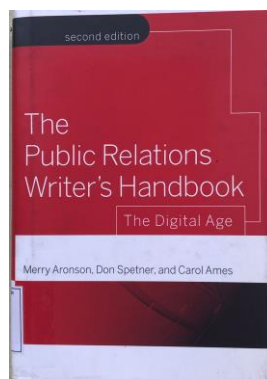
Dữ liệu ban đầu được thu thập bằng cách chụp ảnh bìa sách từ thực tế. Nguồn dữ liệu được lấy từ sách có trong các thư viện như Thư viện Trường Đại học Công nghệ Thông tin ĐHQG - TP.HCM, Thư viện Trung tâm ĐHQG - TP.HCM, Thư viện Trung tâm ĐHQG - TP.HCM chi nhánh KTX khu B, là chủ yếu nhất. Ngoài ra còn đến từ nguồn sách cá nhân, từ các nhà sách và các nguồn internet khác nhưng đảm bảo ảnh được chụp từ camera. Ngôn ngữ được sử dụng trên bìa sách có thể là tiếng Việt hoặc tiếng Anh, trong đó tiếng Việt chiếm đa số.



Hình 6 - Ví dụ về một mẫu dữ liệu ban đầu

2.2 Kiểm định dữ liệu ban đầu (Data Validation):

Dữ liệu ban đầu được thu thập là các ảnh bìa sách, các ảnh này cần phải được kiểm định lại. Trong các ảnh được thu thập có thể bao gồm nhiều phần không liên quan đến bìa sách ví dụ như: chụp dính tay người chụp, mặt bàn,... những phần này cần được cắt bỏ. Các ảnh quá mờ hoặc do điều kiện bên ngoài khác như chói sáng dẫn đến không thấy rõ được tựa sách,... các ảnh này cần được loại bỏ. Nhóm đã tiến hành chụp ảnh thu thập dữ liệu **hơn 1000** tựa sách khác nhau, tuy nhiên tổng kết thu thập được **770** ảnh đạt chất lượng, **700** ảnh sẽ được rút trích đặc trưng để tạo dữ liệu cho việc huấn luyện và kiểm tra mô hình Binary Classification, **70** ảnh còn lại dùng để kiểm tra thống kê số liệu cho toàn bộ hệ thống.



Hình 7 - Ví dụ về một mẫu dữ liệu ban đầu đã qua kiểm định

2.3 Rút trích đặc trưng và đánh nhãn dữ liệu (Data Preparation):

2.3.1 Rút trích đặc trưng

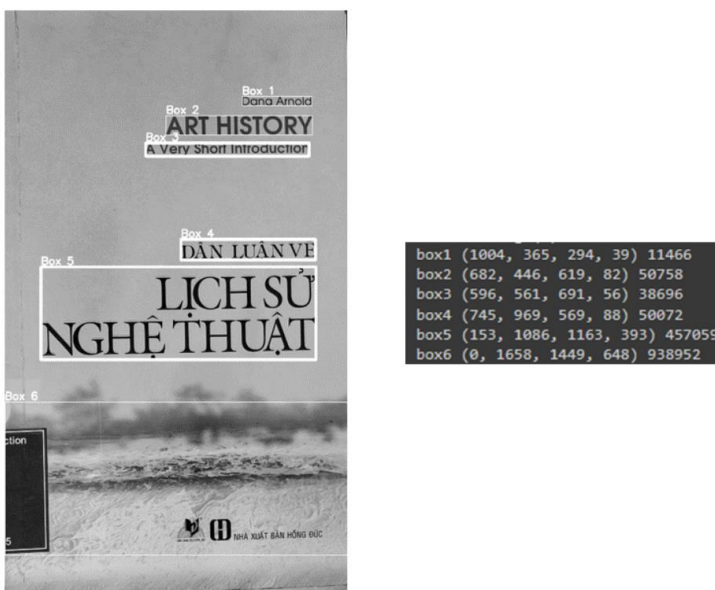
Dữ liệu là các ảnh sau khi được kiểm định sẽ tiếp tục được rút trích đặc trưng thông qua công cụ PyTesseract và các câu lệnh tương tự như giai đoạn “*Text Detection*” đã được trình bày ở trên. Trước khi được rút trích dữ liệu ảnh bìa sách sẽ được lược bỏ màu (màu sắc trong cách tiếp cận của đề tài không quan trọng cho việc phân loại tựa sách), tinh chỉnh thêm một số yếu tố khác nhằm phục vụ cho việc trích xuất đặc trưng được hiệu quả. PyTesseract cung cấp các thông tin về các đặc trưng của vùng chứa văn bản là: 'left', 'top', 'width', 'height', đó cũng chính là những đặc trưng cơ bản nhất đã được nhắc đến ở các phần trên. Các đặc trưng đó cũng là những thông số giới hạn vùng văn bản là vùng hình chữ nhật chứa văn bản ở bên trong. Theo đó:

- 'left' là khoảng cách từ điểm ở góc trên bên trái của vùng văn bản đến cạnh bên trái của ảnh.
- 'top' là khoảng cách từ điểm ở góc trên bên trái của vùng văn bản đến cạnh bên trên của ảnh.
- 'width' là chiều rộng (có thể xem là chiều dài tùy theo quy ước) của vùng văn bản.
- 'height' là chiều cao (có thể xem là chiều rộng tùy theo quy ước) của vùng văn bản.

Từ các đặc trưng cơ bản nhất trên, nhóm tiếp tục tạo ra đặc trưng mới (Feature Engineering) là đặc trưng 'size' thể hiện diện tích vùng chứa văn bản, từ hai đặc trưng cơ bản là 'width' và 'height' bằng cách nhân hai đặc trưng này lại với nhau. Đây là một đặc trưng mang ý nghĩa quan trọng, bởi tựa sách thường là vùng mang diện tích lớn.

Ngoài 5 đặc trưng trên, nhóm tiếp tục rút trích đặc trưng 'center' mà không từ đặc trưng nào có sẵn (Feature Generation), đặc trưng này mang ý nghĩa về vị trí của vùng văn bản có nằm gần vị trí trung tâm của ảnh hay không (tựa sách thường sẽ nằm ở vị trí trung tâm sách, có thể lệch trên hoặc lệch dưới). Đặc trưng này thể hiện vùng văn bản có là 1 trong 3 vùng văn bản gần trung tâm ảnh nhất hay không (phải thì đặc trưng mang giá trị **1**, không thì đặc trưng mang giá trị **0**). Mức độ gần trung tâm của vùng văn bản, được đánh giá dựa trên khoảng cách từ tâm vùng chứa văn bản đến tâm ảnh. Tuy vậy, đặc trưng này sẽ được đánh giá lại bằng năng lực của con người trong giai đoạn đánh nhãn dữ liệu.

Sau bước rút trích, các đặc trưng thu được là: 'left', 'top', 'width', 'height', 'size' và 'center', cũng sẽ lần lượt là các cột đặc trưng tương ứng trong tập dữ liệu. Các đặc trưng này cũng sẽ là các đặc trưng theo nhóm là đủ để đánh giá vùng văn bản được nhận diện trên ảnh bìa sách có phải là tựa sách hay không. Hình sau là ví dụ về một bìa sách mà vùng chứa văn bản đã được khoanh vùng, vùng được tô đậm hơn thể hiện vùng văn bản là 1 trong 3 vùng văn bản gần trung tâm ảnh nhất. Ảnh bên phải thể hiện các đặc trưng của các vùng văn bản được nhận diện bên trái, theo thứ tự lần lượt là: 'left', 'top', 'width', 'height', 'size'.



Hình 8 - Ví dụ về quá trình khoanh vùng văn bản và trích xuất dữ liệu

Đoạn mã sau được dùng để nhận diện vùng văn bản và trích xuất đặc trưng từ vùng văn bản ấy, cho ra kết quả như hình 7:

Bảng 5 - Hàm khoanh vùng và trích xuất đặc trưng từ vùng văn bản

```
def box_text (image_path, d_):
    global t
    image = cv2.imread(image_path)
    image = process_img(image)
    height, width = image.shape
    image_center = (width / 2, height / 2)
    min_distance = float('inf')
    center_box = None
    d = pytesseract.image_to_data(image, output_type=pytesseract.Output.DICT)
    box = []
    boxes_with_distances = []
    c = 0
    n_boxes = len(d['level'])
    for i in range(n_boxes):
        if d['level'][i] == 2:
            (x, y, w, h) = (d['left'][i], d['top'][i], d['width'][i], d['height'][i])
            #cv2_imshow(image[y:y+h, x:x+w])
            #print(pytesseract.image_to_string(image[y:y+h, x:x+w], lang='vie'))
            box.append((x, y, w, h))
            text = f'Box {c+1}'
            c+=1
            t+=1
            box_center = (x + w / 2, y + h / 2)
            distance = np.sqrt((box_center[0] - image_center[0])**2 + (box_center[1] - image_center[1])**2)
            boxes_with_distances.append(((x, y, w, h), distance))
            cv2.putText(image, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.5, (255, 255, 255), 5)
            cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
            fo.write(str(t)+' '+str(d_)+'\n')
            fo.write(('box'+ str(c)).ljust(8, ' ')+str(x).ljust(8, ' ')+str(y).ljust(8, ' ')+str(w).ljust(8, ' '))
```

```

+str(h).ljust(8, ' ')+str(w*h).ljust(8, ' ')+ '0'.ljust(8, ' ')+ '0'.ljust(8, ' ')+ '\n')
print ('box'+ str(c), ((x, y, w, h)), w*h)

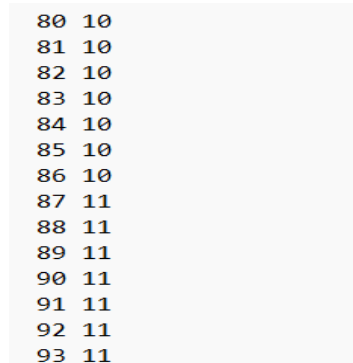
boxes_with_distances.sort(key=lambda x: x[1])
# Lấy 3 hộp gần trung tâm nhất
closest_boxes = boxes_with_distances[:3]

for box in closest_boxes:
    x, y, w, h = box[0]
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 12) # Sử dụng màu đỏ để phân biệt
cv2.imshow(image)

```

Như đã nêu ở trên, ảnh bìa sách cần được tinh chỉnh để phục vụ cho việc khoanh vùng và trích xuất đặc trưng được hiệu quả hơn, việc đó được thực hiện thông qua hàm ‘process_img’ được đề cập trong *Bảng 3 - Hàm "process_img"*.

Sau khi rút trích đặc trưng, thông tin về các đặc trưng sẽ được ghi vào file “final_data_set.txt” để tiến hành bước đánh nhãn dữ liệu. Ngoài ra trong quá trình rút trích đặc trưng còn sinh ra file “map_final.txt”- đây là file được dùng để đánh dấu mẫu dữ liệu trong tập dữ liệu được lấy ra từ ảnh nào trong 700 ảnh được chọn để rút trích đặc trưng. Trong file “map_final.txt” cột bên trái là thứ tự của mẫu dữ liệu (thứ tự hàng) trong tập dữ liệu, bên phải là thứ tự của ảnh mà từ đó mẫu dữ liệu bên trái được lấy ra. File “map_final.txt” giúp ta có thể biết được mẫu dữ liệu tại bất cứ hàng nào trong tập dữ liệu được trích xuất từ ảnh nào trong 700 ảnh được chọn để rút trích đặc trưng.



```

80 10
81 10
82 10
83 10
84 10
85 10
86 10
87 11
88 11
89 11
90 11
91 11
92 11
93 11

```

Hình 9 - file “map_final.txt”

2.3.2 Đánh nhãn dữ liệu

Sau khi đã có các đặc trưng về vùng văn bản được nhận diện, công việc kế tiếp là đánh nhãn cho các mẫu dữ liệu đó (mỗi mẫu đại diện cho 1 vùng văn bản). Việc đánh nhãn tương ứng với việc gán giá trị ‘0’ hoặc ‘1’ cho cột ‘title’. Nhãn dữ liệu sẽ được đánh giá dựa trên năng lực của con người, nếu vùng văn bản được nhận diện là tựa sách sẽ gán giá trị cho cột ‘title’ là **1** ngược lại là **0**. Ngoài việc đánh nhãn cho dữ liệu đồng thời cũng tiến hành việc đánh nhãn cho đặc trưng ‘center’, sở dĩ cần làm vậy bởi vì có một số trường hợp vùng văn bản được đánh dấu là ‘center’ không phù hợp như: không phải vùng chứa văn bản như mong muốn (một số trường hợp vùng được nhận diện là chứa văn bản lại là toàn bộ bìa sách); sách có ít hơn bằng 3 vùng văn bản, thì cả ba vùng đều là 3 vùng văn bản gần tâm nhất, khi đó vùng nào là trung tâm cần được xét lại. Sau công đoạn đánh nhãn dữ liệu sẽ thu được tập dữ liệu giống như mô tả ở phần “2.2 Mô tả bộ dữ liệu”. Dữ liệu sau đó sẽ được chia theo tỉ lệ **75%** cho việc huấn luyện mô hình Binary Classification và **25%** cho việc kiểm tra.

370	235	1479	255	377145	0	0
72	1598	1990	575	1144250	1	1
493	2095	1370	218	298660	1	1
0	0	2375	3251	7721125	0	0
1561	64	4	189	756	0	0
51	180	2064	980	2022720	0	0
1807	80	86	19	1634	0	0
51	70	1514	431	652534	0	0
594	86	1232	199	245168	0	0
512	221	1351	229	309379	0	0

Hình 10 - Dữ liệu thu được sau đánh nhãn

3. Huấn luyện và đánh giá mô hình Binary Classification

3.1 Các thuật toán được dùng để huấn luyện mô hình Binary Classification

Để có được mô hình Binary Classification tốt nhất, nhóm đã thực hiện huấn luyện 6 mô hình Binary Classification dựa trên 6 thuật toán khác nhau trên tập dữ liệu có được để chọn ra mô hình tốt nhất cho hệ thống. Cụ thể 6 thuật toán được áp dụng là:

- Logistic regression
- Support Vector Machine
- Decision Tree
- Random Forest
- Naïve Bayes
- K-nearest Neighbor

Các mô hình Binary Classification này đều đã được tích hợp sẵn trong thư viện 'sklearn' nhóm sẽ tiến hành sử dụng như sử dụng các đối tượng bình thường khác.

3.2 Huấn luyện mô hình Binary Classification

Tiến hành quá trình huấn luyện mô hình Binary Classification, như đã trình bày ở các mô hình Máy học áp dụng có thuật toán như: Logistic regression, Support Vector Machine, Decision Tree, Random Forest, Naïve Bayes, K-nearest Neighbor sẽ được gọi như các đối tượng trong thư viện 'sklearn' và được huấn luyện với bộ dữ liệu có sẵn. Các đoạn mã quan trọng nhất để khai báo đối tượng và huấn luyện các mô hình được trình bày như sau:

Bảng 6 - Khai báo các đối tượng mô hình Máy học

```
models = {}

# Logistic Regression
from sklearn.linear_model import LogisticRegression
models['Logistic Regression'] = LogisticRegression()

# Support Vector Machines
from sklearn.svm import LinearSVC
models['Support Vector Machines'] = LinearSVC()

# Decision Trees
from sklearn.tree import DecisionTreeClassifier
models['Decision Trees'] = DecisionTreeClassifier()
```

```
# Random Forest
from sklearn.ensemble import RandomForestClassifier
models['Random Forest'] = RandomForestClassifier()

# Naive Bayes
from sklearn.naive_bayes import GaussianNB
models['Naive Bayes'] = GaussianNB()

# K-Nearest Neighbors
from sklearn.neighbors import KNeighborsClassifier
models['K-Nearest Neighbor'] = KNeighborsClassifier()
Khai báo các đối tượng mô hình Máy học phổ biến để cho bài toán Binary Classification
```

Bảng 7 - Tiến hành huấn luyện mô hình Binary Classification

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

accuracy, precision, recall, f1_s = {}, {}, {}, {}

for key in models.keys():

    # Fit the classifier
    models[key].fit(X_train, y_train)

    # Make predictions
    predictions = models[key].predict(X_test)

    # Calculate metrics
    accuracy[key] = accuracy_score(predictions, y_test)
    precision[key] = precision_score(predictions, y_test)
    recall[key] = recall_score(predictions, y_test)
    f1_s[key] = f1_score(predictions, y_test)
    best_model_key = max(f1_s, key=f1_s.get)
    print('Best model is: '+best_model_key)
```

X_train và y_train là tập dữ liệu lần lượt tương ứng với cột đặc trưng và mục tiêu trong 75% tập dữ liệu dùng để huấn luyện mô hình. Tất cả mô hình Máy học được khai báo ở trên đều được huấn luyện với cùng 1 tập dữ liệu. Các thông số Accuracy, Precision, Recall, F1-score sẽ được thu thập để làm dữ kiện đánh giá mô hình, trong đó thông số F1 score được dùng để chọn ra mô hình tốt nhất.

3.3 Đánh giá mô hình Binary Classification

Tập dữ liệu để huấn luyện mô hình là tập dữ liệu lệch (các mẫu được phân vào lớp ‘0’ nhiều hơn so với lớp ‘1’) nên việc đánh giá mô hình bằng thông số độ chính xác (Accuracy) có thể không phản ánh đầy đủ khả năng của mô hình. Mô hình sẽ được đánh giá dựa trên 4 thông số là: Accuracy, Precision, Recall, F1-score. Theo đó:

- **Accuracy** là thông số thể hiện tỉ lệ giữa số lượng mẫu dữ liệu được phân lớp **đúng** trên tất cả dữ liệu được phân lớp.
- **Precision** là thông số thể hiện tỉ lệ giữa số lượng mẫu dữ liệu được phân lớp **đúng** cho lớp **1** trên tất cả các mẫu được phân lớp cho lớp **1**.
- **Recall** là thông số thể hiện tỉ lệ mẫu dữ liệu được phân lớp **đúng** cho lớp **1** trên tất cả các mẫu **thật sự** thuộc lớp **1**.

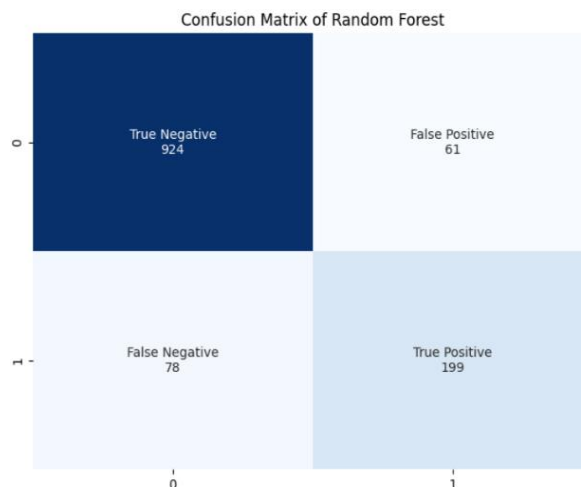
- **F1-score** là thông số kết hợp giữa Precision và Recall. F1_score cao thể hiện mô hình có độ chính xác cao, làm tốt trong cả việc tránh dự đoán sai và tránh bỏ sót các trường hợp quan trọng. Đây cũng là thông số được dùng để chọn ra mô hình tốt nhất trong đề tài.

Hình sau là kết quả các thông số Accuracy, Precision, Recall, F1-score thu được từ 6 mô hình Binary Classification đã huấn luyện.

	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.851030	0.588448	0.687764	0.634241
Support Vector Machines	0.851030	0.592058	0.686192	0.635659
Decision Trees	0.851030	0.707581	0.646865	0.675862
Random Forest	0.889857	0.732852	0.757463	0.744954
Naive Bayes	0.830428	0.837545	0.578554	0.684366
K-Nearest Neighbor	0.883518	0.765343	0.721088	0.742557

Hình 11 - Thông số các mô hình Binary Classification

Mô hình Binary Classification áp dụng thuật toán Random Forest cho tỉ lệ F1-score cao hơn cả, sẽ được chọn là mô hình Binary Classification cho hệ thống.



Hình 12 - "Confusion matrix" của mô hình Binary Classification

CHƯƠNG 3: ĐÁNH GIÁ HỆ THỐNG VÀ KẾT LUẬN

1. Đánh giá hệ thống

Để đánh giá tổng thể đề tài một cách chi tiết nhất, nhóm thực hiện đánh giá từng giai đoạn: Text Detection, Classification, Text Extraction và cuối cùng là đánh giá toàn bộ hệ thống.

1.1 Đánh giá giai đoạn thứ nhất (Text Detection)

Giai đoạn Text Detection có chức năng chính là nhận diện vùng có văn bản. Tiêu chuẩn đánh giá của nhóm chỉ chọn những ảnh bìa sách khi qua giai đoạn này mà tựa sách nằm trong các vùng văn bản được nhận diện. Dựa vào tiêu chí trên, nhóm đưa ra phương pháp đánh giá là tính tỉ lệ giữa

các ảnh bìa sách đạt tiêu chuẩn trên tổng các ảnh bìa sách dùng để kiểm tra. Dưới đây ảnh ví dụ về 2 bìa sách đạt chuẩn đánh giá của nhóm (bên trái) và không đạt chuẩn đánh giá của nhóm (bên phải).



Hình 13 - Ví dụ về cách đánh giá giai đoạn Text Detection

Nhóm thực hiện đánh giá giai đoạn Text Detection trên tập hợp **100** ảnh bìa sách, thu được **66** ảnh đạt chuẩn. Qua đánh giá có thể kết luận giai đoạn Text Detection có hiệu suất là **66%**.

1.2 Đánh giá giai đoạn thứ hai (Classification)

Giai đoạn Text Detection với chức năng nhận diện vùng văn bản, giai đoạn Classification (giai đoạn thứ hai) có chức năng phân loại vùng văn bản có phải tựa sách hay không. Kết hợp 2 giai đoạn trên sẽ thu được kết quả là vùng chứa văn bản là tựa của sách (ảnh đã được cắt chỉ còn lại vùng văn bản). Khác với việc đánh giá mô hình Binary Classification là đánh giá độ chính xác của việc phân loại vùng có văn bản có phải tựa sách hay không (đánh giá theo từng mẫu). Việc đánh giá giai đoạn 2 này là đánh giá xem từ ảnh bìa sách có vùng văn bản nào được phân loại là tựa sách hay không (không có vùng nào được phân loại là tựa sách sẽ không dẫn đến giai đoạn thứ ba). Tiêu chuẩn của nhóm dùng cho việc đánh giá giai đoạn này là mỗi ảnh bìa sách phải có ít nhất một vùng được nhận diện là tựa sách. Tiêu chí đánh giá cho giai đoạn này sẽ là tỉ lệ số ảnh đạt chuẩn trên tổng số ảnh dùng để kiểm tra. Tiến hành kiểm tra trên mẫu gồm **70** ảnh (tập ảnh này là tập ảnh đã đạt tiêu chuẩn tại giai đoạn thứ nhất) thu được **57** ảnh đạt tiêu chuẩn đề ra, hiệu suất của giai đoạn này đạt khoảng **81%**

1.3 Đánh giá giai đoạn thứ ba (Text Extraction)

Giai đoạn Text Extraction có chức năng là xuất ra văn bản có trên ảnh được đưa vào. Công cụ EasyOCR ngoài cung cấp chức xuất ra văn bản còn cung cấp chức năng cho biết mức tin cậy của văn bản vừa được xuất ra. Dựa vào chức năng này nhóm đưa ra tiêu chí đánh giá cho giai đoạn Text Extraction là trung bình mức tin cậy của các vùng văn bản được xuất ra (được tính toán tại đoạn mã tại “Bảng 2 - Pha Serving trong Machine Learning Pipeline” thông qua hai biến ‘sum_acc’ và ‘count’). Qua đánh giá trên tập **70** ảnh (tập ảnh này là tập ảnh đã đạt tiêu chuẩn tại giai đoạn thứ nhất) thu trung bình mức tin cậy của giai đoạn “Text Extraction” thu được khoảng **44%**.

Accuracy of text extraction: 0.43643481599984263

Hình 14 - Kết quả đánh giá giai đoạn Text Extraction

1.4 Đánh giá hệ thống

Mục tiêu cuối cùng của hệ thống là trích xuất được tựa của sách dựa trên ảnh bìa sách. Dựa vào mục tiêu đó nhóm sẽ đưa ra tiêu chí đánh giá là tỉ lệ giữa số lượng sách được trích xuất đạt tiêu chuẩn chia cho tổng số lượng sách được trích xuất. Tiêu chuẩn đánh giá của nhóm sẽ được chia thành 2 tiêu chuẩn là đánh giá ‘mềm’ và đánh giá ‘cứng’, các ảnh bìa sách đạt tiêu chuẩn đánh giá ‘mềm’ là những ảnh cho tựa được trích xuất ra không giống hoàn toàn so với tựa sách thực tế nhưng người đọc có thể hiểu được, các ảnh bìa sách đạt tiêu chuẩn đánh giá ‘cứng’ phải giống tuyệt đối với tựa sách thực tế. Hình bên dưới là các ví dụ về các bìa sách đạt tiêu chuẩn đánh giá ‘cứng’ (bên trái), đạt tiêu chuẩn đánh giá ‘mềm’ (ở giữa), và không đạt (bên phải).



Hình 15 - Minh họa các tiêu chuẩn đánh giá hệ thống

Tiến hành đánh giá hệ thống trên tập **70** ảnh (tập ảnh này là tập ảnh đã đạt tiêu chuẩn tại giai đoạn thứ nhất) kết quả thu được số lượng ảnh bìa sách đạt tiêu chuẩn đánh giá ‘mềm’ là **24**, số lượng ảnh bìa sách đạt tiêu chuẩn đánh giá ‘cứng’ là **5**. Tiêu chuẩn đánh giá ‘cứng’ sẽ là tiêu chuẩn đánh giá được sử dụng để kết luận hiệu suất của hệ thống, từ đó kết luận được độ chính xác của hệ thống là khoảng **7%**.

2. Hướng phát triển tiếp theo và kết luận đề tài hiện tại

2.1 Hướng phát triển tiếp theo

Sau quá trình làm việc, nhóm thực hiện đã phát triển được hệ thống đạt được những yêu cầu đề ra từ đầu, là từ ảnh bìa sách (ngõ vào) trích xuất ra được tựa quyển sách (ngõ ra), tuy độ chính xác của hệ thống là không cao nhưng đó không phải là mục tiêu mà nhóm đặt ra từ đầu. Điểm yếu của hệ thống đến từ giai đoạn thứ nhất và giai đoạn thứ ba, hai giai đoạn này chịu ảnh hưởng lớn bởi chất lượng ảnh bìa sách. Ảnh được nhóm thu thập là những ảnh sách thực thể được chụp từ camera, chất lượng ảnh chịu tác động bởi nhiều yếu tố như: góc chụp, ánh sáng, độ sắc nét,... Các yếu tố trên bìa sách như: bìa sách bị nhào, bị bẩn,.. cũng tác động đáng kể. Ngoài ra ‘font’ chữ, cách thiết kế trang bìa sách cũng tác động rất lớn. Từ các yếu tố trên có thể thấy chất lượng ảnh bìa sách bị chi phối bởi rất nhiều yếu tố, ảnh bìa sách là một loại ảnh phức tạp, việc tinh chỉnh ảnh cơ bản là không đủ để khắc phục có hiệu quả các yếu tố trên. Để khắc phục những yếu tố trên có hiệu quả, nhóm thực hiện đề xuất nên thêm vào các bước xử lý ảnh chuyên biệt trước các giai đoạn thứ nhất và thứ ba, đây sẽ là những hướng phát triển tiếp theo cho đề tài trong tương lai.

2.2 Kết luận đề tài hiện tại

Đề tài hiện tại còn nhiều điểm yếu và chưa đủ khả năng để áp dụng vào thực tế, tuy nhiên đó không phải mục tiêu mà nhóm thực hiện đề ra. Sự hạn chế về kiến thức và thời gian là rào cản để nhóm có thể phát triển một mô hình đủ khả năng ứng dụng. Các hạn chế của của hệ thống đã được nhóm đánh giá và phân tích kỹ lưỡng, tạo nền tảng cho những hướng phát triển tiếp theo.

Dựa vào những mục tiêu đã đề ra tại phần “2.4 Mục tiêu của đề tài”, đến đây có thể kết luận nhóm đã hoàn thành được những mục tiêu đề ra: phát triển được một hệ thống hoàn chỉnh cho phép trích xuất từ ảnh bìa sách (tại ngõ vào) cho ra tên sách (tại ngõ ra); phát triển được mô hình Máy học Binary Classification có khả năng dựa vào các đặc trưng của vùng văn bản trên bìa sách phân loại ra vùng văn bản có phải là tên của sách hay không.

CHƯƠNG 4: CÁC NGUỒN THAM KHẢO

- [1] “Jaiedai/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, cyrillic and etc., GitHub” [JaiedAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc. \(github.com\)](https://github.com/Jaiedai/EasyOCR) (Truy cập: 25 tháng một 2024).
- [2] “Title Extraction from Book Cover Images Using Histogram of Oriented Gradients and Color Information” https://www.researchgate.net/publication/271130671_Title_Extraction_from_Book_Cover_Images_Using_Histogram_of_Oriented_Gradients_and_Color_Information (Truy cập: 18 tháng một 2024).
- [3] “Vietnamese text extraction from book covers – researchgate” https://www.researchgate.net/publication/339359700_VIETNAMESE_TEXT_EXTRACTION_FROM_BOOK_COVERS (Truy cập: 18 tháng một 2024).