

ĐẠI HỌC QUỐC GIA TP.HCM  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



NGÀNH KHOA HỌC MÁY TÍNH

MÔN HỌC: MÁY HỌC

HỌC KỲ II (2020-2021)

---

ĐỒ ÁN  
SỐ HÓA TỦ SÁCH

---

*Sinh viên 1:*

Nguyễn Lâm Thảo Vy  
MSSV: 19522547

*Sinh Viên 2:*

Nguyễn Thị Thúy An  
MSSV: 19521183

*Sinh Viên 3:*

Huỳnh Đỗ Tấn Thành  
MSSV: 19522227

*Giảng viên:*

PGS.TS. Lê Dinh Duy  
ThS. Phạm Nguyễn Trường An

Thành phố Hồ Chí Minh, tháng 07 năm 2021

# Contents

<b>Giới thiệu đồ án</b>	<b>3</b>
<b>Chương 1: Tổng quan</b>	<b>6</b>
I Mô tả bài toán . . . . .	6
1 Ngữ cảnh ứng dụng . . . . .	6
2 Input và output . . . . .	6
3 Lời giải cho bài toán . . . . .	7
4 Các models mà nhóm sử dụng để giải quyết bài toán . . . . .	7
II Mô tả dữ liệu thu thập . . . . .	8
1 Ảnh input . . . . .	8
2 Data dành để train model YOLOv5 . . . . .	8
3 Data dành để train model VietOCR . . . . .	9
<b>Chương 2: Các nghiên cứu trước</b>	<b>11</b>
I Mô hình YOLO cho Object Detection . . . . .	11
1 Giới thiệu về YOLO . . . . .	11
2 Mô hình YOLO . . . . .	11
II CRAFT text detector cho Text localization . . . . .	15
1 Giới thiệu CRAFT text detector . . . . .	15
2 Kiến trúc network . . . . .	15
III Mô hình VietOCR cho Text recognition . . . . .	17
1 Giới thiệu mô hình VietOCR . . . . .	17
2 Kiến trúc network . . . . .	17
<b>Chương 3: Xây dựng bộ dữ liệu</b>	<b>20</b>
I Tiêu chí xây dựng bộ dữ liệu . . . . .	20
1 Ảnh input . . . . .	20
2 Data dành để train model VietOCR . . . . .	20
II Các mẫu dữ liệu khó . . . . .	21
1 Các mẫu khó đối với model YOLOv5 . . . . .	21
2 Các mẫu khó đối với model VietOCR . . . . .	22
<b>Chương 4: Training và đánh giá model</b>	<b>23</b>
I Preprocessing . . . . .	23
II Object detection . . . . .	24
1 Chuẩn bị training và testing data . . . . .	24
2 Tổng kết quá trình training . . . . .	25

3	Dánh giá kết quả . . . . .	25
III	Text localization . . . . .	28
IV	Text recognition . . . . .	35
1	Training và testing data . . . . .	35
2	Quá trình training . . . . .	35
3	Dánh giá . . . . .	35
V	Đánh giá chung . . . . .	36
VI	Nhận xét . . . . .	43
<b>Chương 5: Ứng dụng và hướng phát triển</b>		<b>44</b>
I	Ứng dụng . . . . .	44
II	Hướng phát triển . . . . .	44

## Giới thiệu đồ án

Sách - được định nghĩa như một tập hợp các thông tin, dữ liệu được chuyển hóa thành lời văn và được lưu trữ bằng từ ngữ trên trang giấy. Sách được dùng với mục đích lưu trữ và truyền bá thông tin và tri thức cho con người. Cho đến ngày nay, nhiều loại hình của sách đã được tạo ra nhằm phục vụ nhu cầu của con người trong nhiều trường hợp khác nhau. Sách (book) là thuật ngữ được mở rộng từ "scroll" (cuộn giấy). Sau đó, sách có nhiều biến thể khác nhau ở các nền văn minh khác nhau, từ những mảng đất sét, thẻ tre, thanh gỗ,... và chạm đến mốc phổ biến nhất là sách giấy như ngày nay, từ viết tay cho đến in ấn. Công nghệ ngày càng phát triển nhiều loại hình sách còn tân tiến hơn xuất hiện như audiobook (đầu những năm 1950), e-book (cuối những năm 1990),...

Thời thế phát triển, sách đã là khái niệm không chỉ là những trang giấy mà đã chuyển thành âm thanh hay những con số nằm trên máy tính. Tuy nhiên sách giấy vẫn chiếm ngôi vị độc tôn mà không có bất cứ loại sách nào có thể chiếm giữ. Với lịch sử lâu đời cùng sự phổ biến ngấm vào máu của con người, sách giấy được phát minh với mục đích phục vụ con người cũng như sự tiện lợi nhất định. Việc đọc e-book cần thiết bị hiện đại nhất định cũng như sự hiểu biết với chúng. Sự tiện dụng cũng như thân thuộc với mọi lứa tuổi làm cho sự tồn tại của sách giấy là không thể biến mất. Hắn là các bạn khi cầm trên tay một quyển sách sẽ có cảm giác rất khác so với khi đọc sách online rồi phải không? Sưu tập sách, tiểu thuyết, bách khoa toàn thư, truyện tranh,... là một sở thích phổ biến từ những người trẻ tuổi đến những người trưởng thành, đến cả những ông cụ bà cụ mắt đã mờ đến mức không nhìn rõ những dòng chữ chi chít.



Bộ sưu tập sách ngổn ngang

Là một người sưu tầm sách thì việc sắp xếp các cuốn sách một cách gọn gàng, ngăn nắp có vẻ là công việc khá lý thú khi tự mình được phân chia và ngắm nhìn các cuốn sách của bản thân. Tuy nhiên, việc này trở nên kinh khủng khi số lượng sách tăng lên hàng ngàn cuốn. Thư viện là một phiên bản khổng lồ của bộ sưu tập sách với số lượng sách có thể lên tới hàng triệu cuốn.



Thư viện chứa hơn 7 triệu cuốn sách - thư viện công cộng lớn nhất ở Turkey (Istanbul)

Thủ tướng tương bạn là một nhân viên trong thư viện này, công việc của bạn chính là sắp xếp sách theo thứ tự nhất định cũng như tìm kiếm cuốn sách mà mọi người muốn. Công việc vừa nghe thôi cũng đã muôn trốn.

Đối với những công việc này, làm bằng phương pháp thủ công rõ ràng là không hề hiệu quả và vô cùng tốn thời gian. Và như bao công việc phiền phức và nhảm chán khác, con người liền tìm cách đẩy công việc này sang cho máy tính, một thiết bị không thể thiếu con người thiết lập cho nó nhưng hiệu suất, tốc độ làm việc của nó cao gấp nhiều lần so với con người.

Việc quản lý thư viện với máy tính có vẻ chẳng phải việc gì xa lạ, dùng tên sách để sắp xếp rồi dùng các thuật toán hiệu quả để tìm kiếp hay sắp xếp hàng ngàn, hàng triệu cuốn sách. Nhân loại là sinh vật tham lam nên thế giới mới ngày càng cải tiến, việc quản lý sách bằng máy tính đã đem lại hiệu quả rõ rệt với thời gian sắp xếp cũng như tìm kiếm được tối ưu ở mức nào đó. Khi đã thõa mãn được những nhu cầu đó thì nhân loại lại bắt đầu nảy sinh ra những nhu cầu khác như: việc nhập liệu của từng cuốn sách vào máy tính để dễ dàng lấy dữ liệu cho việc sắp xếp hay tìm kiếm. Công việc này nghe có vẻ vô cùng đơn giản, nhưng cho dù đơn giản mà phải lặp đi lặp lại hàng triệu lần thì lại trở nên vô cùng nhảm chán, thậm chí dẫn đến sai sót. Từ đây ta lại có ý nghĩ đẩy công việc này lại cho máy tính.

Với nhu cầu này, nhóm đã tìm hiểu và kết hợp nhiều models cũng như kỹ thuật cần thiết để tạo thành một ứng dụng dùng để lấy thông tin từ ảnh bìa của một quyển sách. Ứng dụng mang tên "Số hóa tủ sách", cho phép người dùng đưa vào hình ảnh mặt trước của một quyển sách, sau đó cung cấp tất cả thông tin quan trọng có trên bìa sách.

Việc quan sát một quyển sách để tìm vị trí của từng thành phần như tên sách, tên tác giả, nhà xuất bản,... ở đâu cũng đã là một chuyện khá khó khăn và mất thời gian. Đến cả việc nhập liệu chúng. Nếu để con người làm việc này thì độ chính xác sẽ rất cao, tuy nhiên nhiều lúc sẽ có sai sót và hiệu suất sẽ rất thấp, đặc biệt là khi so sánh với máy tính.

Với ứng dụng "Số hóa tủ sách" này, nhóm hy vọng những việc trên sẽ trở thành tự động, từ việc tìm vị trí của từng thành phần của thông tin trên sách đến việc nhập liệu thông tin. Việc mà con người làm chỉ là cung cấp ảnh của cuốn sách và bỏ vào để ứng dụng đó cung cấp cho tacác thông tin cần thiết.

Việc chụp ảnh hàng triệu cuốn sách và chụp sao cho phù hợp với ứng dụng thì có vẻ cũng khá là rắc rối và nhàn chán, vấn đề này không nằm trong phạm vi nghiên cứu của đồ án nhưng có lẽ đã/đang/sẽ có ai đó tìm cách tự động hóa công việc này ở một mức nào đó, khi đó thì con người có thể để máy móc thực hiện mọi công việc nhàn chán và lặp đi lặp lại trong việc quản lý kho sách.

# Chương 1: Tổng quan

## I Mô tả bài toán

### 1 Ngữ cảnh ứng dụng

Số hóa tủ sách có thể được ứng dụng để số hóa các tủ sách tại gia của các gia đình, các thư viện tại trường học, thư viện công cộng hoặc bất kì nơi nào gặp khó khăn trong việc quản lý sách. Ứng dụng số hóa tủ sách sẽ tự động hóa việc tìm kiếm và nhập liệu các thông tin trên bìa sách, giúp giảm bớt thời gian và công sức cho việc nhập liệu thủ công, từ đó giảm gánh nặng cho việc quản lý tủ sách. Trong đồ án Số hóa tủ sách mà nhóm thực hiện lần này, ngữ cảnh ứng dụng cụ thể của bài toán là số hóa tủ sách trong nhà của các thành viên trong nhóm.

### 2 Input và output

- **Input:** Các ảnh chụp hình bìa sách trên nền đen bằng camera với chất lượng ảnh tối thiểu là Full HD (720x960)
- **Output:** File csv chứa tên các ảnh input và các thông tin trên bìa sách trong ảnh, cụ thể gồm 7 trường dữ liệu sau đây:
  - Tên file ảnh chụp hình bìa sách
  - Tên sách
  - Tên tác giả (+ người minh họa)
  - Nhà xuất bản
  - Tập (số tập/phần của một cuốn sách dài kỳ)
  - Người dịch
  - Tái bản (số lần tái bản)

Dưới đây là hình minh họa cho input và output của bài toán:

1.jpg



2.jpg



Input gồm 2 file ảnh chụp bìa sách là 1.jpg và 2.jpg

file names	tên sách	tên tác giả	nha xuất bản	tập	người dịch	tái bản
1.jpg	bồ câu không đưa thư	nguyễn nhật ánh	NHÀ XUẤT BẢN TRẺ			
2.jpg	TƯ DIỄN TRANH VỀ CÁC LOÀI CÂY	LÊ QUANG LONG	NHÀ XUẤT BẢN GIÁO DỤC			

Output sẽ gồm file csv chứa thông tin về bìa sách có trong 2 file ảnh này kèm với tên file ảnh tương ứng.

### 3 Lời giải cho bài toán

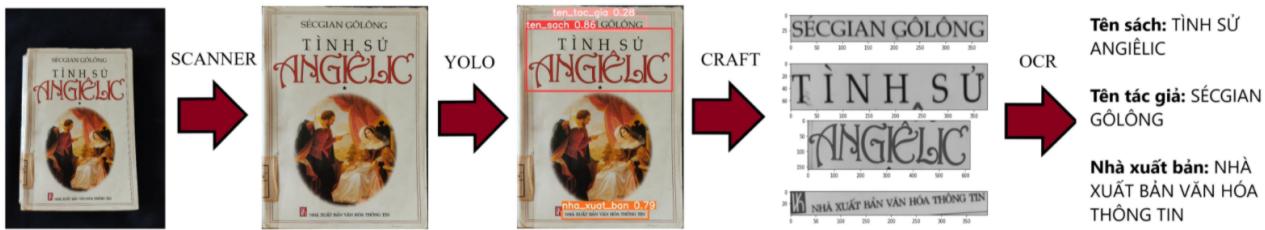
Để giải quyết bài toán Số hóa tủ sách, ta cần phải giải quyết các bài toán nhỏ hơn sau đây:

- **Xử lý ảnh input:** Từ ảnh input thô chụp hình bìa sách trên nền đen bằng camera smartphone, ta cần phải lấy ảnh bìa sách ra khỏi nền đen để thu được ảnh bìa sách gốc.
- **Object detection:** Với các ảnh bìa sách gốc này, ta cần detect các vùng trên bìa sách chứa thông tin thuộc 1 trong 6 trường dữ liệu sau: tên sách, tên tác giả, nhà xuất bản, tập người dịch, tái bản.
- **Text localization:** Sau khi đã detect được các vùng chứa 1 trong 6 thông tin quan trọng trên bìa sách, ta phải tìm vị trí chứa text trong các vùng này và cắt chúng ra dưới dạng những ảnh nhỏ chứa các dòng text.
- **Text recognition:** Từ các ảnh nhỏ chứa text này, ta sẽ nhận dạng các ký tự có trong chúng để thu được những dòng text thực sự dưới dạng văn bản.
- **Lưu kết quả:** Sau đó ta lưu trữ các dòng văn bản trên vào file csv, kết hợp với 1 trong 6 trường dữ liệu mà chúng thu được vào để hoàn tất việc thu thập thông tin từ bìa sách.

### 4 Các models mà nhóm sử dụng để giải quyết bài toán

- **Object detection:** sử dụng model [YOLOv5](#) cho bài toán Object Detection
- **Text localization:** sử dụng pretrained model [CRAFT text detector](#) của [CLOVA AI](#), model này có trên [pypi/craft-text-detector 0.3.5](#)
- **Text recognition:** sử dụng model [VietOCR](#) cho bài toán text recognition chữ tiếng Việt

Dưới đây là hình tóm tắt các khâu cần thực hiện trong quá trình Số hóa tủ sách:



Các khâu của quá trình Số hóa tủ sách

Các thông tin thu được từ các ảnh input chụp bìa sách trên nền đen sẽ được lưu trữ trong 1 file csv để ra được output hoàn chỉnh.

## II Mô tả dữ liệu thu thập

### 1 Ảnh input

- Mô tả:** gồm 320 ảnh bìa sách được chụp bằng camera smartphone mô phỏng như một webcam chụp trực tiếp xuống bìa sách trên nền đen với chất lượng ảnh tối thiểu là Full HD (720x960). 320 quyển sách này được lấy từ tủ sách gia đình của các thành viên trong nhóm.
- Hạn chế trong quá trình thu thập:** Do dịch Covid, các thành viên trong nhóm chưa có dịp ghé nhà sách để chụp ảnh nên số lượng ảnh input vẫn chưa được nhiều.
- Minh họa dữ liệu thu thập:**



Các ảnh input chụp bìa sách trên nền đen

- Link data:** [Link 320 ảnh input](#)

### 2 Data dành để train model YOLOv5

- Mô tả:** Hơn 7000 ảnh bìa sách được crawl từ trang web của nhiều nhà xuất bản khác nhau như:
  - Nhà xuất bản Trẻ
  - Nhà xuất bản Kim Đồng
  - Nhà xuất bản DHQG-TPHCM

- Nhà xuất bản Hà Nội
- Nhà xuất bản Đà Nẵng
- Nhà xuất bản khoa học xã hội
- Nhà xuất bản thanh niên
- Và nhiều nhà xuất bản khác...

Nhóm sử dụng online tool [makesense.ai](https://makesense.ai) để dán nhãn gần 7000 ảnh bìa sách này, thu được gần 7000 file .txt chứa tọa độ các bounding boxes của các vùng có thông tin cần thu thập trên bìa sách.

- **Hạn chế trong quá trình thu thập:** Ban đầu nhóm crawl được khoảng 15000 ảnh bìa sách. Nhưng trong đó có các ảnh độ phân giải thấp nhìn mờ, gây mất thời gian để lọc ra và cuối cùng chỉ còn lại được khoảng 7000 ảnh.
- **Minh họa dữ liệu thu thập:**



Data cho model YOLOv5 gồm các ảnh có kích thước tối thiểu là 560x720

- **Link data:** [Link data cho model YOLOv5](#)

### 3 Data dành để train model VietOCR

- **Mô tả:** gồm các data sau:

- 45000 dòng text được cắt ảnh ra từ 7000 ảnh bìa sách đã crawl được. Trong 45000 dòng text này, các thành viên đã sử dụng công cụ [VGG Image Annotator \(VIA\)](#) và dán nhãn được hơn 22000 dòng text.
- 100000 dòng text được generate cũng là data lấy từ GitHub của VietOCR

- **Minh họa dữ liệu thu thập:**

- Data mà nhóm label:

Lời: Nguyễn Trần Thiên Lộc

Mĩ thuật: Vũ Thị Thùy

NHÀ XUẤT BẢN KIM ĐỒNG

Người dịch: Thanh Hà

Các ảnh chứa text mà nhóm cắt ra từ 7000 sách

– Data có sẵn của VietOCR:

Bản chất của thành công

chiếc xe thể hệ 3

Cách giảm độ đèn bắt ngđ.

Britannia On Fawkes

Các ảnh chứa chữ viết tay

Các ảnh chứa text được generate

- **Link data:** [Link data cho model VietOCR](#)

# Chương 2: Các nghiên cứu trước

## I Mô hình YOLO cho Object Detection

### 1 Giới thiệu về YOLO

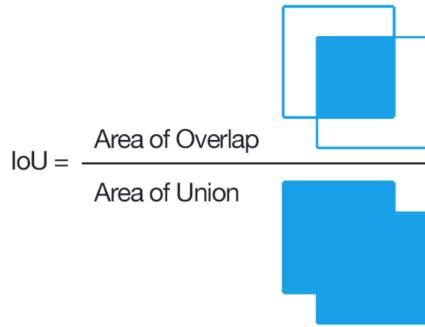
YOLO (You only look once) là một mô hình mạng CNN (Convolutional Neural Network) dành cho việc phát hiện, nhận dạng và phân loại đối tượng. Các mô hình R-CNN (Region Based CNN) trước đó dành cho object detection phải trải qua hai giai đoạn là dự đoán các bounding boxes có khả năng và chạy một classifier để phân loại chúng. Sau đó, mô hình sẽ tinh chỉnh lại các bounding boxes đã được phân loại, loại bỏ các phát hiện trùng nhau cũng như các bounding boxes không chứa object. Quá trình này chậm, phức tạp và khó tối ưu vì mỗi giai đoạn khác nhau diễn ra riêng biệt với nhau. Nhưng với mô hình YOLO, object detection sẽ được xem như một vấn đề hồi quy duy nhất, đi thẳng từ các pixels trong ảnh cho đến các bounding boxes cùng xác suất class của chúng. Điều này giống với việc chỉ nhìn vào ảnh một lần duy nhất và xác định được có những objects nào và chúng ở đâu trong ảnh.

## 2 Mô hình YOLO

### 2.1 Ý tưởng

Mô hình YOLO thống nhất tất cả giai đoạn trong object detection thành một neural network duy nhất. Network này sử dụng toàn bộ các features của ảnh để dự đoán mọi bounding boxes của mọi classes trong ảnh.

Sử dụng YOLO, ảnh input sẽ được resize và chia thành một lưới (grid) gồm  $S \times S$  ô vuông. Nếu tâm của object rơi vào ô nào thì ô đó chịu trách nhiệm detect object đó. Mỗi ô sẽ dự đoán  $B$  bounding boxes và confidence score cho mỗi box. Confidence score là điểm số phản ánh độ chắc chắn có object trong bounding box hay không cũng như độ chính xác của bounding box được dự đoán. Cụ thể, confidence score được định nghĩa là  $P(\text{Object}) * IOU_{pred}^{truth}$ . Trong đó,  $P(\text{Object})$  là xác suất có object trong ô và  $IOU_{pred}^{truth}$  là Intersection Over Union của bounding box dự đoán và ground truth box (bounding box đã được label và đưa vào làm training/testing data) với IOU bằng  $\frac{\text{Diện tích giao nhau của 2 bounding boxes}}{\text{Diện tích hợp nhau của 2 bounding boxes}}$ .

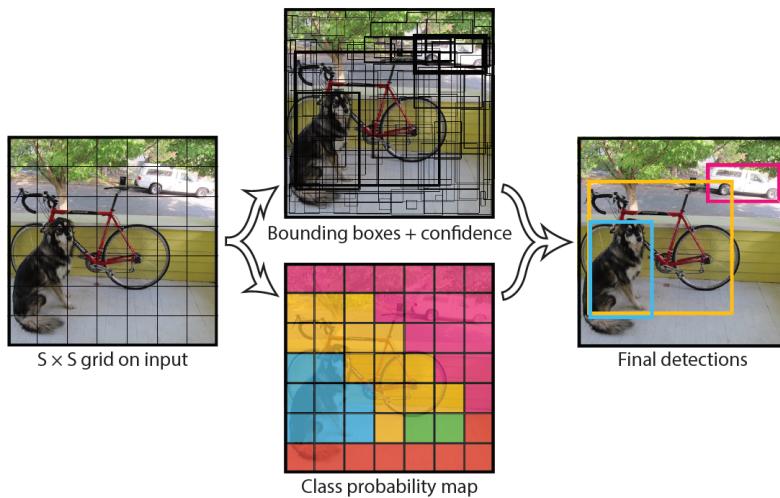


Như vậy, confidence score sẽ bằng 0 nếu object không có trong ô. Ngược lại nếu ô có object, confidence score sẽ bằng IOU giữa bounding box dự đoán và ground truth box.

Mỗi bounding box được dự đoán với 5 tham số:  $x, y, w, h$  và confidence. Trong đó,  $(x, y)$  là tọa độ tâm bounding box so với các đường giới hạn của ô chứa nó. Giá trị  $w, h$  lần lượt là chiều rộng và chiều cao của bounding box dự đoán và confidence thể hiện IOU giữa bounding box dự đoán và ground truth box.

Mỗi ô trong ảnh cũng dự đoán  $Pr(Class_i|Object)$  là xác suất roi vào mỗi class của ô. Đây là xác suất có điều kiện với điều kiện là ô có chứa object. Các giá trị xác suất cho C classes sẽ tạo ra C outputs cho mỗi ô. B bounding boxes của cùng một ô sẽ chia sẻ chung một tập các dự đoán về class của object, đồng nghĩa với việc tất cả các bounding boxes trong cùng một ô sẽ có chung một class. (Từ version YOLOv2 trở lên đã có thể detect được nhiều class) Vào thời điểm test, xác suất class của ô sẽ được nhân với tham số confidence dự đoán được của mỗi bounding box để ra được confidence score theo class cho mỗi box:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$$

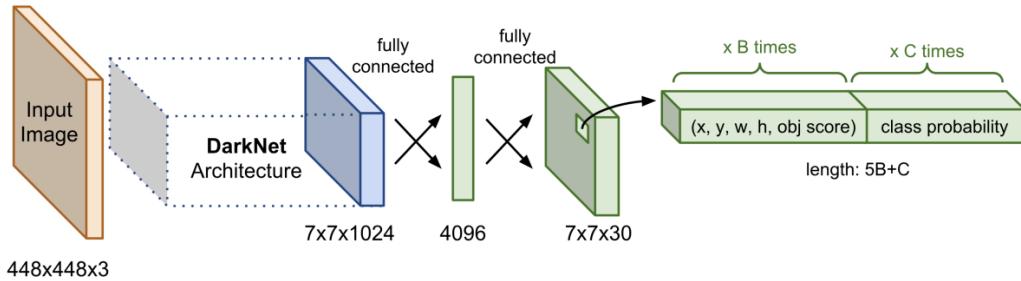


Ảnh input được chia thành lưới  $S \times S$  ô, mỗi ô dự đoán B bounding boxes (5 tham số) và C xác suất class nên kích thước output là  $S \times S \times (B*5 + C)$

## 2.2 Kiến trúc Network

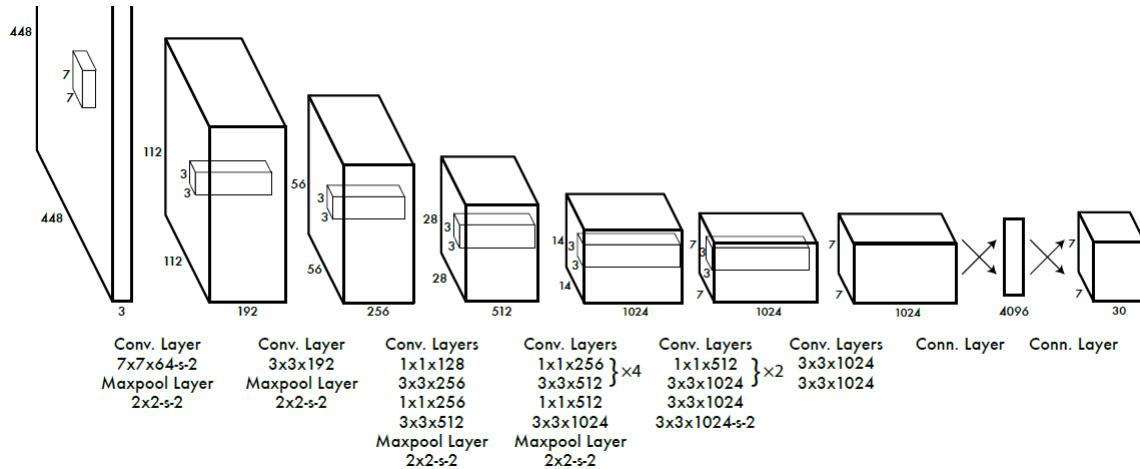
Kiến trúc mạng YOLO được lấy cảm hứng từ mô hình GoogLeNet dành cho phân loại hình ảnh. Mô hình YOLO được thiết kế để bao gồm một kiến trúc xử lý tất cả các features của hình ảnh (được các tác giả gọi là kiến trúc Darknet) và sau là 2 fully connected layers

thực hiện dự đoán tọa độ và xác suất class của các bounding boxes được cho là chứa object. Mô hình này đã được dùng để đánh giá bộ dữ liệu Pascal VOC mà trong đó các tác giả sử dụng  $S = 7$ ,  $B = 2$  và  $C = 20$ . Điều này giải thích tại sao các feature maps cuối cùng là  $7 \times 7$  và kích thước output là  $(7 \times 7 \times (2^5 + 20))$ .



Kiến trúc tổng quan của YOLOv1

Các tác giả đã giới thiệu mô hình fast-YOLO với kiến trúc Darknet dùng 9 convolutional layers cho các tập dữ liệu không phức tạp và mô hình YOLO bình thường với kiến trúc Darknet dùng 24 convolutional layers có thể xử lý các tập dữ liệu phức tạp hơn để tạo độ chính xác cao hơn. Thứ tự của các convolutional layers  $1 \times 1$  và  $3 \times 3$  được lấy cảm hứng từ mô hình GoogLeNet giúp giảm không gian features từ các layers trước. Layer cuối cùng sử dụng hàm kích hoạt tuyến tính thay vì Leaky Rectified Linear Unit (Leaky ReLU) như các layers trước.



Kiến trúc mạng YOLO với 24 convolutional layers và 2 fully connected layers theo sau

Từ YOLOv2 trở lên, kiến trúc network đã được cải tiến với số layers nhiều hơn cũng như cách thiết kế và bố trí các layers hoàn thiện hơn để làm tăng tính ổn định và cải thiện khả năng nhận diện của model.

### 2.3 Loss function

YOLO sử dụng sum-squared error làm loss function vì đây là hàm dễ tối ưu hóa. Tuy nhiên, hạn chế của nó là xem localization loss (độ lỗi vị trí bounding box) ngang bằng với classification loss (độ lỗi phân loại). Hơn nữa, phần lớn các ô trong ảnh đều không chứa objects, điều này làm cho confidence scores của những ô đó bị đẩy về 0, áp đảo gradient của những ô chứa object. Để tránh áp đảo như vậy dẫn đến phân kỳ (divergence) trong quá

trình training và khiến cho model mất ổn định từ sớm, các tác giả cho tăng độ lỗi của tọa độ các bounding boxes thông qua hằng số  $\lambda_{coord} = 5$  và giảm độ lỗi của tham số confidence đối với những boxes không chứa object thông qua  $\lambda_{noobj} = 0.5$ .

$$\begin{aligned}\mathcal{L} = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2\end{aligned}$$

#### Loss function của YOLO

Phần đầu tiên trong loss function là localization loss. Nó tính sai số giữa vị trí bounding box dự đoán và ground truth box dựa trên tọa độ tâm (x, y), chiều ngang w và chiều cao h. Giá trị  $\mathbb{I}_{ij}^{obj}$  được định nghĩa bằng 1 nếu có object trong bounding box thứ j của ô thứ i và bằng 0 nếu ngược lại. Trong phần này, căn bậc hai của w và h được sử dụng thay cho w và h vì chiều rộng và chiều cao đã được chuẩn hóa từ 0 đến 1, sử dụng căn bậc hai sẽ giúp tăng hiệu suất.

$$\begin{aligned}& \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]\end{aligned}$$

#### Localization loss

Phần thứ hai là confidence loss. Nó tính sai số giữa tham số confidence dự đoán và tham số confidence thực sự cho cả hai trường hợp bounding box có và không có object. Giá trị  $\mathbb{I}_{ij}^{noobj}$  được định nghĩa bằng 1 nếu không có object trong bounding box thứ j của ô thứ i và bằng 0 nếu ngược lại (ngược lại với  $\mathbb{I}_{ij}^{obj}$ ).

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

#### Confidence loss

Phần cuối cùng của loss function là classification loss, tính sai số giữa xác suất class dự đoán và xác suất class thực sự. Tuy nhiên, YOLO không phạt lỗi phân loại sai trong trường hợp không có object trong ô vì khi đó giá trị  $\mathbb{I}_i^{obj} = 0$ .

$$\sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

#### Classification loss

## II CRAFT text detector cho Text localization

### 1 Giới thiệu CRAFT text detector

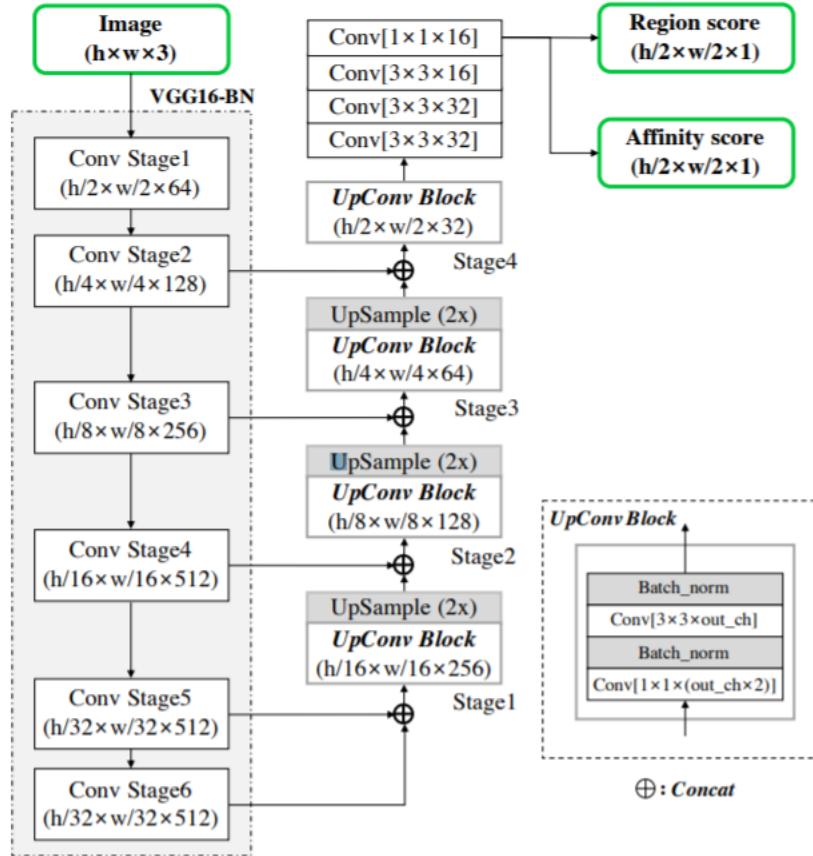
Tác giả: Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, Hwalsuk Lee.  
Nhóm sử dụng model deep-learning có sẵn trên pypi/craft-text-detector 0.3.5: [CRAFT: Character-Region Awareness For Text detection](#) để thực hiện locate các text trên bìa sách  
Đây là một PyTorch dùng cho craft text detection, nó detect được khá hiệu quả bằng cách tìm ra phân vùng của từng từ chữ cái và mối quan hệ giữ các chữ cái đó. Nó tạo ra hộp chữ nhật chứa các đoạn text dựa vào mối quan hệ giữ các chữ nó tách ra được.  
Nhóm sử dụng đoạn code có sẵn trên pypi, chỉ điều chỉnh một số tham số để thực hiện craft ảnh bìa sách.  
Ứng dụng này bao gồm:

- Input: ảnh cần nhận diện chữ.
- Output: một số ảnh đã được crop tự động sau khi nhận diện được.

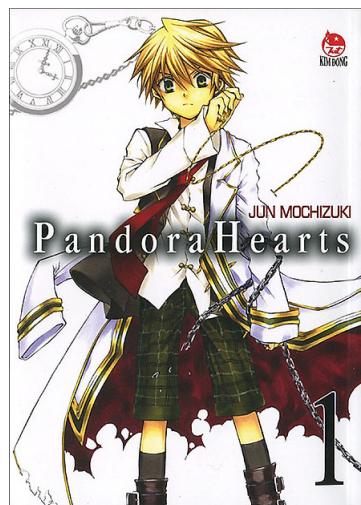
craft-text-detector output còn có nhiều thông tin khác như: text-detection.txt, heatmap,...

### 2 Kiến trúc network

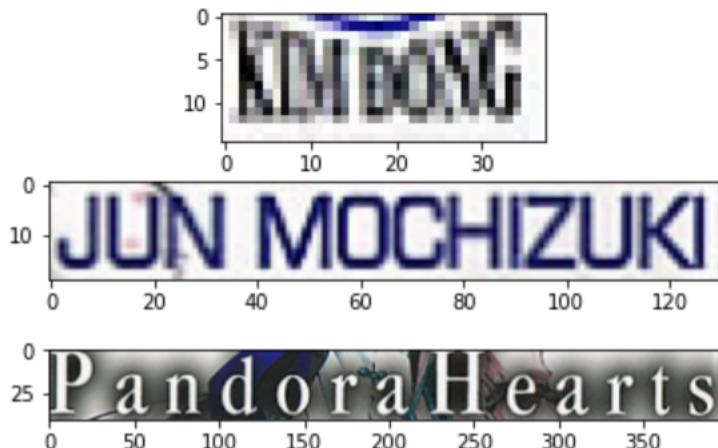
Theo [bài báo](#) chính thức trên [github](#) thì ứng dụng này hoạt động với mục đích chính là locate chính xác từng ký tự trong ảnh. Họ train một deep-learning neural network để predict ra vị trí của ký tự và mối quan hệ của chúng với nhau. Họ train model bằng một mạng tích chập đầy đủ được minh họa như sau:



Cấu trúc mạng sử dụng trong model.



Ảnh cần craft



Kết quả thu được

### III Mô hình VietOCR cho Text recognition

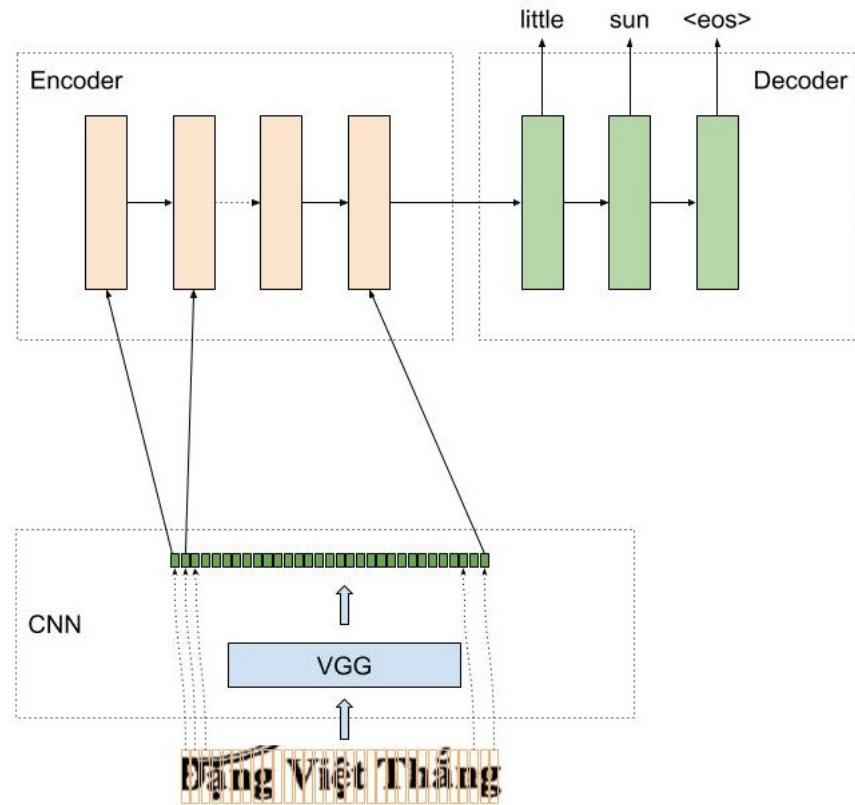
#### 1 Giới thiệu mô hình VietOCR

Thư viện này kết hợp CNN cùng hai mô hình khá nổi tiếng trong việc xử lý ngôn ngữ tự nhiên (cũng như về mặt hình ảnh) là: Transformer và Attention của seq2seq. Đây đều là những mô hình nổi tiếng, hiệu quả, đã được khắc phục nhiều hạn chế của các mô hình trước đó.

Đặc biệt là Transformer (mới xuất hiện gần đây), khắc phục được tốc độ train của model sử dụng RNN cũng như về độ chính xác. Tuy nhiên Transformer lại predict khá chậm (cụ thể là so với Attention).

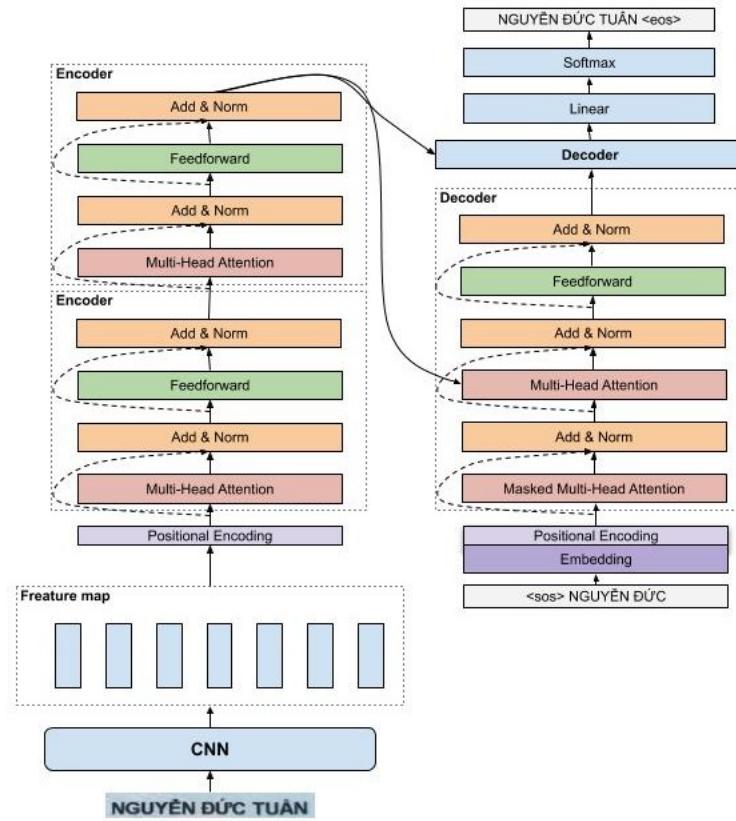
#### 2 Kiến trúc network

##### AttentionOCR



Mô hình dùng CNN để trích xuất đặc trưng sau đó đi qua seq2seq sử dụng cơ chế attention.

### TransformerOCR



Mô hình sử dụng CNN để trích xuất đặc trưng sau đó đi qua transformer.

Dầu tiên là nhóm không sử dụng model pretrain vì khi thử nó vô cùng không chính xác, gần như độ chính xác rất thấp.

Nhóm chọn model Transformer\\_OCR do nó train nhanh hơn và có độ chính xác cao hơn nhiều so với Attention\\_OCR, điểm bất lợi duy nhất so với mô hình kia chính là thời gian predict chậm hơn như đã đề cập ở trên.

**Model Zoo** Mô hình này được huấn luyện trên tập dữ liệu gồm 10m ảnh, bao gồm nhiều loại ảnh khác nhau như ảnh tự phát sinh, chữ viết tay, các văn bản scan thực tế. Pretrain model được cung cấp sẵn. Model này có vẻ thích hợp với các tài liệu scan, đánh máy trên giấy,...

Mô hình được train bằng 2 phương pháp attention và cả transformer với độ chính xác cùng thời gian predict như sau:

Backbone	Config	Precision full sequence	time
VGG19-bn - Transformer	vgg_transformer	0.8800	86ms @ 1080ti
VGG19-bn - Seq2Seq	vgg_seq2seq	0.8701	12ms @ 1080ti

Ta có thể thấy độ chính xác của transformer cao hơn nhưng thời gian predict lại lâu hơn.

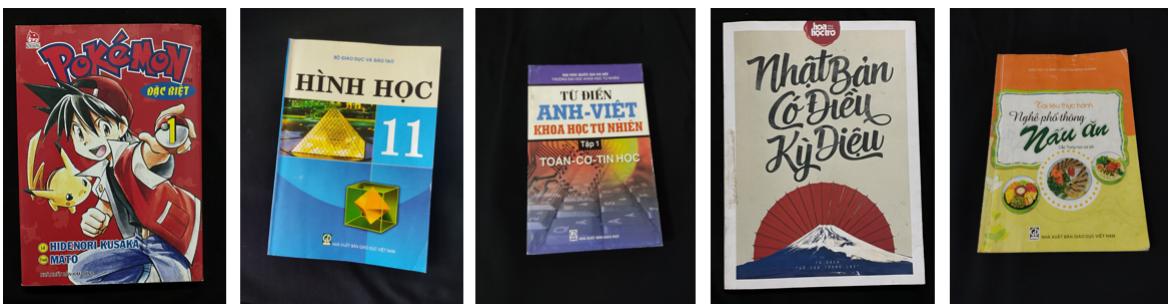
# Chương 3: Xây dựng bộ dữ liệu

## I Tiêu chí xây dựng bộ dữ liệu

### 1 Ảnh input

Sau khi chụp bìa sách trên các màu nền khác nhau, các thành viên trong nhóm thống nhất với nhau chụp các bìa sách trên nền đen để dễ sử dụng find contour cho bước scan ảnh bìa sách ra khỏi nền.

Các ảnh input chứa sách thuộc nhiều thể loại khác nhau và được chụp dưới nhiều góc khác nhau một cách ngẫu nhiên, mỗi ảnh chỉ chụp 1 bìa sách.



Các ảnh input mà nhóm đã chụp

### 2 Data dành để train model VietOCR

Nhóm không dán nhãn các ảnh chứa text dọc, text đa dòng hay ảnh không chứa text. Những data mà nhóm dán nhãn là những dòng text nằm ngang

VD: Data mà nhóm dán nhãn

Tác giả: Cécile Jugla và Jack Guichard

Lời: Nguyễn Trần Thiên Lộc

Người dịch: Thanh Hà

Tranh: Nguyễn Công Hoan - Biên soạn: Hiếu Minh

NHÀ XUẤT BẢN KIM ĐÔNG

Các ảnh chứa text ngang

VD: Data mà nhóm không dán nhãn



**DRAGON BALL**  
7 VIÊN NGỌC RỒNG

câu chuyện  
lãng mạn

Những kẻ  
(trong bong tôi)



Các ảnh chứa text dọc

Các ảnh chứa text đa dòng

Các ảnh không chứa text

## II Các mẫu dữ liệu khó

### 1 Các mẫu khó đối với model YOLOv5

- Các mẫu có tên tác giả to hơn tên sách mà tên sách lại ngắn hơn tên tác giả:



- Các mẫu có tên sách và tên tác giả có size, font chữ khá giống nhau và viết gần nhau:

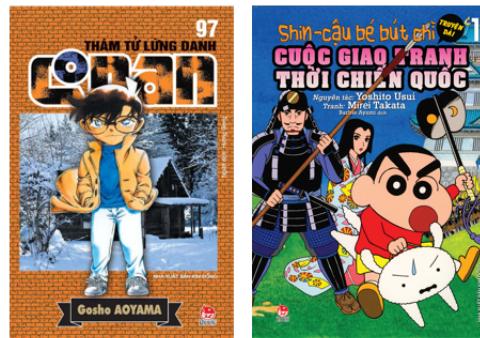


- Các mẫu có quá nhiều chữ trên bìa sách:



## 2 Các mẫu khó đối với model VietOCR

- Các mẫu có chữ bị che bởi hình minh họa:



- Các mẫu có font chữ lạ, chữ được viết kiểu,... hoặc có chữ viết xéo, cong,...:



- Các mẫu có chữ in chìm ở phía sau:



# Chương 4: Training và đánh giá model

## I Preprocessing

Đây là khâu xử lý ảnh input chụp bìa sách để lấy được hình bìa sách gốc ra khỏi nền đen và điều chỉnh góc nhìn bìa sách sao cho máy tính có thể dễ dàng thu được các thông tin trên bìa trong các khâu sau đó.

Cũng giống như con người, để đọc được chữ trên quyển sách một cách dễ dàng thì cần phải nhìn cuốn sách ở góc chính diện, không bị nghiêng cũng như có ánh sáng tốt. Tuy để nghiêng hay ở nơi tối thì vẫn tùy vào thị lực của từng người mà có thể đọc được. Nhưng đối với máy tính, các bìa sách ở góc chính diện trực tiếp nhìn thẳng vào bìa sách kết hợp cùng ánh sáng đủ tốt thì những thông tin trên bìa mới hiện ra rõ ràng, không bị méo mó và dễ được nhận diện hơn bởi các mô hình máy học.

Khâu preprocessing các ảnh input gồm các bước sau:

- Chuyển ảnh input thành Gray Scale
- Sử dụng Gaussian Blur để làmぼt nhiễu ở nền vì nền vải đen trong một số ảnh input còn chưa được bằng phẳng nên bị nhiễu sáng.
- Sử dụng Canny Edge Detection của thư viện OpenCV trích xuất các cạnh của bìa sách trong ảnh.
- Sử dụng kỹ thuật Erosion và Dilation của thư viện OpenCV để làm giảm nhiễu của các cạnh bìa sách bằng cách làm dày các cạnh này lên. Ta cần bước này vì các cạnh bìa sách sau khi qua Canny Edge Detection vẫn còn mỏng và chưa được liền nhau.
- Sử dụng find contour của thư viện OpenCV để xác định đường bao quanh bìa sách trong ảnh. Ta chọn ra contour lớn nhất với approximate contour có số lượng là 4, tương ứng với 4 điểm của một tứ giác, chính là 4 góc bìa sách.
- Dùng kỹ thuật Perspective Warping để chuyển góc nhìn của bìa sách đã find contour ra được bìa sách ở góc chính diện.



## II Object detection

Trong khâu Object detection, vì dữ liệu là bìa sách các Object đều chứa chữ nên việc nhầm lẫn object có thể xảy ra. Việc sử dụng YOLOv5 cho bài toán này vì model có độ chính xác cao và thời gian dự đoán nhanh. Cụ thể, nhóm sử dụng [YOLOv5](#) vì đây là phiên bản mới nhất với nhiều đặc điểm cải thiện và cách dùng đơn giản hơn các phiên bản trước.

Dây là [Colab Notebook](#) ghi nhận lại cách mà nhóm đã train YOLOv5.

### 1 Chuẩn bị training và testing data

**Tổng cộng:** 6982 ảnh bìa sách (do nhóm crawl) và file.txt (do nhóm dán nhãn). Trong đó:

- **Training data (85%):** 5951 ảnh và file.txt, tỉ lệ train:val là 8:2 ([Link training data](#))
  - Tập train: 4760
  - Tập val: 1191
- **Testing data (15%):** 1031 ảnh và file.txt ([Link testing data](#))

Cụ thể số label trong các file.txt của training và testing data như sau:

Label	Training data	Testing data
all	20920	3763
ten sach	6627	1135
ten tac gia	5207	927
nha xuat ban	5408	959
tap	1095	347
nguoil dich	2219	354
tai ban	364	41

Trong đó, số label của tên sách, tên tác giả và nhà xuất bản là nhiều nhất và số label của tái bản là ít nhất.

## 2 Tổng kết quá trình training

- Kích thước data: các ảnh size 736x736
- Model: YOLOv5x6 (extra-large), 607 layers
- Batch size: 8
- GPU RAM sử dụng: 11.8GB
- Quá trình train: train qua 45 epoch
- Thời gian train: 6.055 giờ
- Loss:
  - Localization loss: 0.02127
  - Confidence loss: 0.01714
  - Classification loss: 0.003347
- mAP0.5(all class): 0.90428

## 3 Đánh giá kết quả

- Trên tập validation:

Class	Precision	Recall	mAP0.5
all	0.929	0.948	0.965
ten sach	0.965	0.976	0.983
ten tac gia	0.936	0.923	0.965
nha xuat ban	0.93	0.977	0.98
tap	0.952	0.978	0.969
nguoil dich	0.936	0.94	0.964
tai ban	0.855	0.895	0.925

- Trên tập test:

Class	Precision	Recall	mAP0.5
all	0.862	0.869	0.862
ten sach	0.905	0.905	0.923
ten tac gia	0.879	0.849	0.894
nha xuat ban	0.893	0.97	0.905
tap	0.741	0.66	0.626
nguoi dich	0.851	0.921	0.905
tai ban	0.903	0.909	0.919

Trên tập validation, giá trị Precision và Recall của các class không chênh lệch nhau nhiều. Giá trị mAP0.5 trên all class là 0.965 được nhóm đánh giá là rất tốt, đặc biệt mAP của class tên sách và nhà xuất bản. Trong đó mAP của class tái bản là thấp nhất vì số lượng data của class này vẫn chưa được nhiều.

Trên tập test, giá trị Precision và Recall của các class chênh lệch nhau ít hơn nhưng cũng thấp hơn so với tập validation. Giá trị mAP0.5 trên all class là 0.862 được nhóm đánh giá ở mức khá tốt. Trong đó, mAP của class tên sách là cao nhất và của class tạp là thấp nhất và thấp hơn nhiều so với tập validation.

- Detect thử các mẫu trong testing data (confidence = 0.5):



Minh họa một vài mẫu detect đúng

Model detect đúng phần lớn các mẫu trong testing data. Các mẫu detect đúng bao gồm từ các mẫu đơn giản chỉ gồm vài labels như tên sách, tên tác giả và nhà xuất bản cho đến các mẫu phức tạp với nhiều labels hơn.



Minh họa một vài mẫu không detect được hoặc detect nhầm

Trên đây là một vài mẫu không detect được hoặc detect nhầm do có chữ dọc như trong mẫu đầu (data nhóm dán nhãn chủ yếu là chữ ngang, chữ dọc chỉ có trong một vài truyện tranh nên dán nhãn không được nhiều) hoặc rơi vào trường hợp data khó (tên tác giả lớn hơn tên sách, trên bìa sách có quá nhiều chữ,...) và mẫu cuối thì detect nhầm tên sách thành tên tác giả (do một vài sách có thêm tên tiếng Anh bên dưới với size chữ nhỏ hơn tên sách tiếng Việt nên có thể bị detect nhầm thành tên tác giả).

### III Text localization

Sau khi đã train model yolov5 để detect được các thành phần trên một cuốn sách, ta đã hoàn thành bước tự động hóa công việc đầu tiên mỗi khi ta nhìn vào cuốn sách để nhập liệu.

Việc còn lại mà ứng dụng "Số hóa tủ sách" này cần làm chính là biến những vùng đã detect thành dữ liệu văn bản (text) - công việc chính mà ứng dụng hướng tới. Chúng ta tiến đến bước OCR với chức năng nhận dạng và bóc tách data tự động.

Trước bước đó thì ta sẽ nhận diện chữ từ ảnh bằng craft để tạo train data cho ocr.

Bước này nhóm sử dụng model deep-learning có sẵn trên pypi/craft-text-detector 0.3.5: [CRAFT: Character-Region Awareness For Text detection](#)

Model này nhóm chỉ sử dụng mô hình đã pretrain để tiết kiệm thời gian, chỉ chỉnh các tham số để cho kết quả chính xác hơn:

#### Thiết lập tham số:

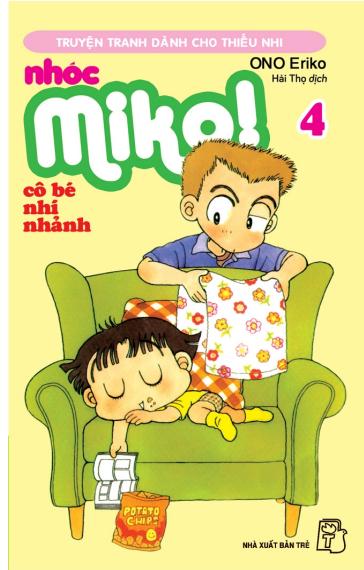
- Với tên sách:
  - text\_threshold=0.7
  - link\_threshold=0.3
  - low\_text=0.3
- Với tập:
  - text\_threshold=0.7
  - link\_threshold=0.1
  - low\_text=0.05
- Với còn lại:
  - text\_threshold=0.7
  - link\_threshold=0.1
  - low\_text=0.2

**Underfit trong CRAFT:** Vì thời gian có hạn hổ thực hiện đồ án nên nhóm đã chưa chuẩn bị kịp dữ liệu cho quá trình CRAFT. Vì sử dụng model đã train sẵn nên gây ra nhiều trường hợp underfit không mong muốn.

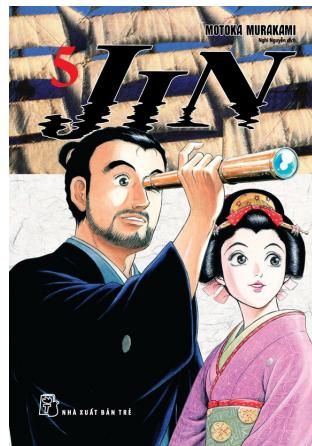
Sau đây là một số ảnh craft được, về cơ bản thì đa số nó craft được vẫn khá là tốt.



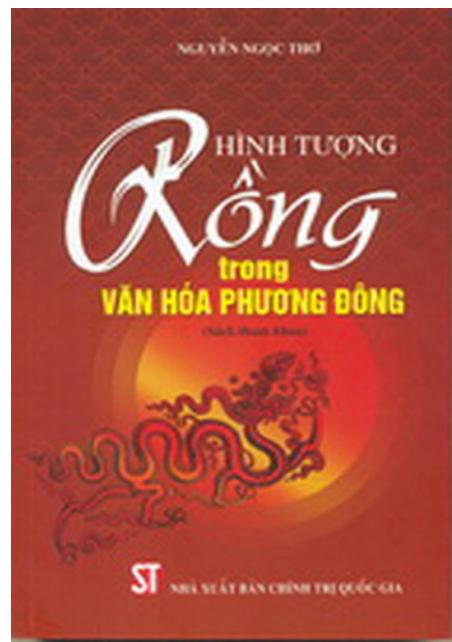
Mẫu chữ đơn giản, có phần hơi nghiêng nhưng vẫn thẳng 1 hàng thì craft vẫn ra kết quả như mong đợi.

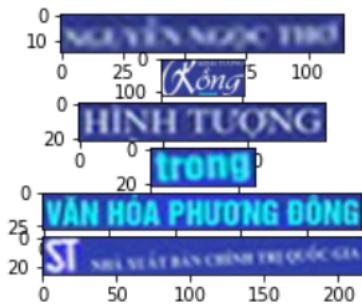


Với chữ bì ảnh minh họa đè lên 1 phần thẻ này nhưng nó vẫn ra kết quả như mong đợi.  
Tuy nhiên ở đây lại bị sót mất nhà xuất bản.

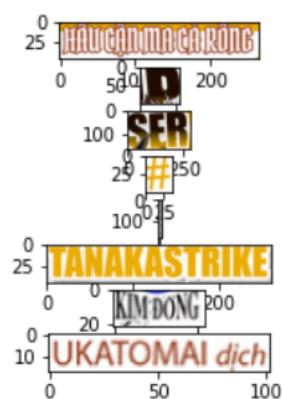


Dường như lấy được tất cả nhưng không lấy được tên truyện là "JIN", font chữ khá đặc biệt, chữ to chữ nhỏ + độ cao khác biệt + bị hình minh họa che mắt.



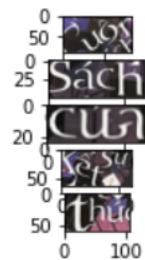


Từ "Rồng" bị cắt mất phần đầu chữ R in hoa và phần đuôi của chữ g, có lẽ là do 2 phần này bị lồng với từ "HÌNH TƯỢNG" và từ "trong".

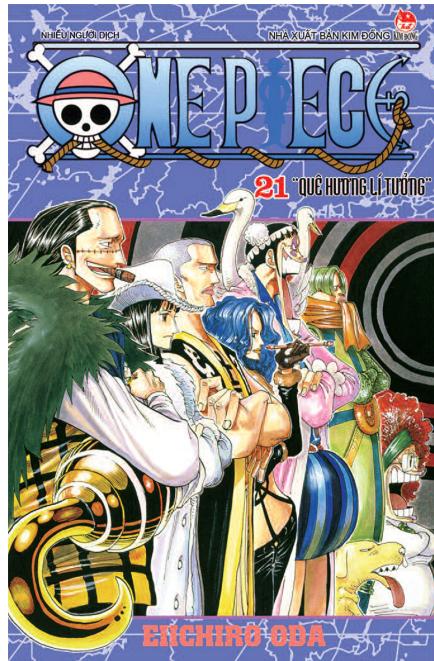


Tên sách tiếng việt "HÀU CẬN MA CÀ RỒNG" bị xét mất 1 phần của dấu, do model này được train multi language nên tiếng việt ít nhiều sẽ bị xót chút ít dấu. Còn tên sách tiếng anh "SERVAMP" bị ảnh minh họa che gần 1/3 chữ nên chỉ lấy được từ "SER" và 1 phần

chữ "P".



Có thể thấy các vùng thông tin trên cuốn này craft ra rất tệ, đặc biệt là tên sách. Các từ trên sách lồng liên tục vào nhau từ trên xuống dưới, do craft cố gắng cắt thành hình chữ nhật và có vẻ không được cắt phạm vào ô của từ khác nên kết quả trở nên lộn xộn như thế.



Các thông tin trên sách được lấy ra gần như hoàn hảo trừ chữ "O" trong từ "ONE PIECE" vì chữ này là font chữ độc nhất vô nhị được thiết kế riêng cho truyện này. Font chữ của bìa truyện tranh nhiều lúc được thiết kế độc quyền cho bộ truyện đó, nhiều khi rất khó để dự đoán.

**Training:** Sử dụng model đã được train sẵn multi language của Clovaai. Sau đó thì output của thao tác trên sẽ trở thành input để train model của text extraction. Để train model thì cần thêm expect output, để dễ dàng tạo output thì nhóm dùng tool có sẵn.

## IV Text recognition

Như đã đề cập ở phần trước thì đây là phần chính chúng ta cần làm để có được kết quả từ đầu đã nghĩ tới đó là trích xuất chữ ra khỏi hình ảnh. Ở đây nhóm sử dụng OCR mà cụ thể hơn là thư viện có sẵn [VietOCR](#).

Đây là [Colab Notebook](#) ghi nhận lại cách mà nhóm đã training VietOCR.

### 1 Training và testing data

Tổng cộng 22600 ảnh chữ từ bìa sách và 100000 ảnh generate. Trong đó:

- Training data:
  - 18080 ảnh chữ từ bìa sách
  - 100000 ảnh generate
- Validation: 4520 ảnh chữ từ bìa sách.

### 2 Quá trình training

- Iter = 10000
- Device: 'cuda:0'
- Có sử dụng checkpoint để lưu lại kết quả tốt nhất.
- Thời gian train khoảng 4 giờ

**\*Lưu ý:** Vì trong tập train chỉ tập trung vào các dòng text nằm ngang vậy nên các dòng text đọc sẽ được xử lý như sau. Đối với các dòng text bị sai hướng (các chữ được cho là đọc: có chiều cao lớn hơn 2 lần chiều rộng) thì ta xoay ảnh 90 độ theo chiều kim đồng hồ và ngược chiều kim đồng hồ. Ta sẽ có 3 ảnh (ảnh gốc, ảnh xoay trái, ảnh xoay phải). Ta tiếp tục dự đoán chữ trên 3 ảnh đó và lấy ra kết quả của dự đoán có điểm dự đoán cao nhất. Từ đó ta có thể tối ưu được việc dự đoán ảnh dòng text cả đọc và ngang.

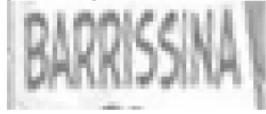
### 3 Dánh giá

Kết quả nhận được đáng mong đợi:

- train loss: 0.553
- valid loss: 0.538, acc full seq: 0.8577, acc per char: 0.9643
- Trên tập train precision đạt: 0.964233
- Trên tập val precision đạt: 0.85958

85% là một con số ấn tượng, tuy nhiên thì độ sai sót vẫn còn khá cao, đặt biệt là đối với các font chữ đặc biệt. Model nhận diện khá tốt các dòng text với font chữ đơn giản, cách bố trí rõ ràng, dễ nhìn, chẳng hạn như dưới đây:

prob: 0.933 - pred: BARRISSINA - actual: BARRISSINA



prob: 0.929 - pred: NHỈ? - actual: NHỈ?



prob: 0.916 - pred: MÂY - actual: mây



prob: 0.935 - pred: LÚC - actual: LÚC



Minh họa một vài mẫu nhận diện đúng

Model nhận diện khá tốt các mẫu trên, dù ảnh có độ phân giải thấp nhưng các kí tự và dấu vẫn được nhận diện khá tốt với xác suất dự đoán đều lớn hơn 0.9.

Tuy nhiên, vẫn có khá nhiều ảnh chứa text bị nhận diện sai các kí tự như dưới đây:

prob: 0.91 - pred: TIN – actual: JIN



prob: 0.92 - pred: KÌ QUAN – actual: KÌ QUAN



prob: 0.9 - pred: Nguyên tác – actual: Nguyên tác



prob: 0.73 - pred: Girls in Dome – actual: Girls in love



Minh họa một vài mẫu nhận diện sai

Các mẫu nhận diện sai này phần lớn đều chứa các font chữ lạ, các chữ được thiết kế riêng cho sách hoặc những chữ nằm lẩn trong hình minh họa (như chữ Nguyên tác) có nền khá khó nhìn nên khiến cho kết quả nhận diện chưa được tốt. Độ lệch màu trong hai nền của chữ Girls in love khiến cho nó không nhận dạng được chữ màu gì, chữ l bị hiểu nhầm thành chữ D do nó nhầm chữ màu trắng, chữ v nhận diện nhầm thành chữ m do nó dính với nền đen ở ngoài.

## V Đánh giá chung

**Công thức đánh giá bài toán:** Sử dụng thư viện [fuzzywuzzy](#) để so sánh khoảng cách giữa 2 chuỗi ( `fuzzywuzzy.fuzz.ratio()` ), với 3 tiêu chí đặt ra:

- Tương đồng 100%
- Tương đồng từ 95% trở lên
- Tương đồng từ 90% trở lên

**Đánh giá dựa trên f1 score:**

- Những thuộc tính thực tế có mang giá trị, dự đoán ra kết quả đúng => TP (True Positive)

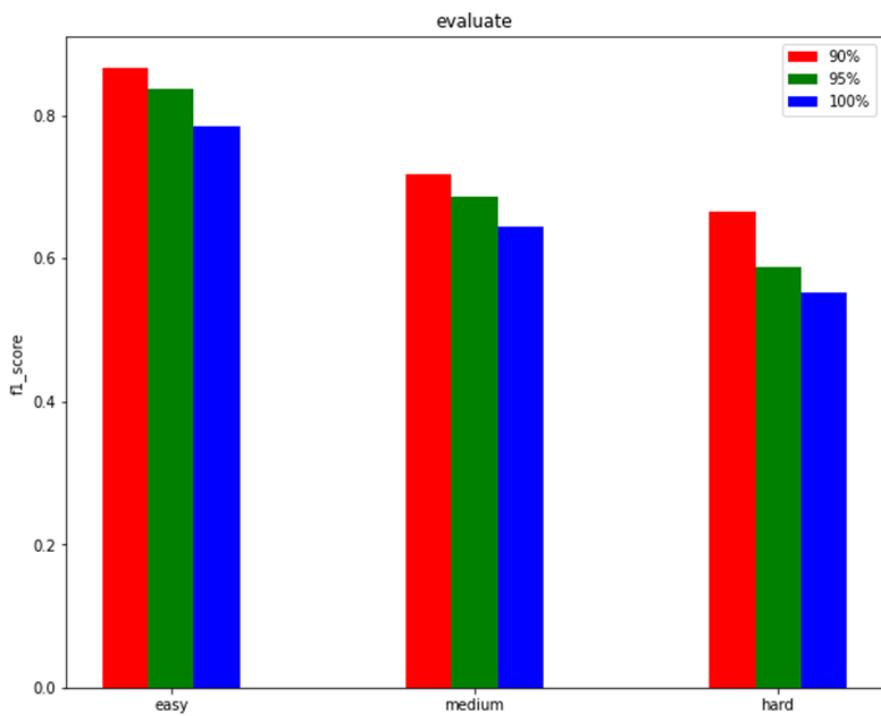
- Những thuộc tính thực tế không có, dự đoán cũng ra không có => TN (True Negative)
- Những thuộc tính thực tế không có nhưng dự đoán ra có => FN (False Negative)
- Những thuộc tính thực tế có mang giá trị nhưng dự đoán ra không có hoặc dự đoán ra kết quả sai => FP (False Positive)

**Phân chia đánh giá:** Tập dữ liệu 320 ảnh được chụp thực tế được nêu ở trên chưa được dùng qua để training YOLO hay VietOCR, ta chia thành 3 tập con:

- Easy: Vị trí thuộc tính và font chữ có thể chấp nhận được.
- Medium: Vị trí thuộc tính khó nhận dạng hay font chữ khó nhận dạng
- Hard: Cả vị trí thuộc tính và font chữ khó nhận dạng.

### Kết quả:

- Easy:
  - 100%: 0.78355
  - >= 95%: 0.83644
  - >= 90%: 0.86694
- Medium:
  - 100%: 0.64382
  - >= 95%: 0.68687
  - >= 90%: 0.71651
- Hard:
  - 100%: 0.55132
  - >= 95%: 0.58857
  - >= 90%: 0.66486

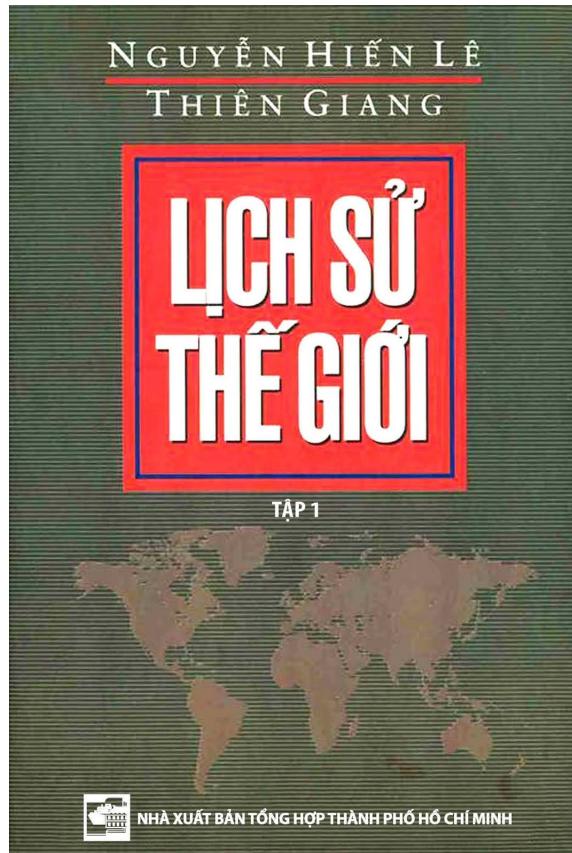


Thống kê đánh giá kết quả

Đây là **Colab Notebook Demo** của nhóm: Phần dưới cùng có phần đánh giá.  
Link data dùng để đánh giá: [Link input và ouput](#) của Dồ án số hóa tủ sách

### **Minh họa output của một vài ảnh bìa sách:**

Các model đều cho output khác tốt với những mẫu bìa sách có font chữ dễ nhìn và các vùng thông tin được bài trí rõ ràng, tách biệt nhau. Với các bìa sách có font chữ lạ, chữ được thiết kế riêng cũng như các thông tin trên bìa sách được bố trí phức tạp hoặc liền nhau thì output của các model cho kết quả không tốt lắm. Chẳng hạn như các mẫu bìa sách dưới đây:



Ảnh đưa vào test. Ảnh này không hề có trong train.

Tên sách: LỊCH SỬ THẾ GIỚI

Tên tác giả : THIÊN GIANG NGUYỄN HIẾN LÊ

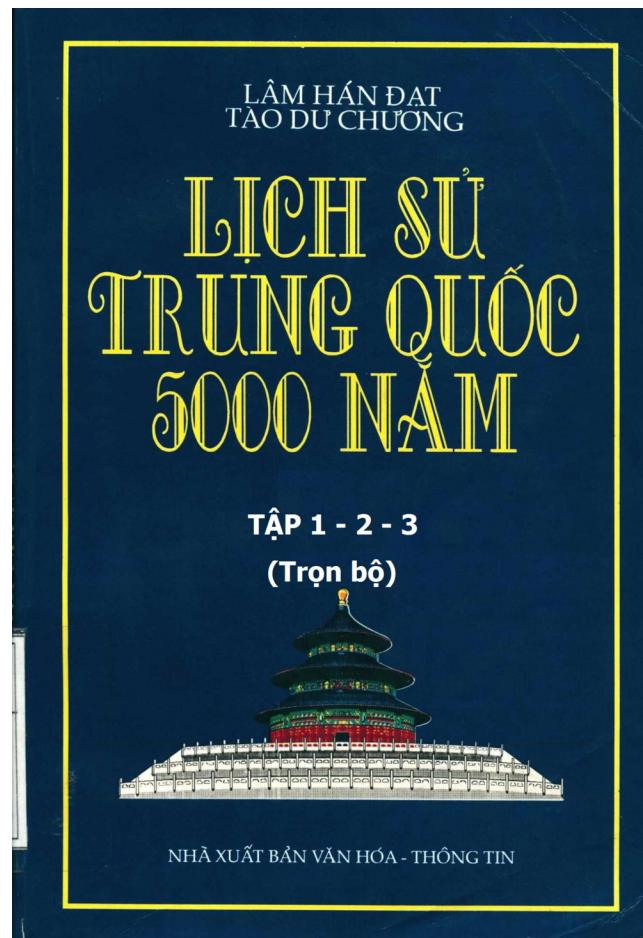
Nhà xuất bản: NHÀ XUẤT BẢN TỔNG HỢP THÀNH PHỐ HỒ CHÍ MINH

Tập: TẬP 1

Người dịch:

Tái bản:

Sách khá dễ nhìn và font chữ phổ biến nên việc nhận diện rất thuận lợi, các thành phần trên sách đều lấy ra rất chính xác



Tên sách: LỊCH SỬ TRUNG QUỐC 5000 NĂM

Tên tác giả : TÂM VĂN ĐỒNG

Nhà xuất bản: NHÀ XUẤT BẢN VĂN HÓA - THÔNG TIN

Tập:

Người dịch:

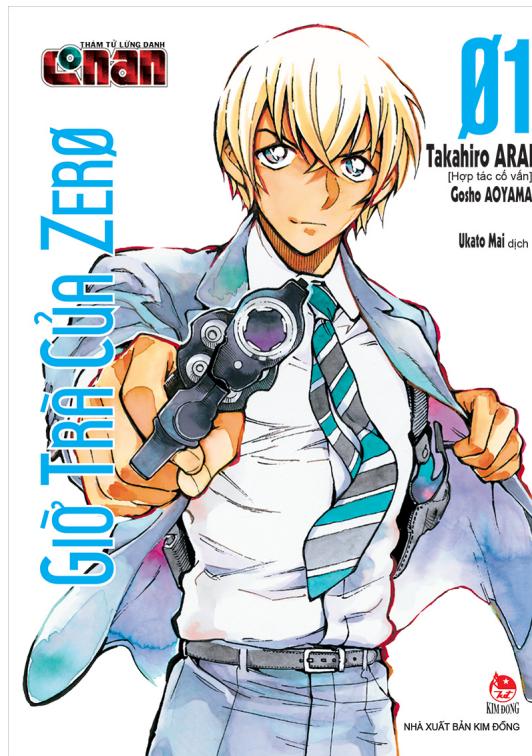
Tái bản:

Kết quả không quá tốt, thứ nhất là do dấu câu của phần tên sách quá nhỏ (đặc điểm của font chữ), tên tác giả thì bị lỗi, nhà xuất bản thì được trích ra rất chính xác và đầy đủ.



Tên sách: THỊNH GƯƠM  
Tên tác giả : KOYOHARU GOTOUGE  
Nhà xuất bản: NHÀ XUẤT BẢN KIM ĐỒNG NAI ĐỒNG  
Tập:  
Người dịch: Simirimi dịch  
Tái bản:

Tên tác giả và tên người dịch được trích ra chính xác, nhà xuất bản và tên sách bị nhận diện sai do font chữ đặc biệt.



Tên sách: GIỜ TRA CỦA ZERO THẨM TỬ LỪNG DANH

Tên tác giả :

Nhà xuất bản: NHÀ XUẤT BẢN KIM ĐỒNG

Tập: B1

Người dịch: Ukato Mai dịch

Tái bản:

Tên tác giả không xác định được. Trong phần tên sách thì bị thiếu từ "CONAN", từ này có font chữ rất đặc biệt nên khó nhận ra. Số tập là "01" nhưng font chữ đặc biệt bị nhầm diện sai thành "B1".

## VI Nhận xét

**Ưu điểm:** Có thể giảm bớt thời gian nhập liệu đối với quy mô lớn. Phép tính đơn giản cho 320 sách đã thu thập được:

- Thời gian chụp ảnh mất 1 giờ.
- Thời gian dự đoán kết quả mất 10 phút.
- Thời gian sửa chữa kết quả mất 2 giờ.

=> Nhanh hơn thực tế (10 giờ) gấp 3 lần.

**Nhược điểm:**

- Mô hình vẫn chưa tối ưu tốt cụ thể là phần craft\_text\_detection.
- Độ chính xác chưa được cao.
- Cần phần cứng mạnh (vẫn còn đang phụ thuộc vào phần cứng Colab)

**Đánh giá:** Mô hình vẫn chưa được tối ưu hóa 2 phần:

- Craft\_text\_detection đã nêu ở trên.
- Có một phần mềm có thể giúp dễ chỉnh sửa hơn và liên kết csdl để upload

# Chương 5: Ứng dụng và hướng phát triển

## I Ứng dụng

Hiện tại việc đưa thông tin vào máy tính giúp ích rất nhiều cho việc quản lý tủ sách nhưng việc nhập liệu bằng tay là rất mất thời gian . Như tên của bài toán số hoá tủ sách giúp ta số hoá một kho sách khổng lồ đưa vào trong máy tính để dễ quản lý, áp dụng cho những quy mô thư viện. Giúp giảm bớt lượng thời gian lớn cho công việc nhập liệu.

Đây là [Colab Notebook Demo ứng dụng của nhóm](#)

## II Hướng phát triển

- Theo ứng dụng đã nói trên cần cải tiến thêm những việc sau:
  - Tối ưu hoá độ chính xác với tập dữ liệu lớn hơn và training model CRAFT để phù hợp với chữ trên bìa sách.
  - Phát triển một phần mềm có thể giúp dễ chỉnh sửa hơn và liên kết csdl để upload.
- Ngoài ra, bài toán này còn giải quyết được một vấn đề khác nữa là chuyển đổi hình ảnh bằng văn bản, giúp việc tìm kiếm bằng hình ảnh (tìm kiếm sách) nhanh, và chính xác (3s cho một bìa sách). Ví dụ cụ thể là khi mượn sách của thư viện thì chỉ cần scan sách là sẽ hiện thông tin để mượn cuốn sách đó lên.
  - Tối ưu hoá về mặt thời gian, sử dụng model vừa phải, vừa nhanh nhưng vẫn đảm bảo kết quả tốt.
  - Văn nên cải thiện CRAFT.
  - Sửa lỗi chính tả (loại bỏ những từ sai chính tả).
  - Thay phát hiện các thuộc tính trong cuốn sách thành phát hiện vùng chữ quan trọng trong cuốn sách, có thể thay bằng các model nhanh hơn.

## TÀI LIỆU THAM KHẢO

- YOLOV5 của Ultralytics
- OpenCV-Python
- Craft-text-detector
- VietOCR
- fuzzywuzzy
- Character Region Awareness for Text Detection
- Introduction to YOLO Algorithm for Object Detection
- Transformer