

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



# **BÁO CÁO ĐỒ ÁN**

**MÔN HỌC: MÁY HỌC**  
**ĐỀ TÀI**  
**SỐ HÓA TỬ SÁCH**

**Giảng viên:** PGS.TS.Lê Đình Duy  
ThS.Phạm Nguyễn Trường An

**Sinh viên thực hiện:** Phan Anh Lộc - 19521766

Lê Đình Đức - 19521372

Lưu Anh Dũng - 19521392

**Lớp:** CS114.M11.KHCL

*Thành phố Hồ Chí Minh, tháng 1 năm 2022*

# MỤC LỤC

<b>GIỚI THIỆU ĐỒ ÁN.....</b>	<b>4</b>
<b>CHƯƠNG I. GIẢI TRÌNH CHỈNH SỬA SAU VẤN ĐÁP.....</b>	<b>6</b>
<b>CHƯƠNG II. TỔNG QUAN.....</b>	<b>7</b>
<b>1. Mô tả bài toán.....</b>	<b>7</b>
1.1 Ngữ cảnh ứng dụng: .....	7
1.2 Input và Output.....	7
1.3 Phương pháp giải quyết bài toán: .....	8
1.4 Các models mà nhóm sử dụng để giải quyết bài toán: .....	8
<b>2. Mô tả dữ liệu thu thập .....</b>	<b>8</b>
2.1 Data dành để train model YOLOv5 .....	9
2.2 Data dành để train model VietOCR .....	10
<b>CHƯƠNG III. GIỚI THIỆU CÁC MODEL THỰC HIỆN ĐỒ ÁN .....</b>	<b>11</b>
<b>1. Giới thiệu YOLO cho object detection.....</b>	<b>11</b>
1.1 Khái niệm: .....	11
1.2 Mô hình YOLO: .....	12
1.3 Loss function:.....	13
<b>2. CRAFT text detector cho Text localization .....</b>	<b>14</b>
2.1 Giới thiệu craft text detector .....	14
2.2 Kiến trúc network.....	15
<b>3. Giới thiệu VietOCR cho text recognition.....</b>	<b>16</b>
3.1 Giới thiệu mô hình VietOCR .....	16
3.2 Kiến trúc network.....	16
<b>CHƯƠNG IV. BỘ DỮ LIỆU.....</b>	<b>18</b>
<b>1. Xây dựng bộ dữ liệu.....</b>	<b>18</b>
1.1 Ảnh input.....	18
1.2 Data train cho model YOLOv5 .....	18
1.3 Data train cho model VietOCR .....	18
<b>2. Các mẫu dữ liệu khó .....</b>	<b>19</b>
2.1 Model YOLOv5 .....	19
2.2 Model VietOCR .....	21
<b>CHƯƠNG V. TRADING VÀ ĐÁNH GIÁ .....</b>	<b>23</b>
<b>1. Preprocessing.....</b>	<b>23</b>
<b>2. Object detection .....</b>	<b>23</b>

2.1 Chuẩn bị training data: .....	24
2.2 Đánh giá quá trình training.....	24
2.3 Đánh giá kết quả trên tập val.....	25
<b>3. Text localization .....</b>	<b>26</b>
<b>4. Text recognition .....</b>	<b>28</b>
4.1 Training .....	28
4.2 Quá trình training .....	28
4.3 Đánh giá .....	28
<b>5. Đánh giá chung.....</b>	<b>29</b>
<b>6. Nhận xét .....</b>	<b>33</b>
<b>CHƯƠNG VI. ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>34</b>
<b>BẢNG PHÂN CÔNG CÔNG VIỆC.....</b>	<b>35</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>35</b>

## GIỚI THIỆU ĐỒ ÁN

- Sách – kho tàng lưu trữ và truyền bá tri thức của nhân loại. Nó là thứ tài sản vô giá vẫn luôn được mọi người nâng lưu cất giữ cẩn thận đặc biệt là sách giấy. Hiện nay 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 cụ thể là sách điện tử e-book, audiobook. Nó đem lại nhiều sự tiện dụng hơn. Tuy nhiên sách giấy vẫn chiếm ngôi vị độc tôn, trải nghiệm nghiền ngẫm lật qua lật lại từng trang giấy là trải nghiệm mà không có bất cứ loại sách nào khác có thể thay thế được.
- Su 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

- Với một gia đình bình thường có khoảng 20-30 cuốn sách (chủ yếu là sách giáo khoa cho

con em đi học), với một gia đình có người hứng thú đọc sách thu thập sách nó lên đến hàng trăm, hàng nghìn cuốn. Với số lượng sách nằm trong khoảng này chúng ta vẫn có thể dễ dàng thông kê và sắp xếp. Tuy nhiên, việc này trở nên kinh khủng khi số lượng sách tăng lên hàng ngàn cuốn. Trong các nhà sách cũ, thư viện số lượng sách có thể lên tới hàng triệu cuốn.

- Thử tưởng tượng bạn là một nhân viên trong một thư viện, công việc của bạn chính là thống kê lại danh sách những cuốn sách mà thư viện đang lưu trữ và sắp xếp sách theo thứ tự nhất định để dễ dàng tìm kiếm cuốn sách mà mọi người muốn. Công việc vừa nghe thôi cũng đã muốn 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. 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 cho dù đơn giản nhưng việc phải lặp đi lặp lại hàng triệu lần, kéo dài cả tháng thì lại trở nên vô cùng nhàm chán, thậm chí dẫn đến sai sót.
- Để giải quyết bài toán 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. Với ứng dụng này việc thống kê sách sẽ trở nên tự động hóa. Việc bạn cần làm chỉ là chụp bìa trước sách từng quyển sách rồi bỏ vào ứng dụng và nó sẽ lưu lại cho ta vào danh sách với những thông tin cần thiết.

## CHƯƠNG I. GIẢI TRÌNH CHỈNH SỬA SAU VẤN ĐÁP

- [Phương pháp giải quyết bài toán](#)
- [Quy trình hoạt động của YOLOv5](#)
- [Các hàm loss function của YOLOv5](#)
- [Quy trình hoạt động của craft text detector](#)
- [Quy trình hoạt động của VietOCR](#)



## CHƯƠNG II. TỔNG QUAN

### 1. Mô tả bài toán

#### 1.1 Ngữ cảnh ứng dụng:

- Mô hình hướng tới người sử dụng là những người làm việc trong các thư viện, nhà sách cũ hay cả những tủ sách gia đình muốn thống kê lại những quyển sách đang được lưu trữ nhưng việc nhập liệu bằng tay quá tốn thời gian. Mục tiêu xây dựng một ứng dụng cho phép người dùng chụp bìa cuốn sách và chương trình máy học sẽ tự động nhận dạng chữ trên đó và nhập những thông tin quan trọng của quyển sách như tên, tác giả, tập, lần tái bản, nhà xuất bản vào danh sách để người dùng dễ dàng quản lý và sắp xếp.

#### 1.2 Input và Output

- Input: Ảnh chụp bìa trước của cuốn sách trên nền đen bằng camera



- **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:

- 1) Tên file ảnh chụp hình bìa sách
- 2) Tên sách
- 3) Tên tác giả (+ người minh họa)
- 4) Nhà xuất bản
- 5) Tập (số tập/phần của một cuốn sách dài kỳ)
- 6) Người dịch
- 7) Tái bản (số lần tái bản)

	file names	tên sách	tên tác giả	nhà xuất bản	tập	người dịch	tái bản
196	1.jpg	NHÀ VĂN VIỆT NAM		NHÀ XUẤT BẢN HỘI	TẬP 3		
199	10.jpg	GIẢI THÍCH NGỮ PH	MATHANHƯƠNG	NXB ĐÃ NẮNG			
58	100.jpg	DORAEMON VOL.23	Fujiko-F-Fujio				
57	101.jpg	KÍNH VẠN HOA	Nguyễn Nhật Ánh	KIM ĐỒNG		48	
56	102.jpg	KÍNH VẠN HOA	Nguyễn Nhật Ánh	KIM ĐỒNG		31	

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

### 1.3 Phương pháp giải quyết 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 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 thuộc vào để hoàn tất việc thu thập thông tin từ bìa sách.

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

- **Object detection:** sử dụng model **YOLOv5** cho bài toán Object Detection, YOLO được xem là phương pháp đầu tiên xử lý dữ liệu theo thời gian thực và đạt đến độ chính xác cao.
- **Text localization:** sử dụng model có sẵn trên pypi/craft-text-detector 0.4.2
- **Text recognition:** sử dụng model **VietOCR** cho bài toán text recognition chữ tiếng Việt

## 2. Mô tả dữ liệu thu thập

Ảnh input: gồm 236 ảnh bìa sách được chụp bằng camera smartphone trên nền đen với chất lượng ảnh tối thiểu là Full HD





Ảnh input được chụp bằng camera smartphone

Link data: [Tai đây](#)

## 2.1 Data dành để train model YOLOv5

- Mô tả: Gồm 7269 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 Đại học quốc gia TP HCM
  - ...
- Nhóm sử dụng online tool **makesense.ai** để dán nhãn 7269 ảnh bìa sách này, thu được 7269 file .txt chứa tọa độ các bounding boxes của các vùng có thông tin cần thu thập trên bìa sách.



Ảnh bìa sách crawl từ Nhà xuất bản Kim Đồng

Link data: [Tại đây](#)

Link label: [Tại đây](#)

## 2.2 Data dành để train model VietOCR

- Mô tả: gồm các data sau:
  - Hơn 30000 dòng text được cắt ảnh ra từ các ảnh bìa sách đã crawl được. Trong 30000 dòng text này, các thành viên đã dán nhãn được hơn 20241 dòng text.
  - 100000 dòng text đã được gán nhãn sẵn là data lấy từ GitHub của VietOCR.

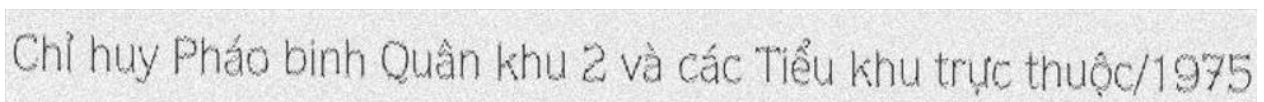

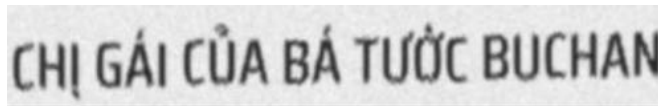
- Minh họa dữ liệu thu thập:

- Data mà nhóm label:



Các ảnh chứa text mà nhóm cắt ra từ 7000 sách

- Data có sẵn của VietOCR:

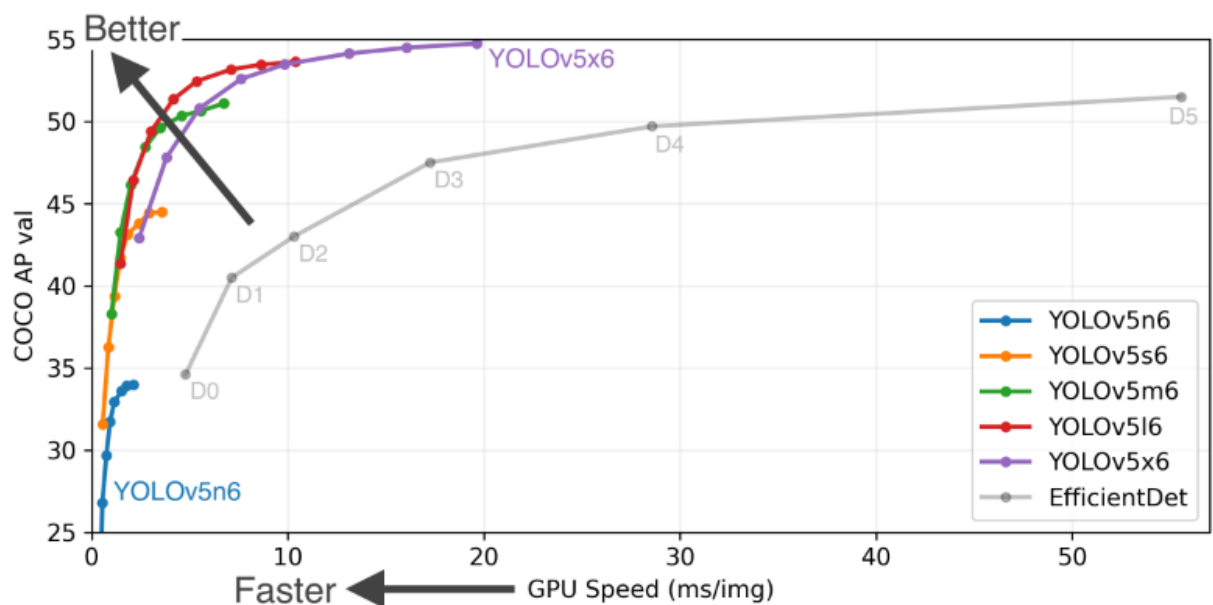


Link data: [Tại đây](#)

## CHƯƠNG III. GIỚI THIỆU CÁC MODEL THỰC HIỆN ĐỒ ÁN

### 1. Giới thiệu YOLO cho object detection

- YOLO-"You Only Look Once" một phương pháp phổ biến và được yêu thích cho các mới tìm hiểu về AI cũng như đang làm về AI. Nó luôn là ưu tiên hàng đầu khi giải quyết các bài toán về detection. Có nhiều phiên bản của YOLO được phát triển, trong chương trình này nhóm em sử dụng model **YOLOv5**. Phiên bản này khá tuyệt vời và vượt trội hơn tất cả các đàn anh trước đó.
- Bạn có thể nhìn thấy qua biểu đồ dưới đây:



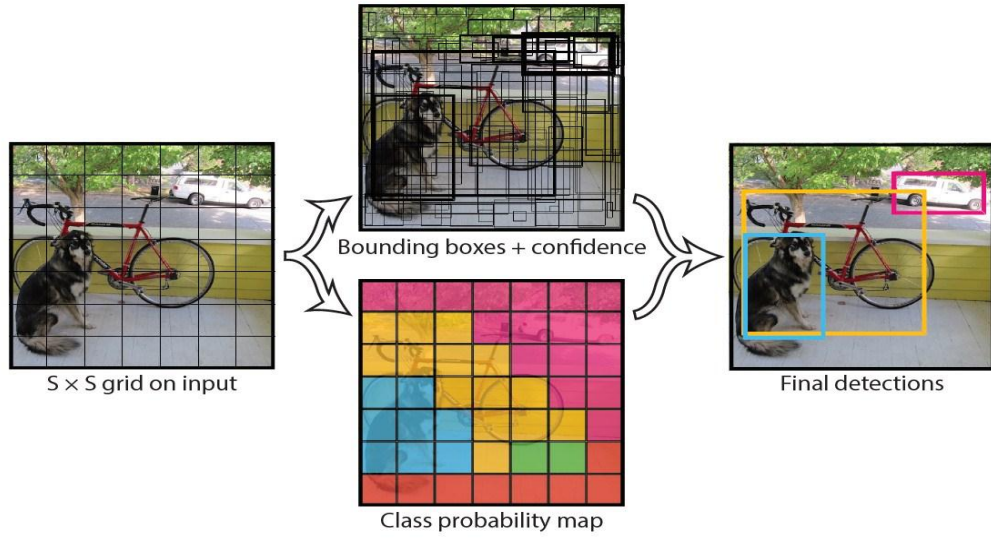
#### 1.1 Khái niệm:

- 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.
- Về cơ bản, YOLO chia ảnh đầu vào thành các ô nhỏ, mô hình sẽ dự đoán xác suất đối tượng

trong các đường bao (bounding-box) xung quanh mỗi ô nhỏ này. Những đường bao có xác suất cao sẽ được giữ lại và sử dụng cho nhiệm vụ xác định vị trí của đối tượng trong ảnh.

## 1.2 Mô hình YOLO:

- 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  $P(Object) * IoU_{pred}^{truth}$ . Trong đó,  $P(Object)$  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 Diện tích giao nhau của 2 bounding boxes / 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 rơi 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:



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

### 1.3 Loss function:

- YOLO sử dụng sum-squared error làm loss function vì đây là hàm để 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 chiều ngang và chiều cao giữa bounding box dự đoán và ground truth box.

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

- 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  $\mathbb{I}_{ij}^{obj}$ ).

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

- 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  $\mathbb{I}_i^{obj} = 0$ .

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

## 2. CRAFT text detector cho Text localization

### 2.1 Giới thiệu craft text detector

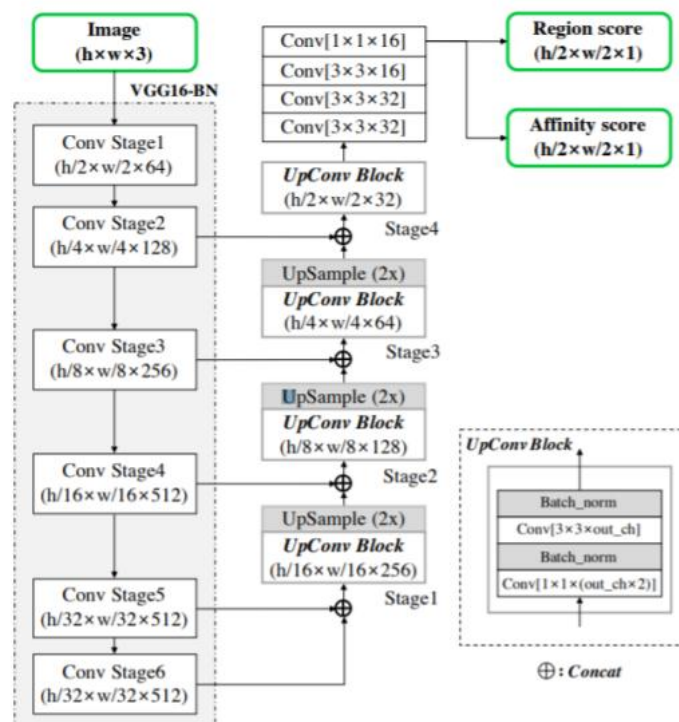
- Nhóm sử dụng model deep-learning có sẵn trên pypi/craft-text-detector 0.4.2: CRAFT: Character-Region Awareness For Text detection để thực hiện locate các text trên bìa sách. Đây là một PyTorch dùng cho craft text detection, nó detect được khá hiệu quả bằng cách tìm ra phân vùng của từng từ chữ cái và mối quan hệ giữ các chữ cái đó. Nó tạo ra hộp chữ nhật chứa các đoạn text dựa vào mối quan hệ giữa 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.



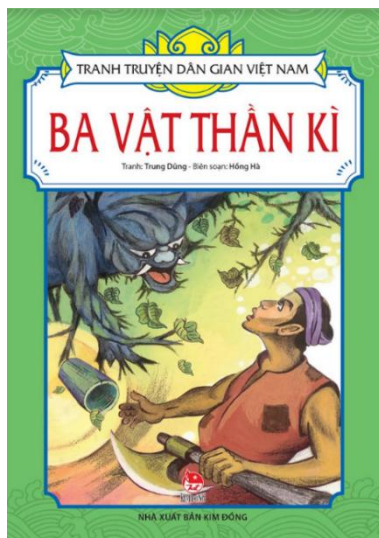
- Ứng dụng này bao gồm:
  - Input: ảnh cần nhận diện chữ
  - Output: một số ảnh đã được crop tự động sau khi nhận diện được

## 2.2 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



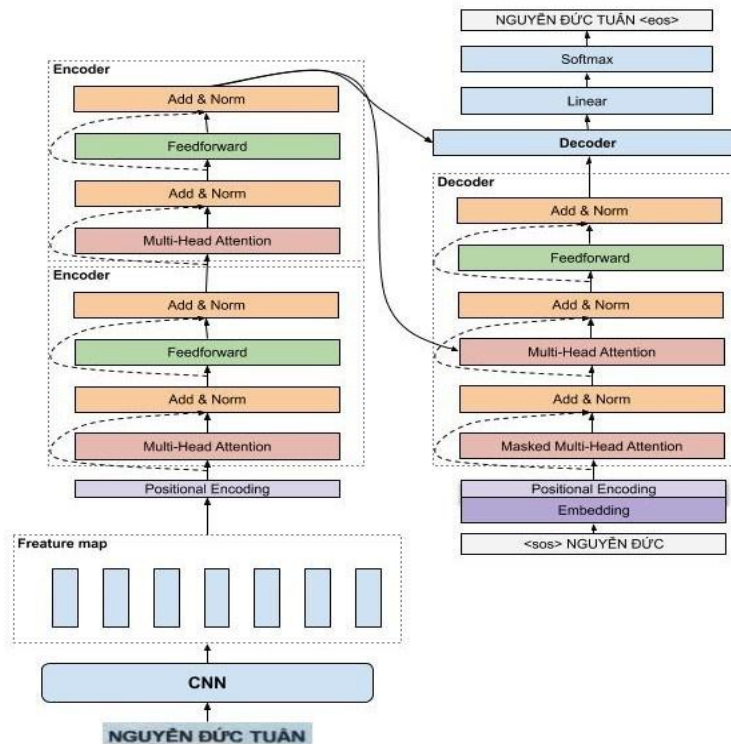
Kết quả

### 3. Giới thiệu VietOCR cho text recognition

#### 3.1 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. Trong ứng dụng này, nhóm em cài đặt mô hình Transformer OCR nhận dạng chữ viết tay, chữ đánh máy cho Tiếng Việt. Kiến trúc mô hình là sự kết hợp tuyệt vời giữa mô hình CNN và Transformer:

#### 3.2 Kiến trúc network



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

- 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,...
- Nhóm sẽ 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 attention.
- 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

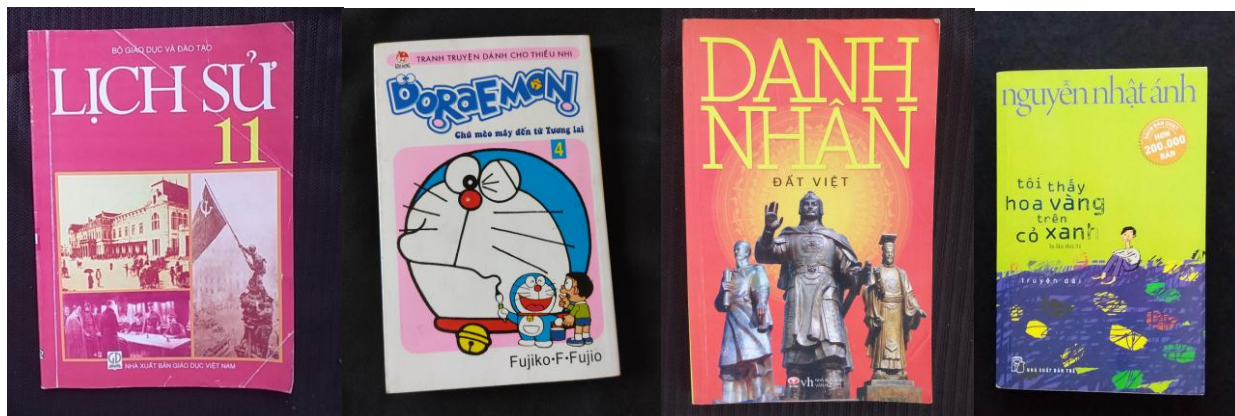
- Ta có thể thấy độ chính xác của transformer cao hơn nhưng thời gian predict lại lâu hơn.

## CHƯƠNG IV. BỘ DỮ LIỆU

### 1. Xây dựng bộ dữ liệu

#### 1.1 Ảnh input

- Các thành viên trong nhóm thống nhất với nhau chụp bìa sách trên nền đen để dễ áp dụng find contour cho bước scan ảnh bìa sách ra khỏi nền
- Các ảnh input thuộc nhiều thể loại khác nhau và được chụp dưới nhiều góc khác nhau một cách ngẫu nhiên, mỗi bức ảnh chỉ chụp **1 bìa sách**



Ví dụ một số ảnh input mà nhóm đã chụp

#### 1.2 Data train cho model YOLOv5

- Data cho model YOLOv5 gồm các ảnh rõ nét, không bị mờ và các kích thước tối thiểu là 560 x 720



#### 1.3 Data train cho model VietOCR

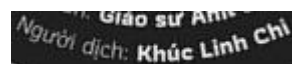
- Nhóm sẽ loại bỏ những hình chứa text đa dòng, hình không chứa text và hình chứa những kí

tự đặc biệt. Những hình này nhóm sẽ không gán nhãn.

- Một số mẫu nhóm gán nhãn:



Một số mẫu nhóm **không** gán nhãn:



## 2. Các mẫu dữ liệu khó

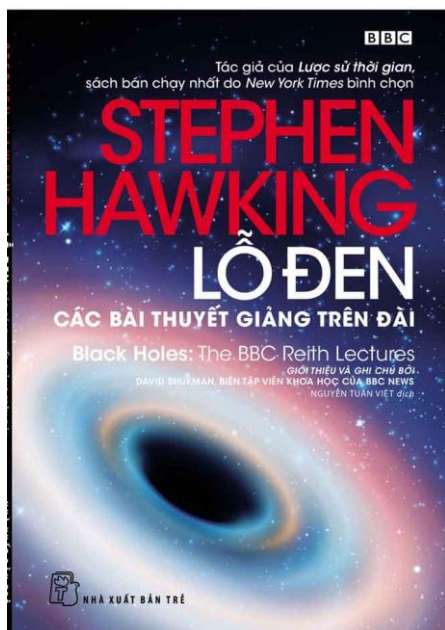
### 2.1 Model YOLOv5

- Các mẫu có tên tác giả nổi hơn tên sách (tên tác giả to hơn hoặc dài hơn tên sách)



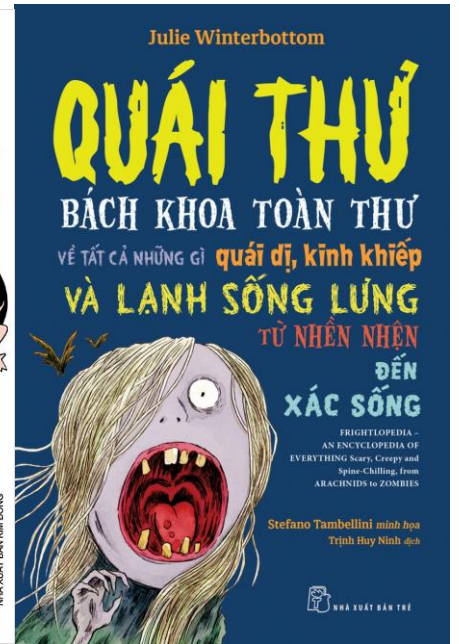
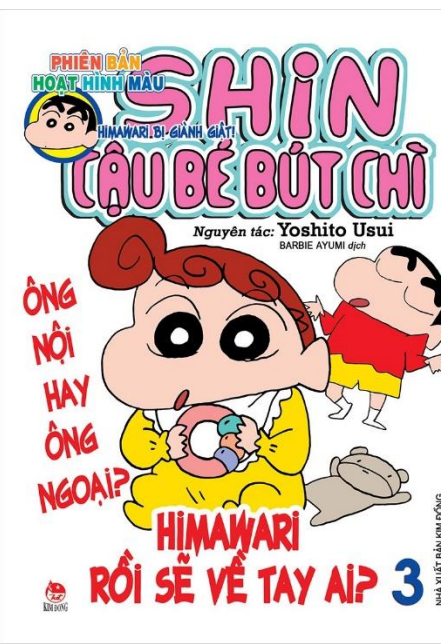


- Các mẫu có tên tác giả và tên sách có font chữ giống nhau và viết gần nhau



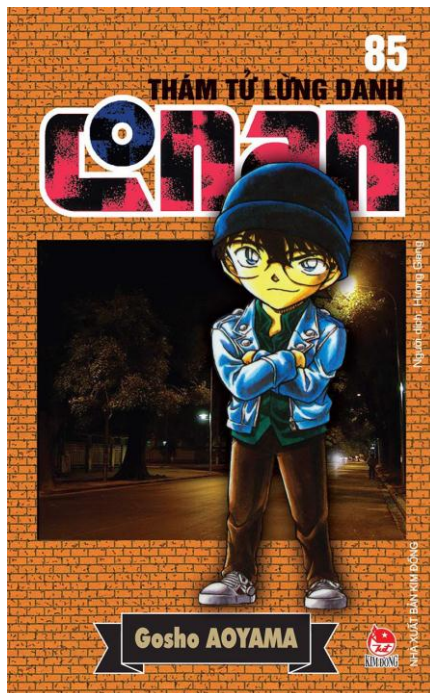
- Các mẫu có quá nhiều chữ hoặc các vật thể ngăn cách các dòng chữ trên bìa sách:



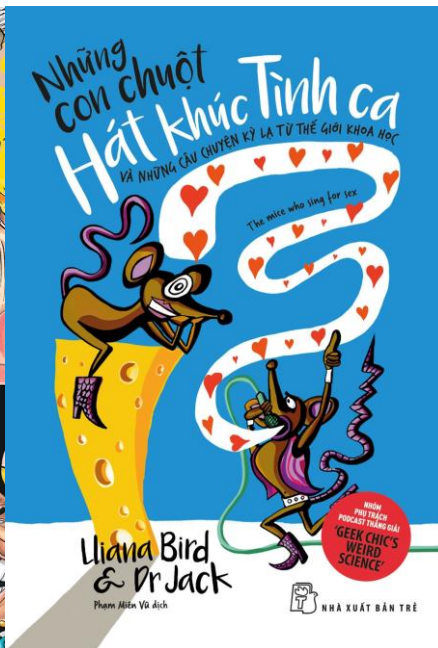
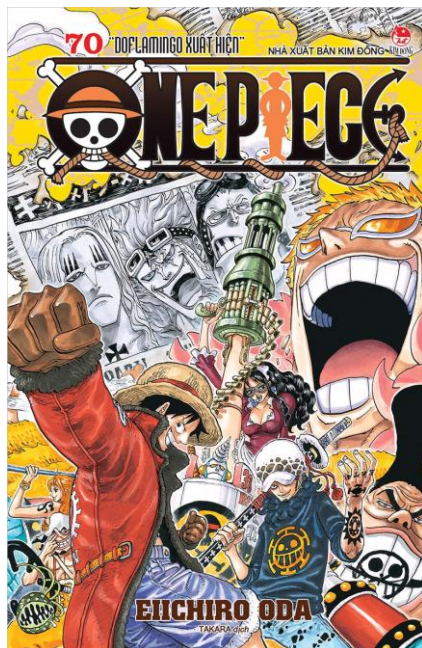


## 2.2 Model VietOCR

- Các mẫu có chữ bị che bởi các vật thể trên bìa sách:



- Các mẫu có font chữ lạ, viết kiểu, viết xéo, cong





## CHƯƠNG V.    TRADING VÀ ĐÁNH GIÁ

### 1. 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 dễ 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 bớt 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 đề ra được bìa sách ở góc chính diện.

### 2. Object detection

- Trong khâu Object detection, vì dữ liệu là bìa sách các Object đều chứa chữ nên việc nhằm

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. Đây là [colab](#) ghi nhận lại cách mà nhóm đã train YOLOv5.

## 2.1 Chuẩn bị training data:

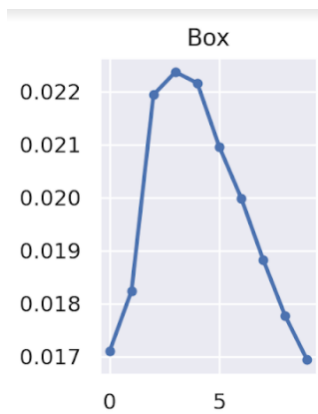
- Tổng cộng: 7269 ảnh bìa sách (do nhóm crawl) và file.txt (do nhóm dán nhãn). Trong đó:
  - Tập train: 6269 (Image - [Link](#), Label - [Link](#))
  - Tập val: 1000 (Image - [Link](#), Label - [Link](#))
- Cụ thể số label trong các file.txt của training và valid data như sau:

Label	Training data	Validation
Tên sách	6501	1001
Tên tác giả	6160	986
NXB	6488	1013
Tập	2652	93
Người dịch	4343	662
Tái bản	433	5
All	26577	3760

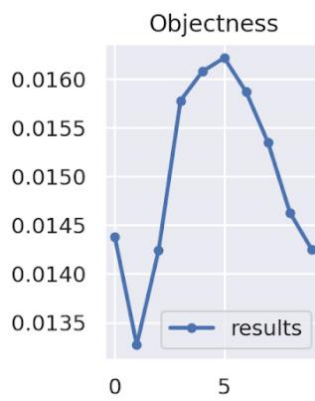
Trong đó số label của tên sách, tên tác giả và nxb chiếm số lượng nhiều nhất. Số label tái bản chiếm số lượng ít nhất

## 2.2 Đánh giá quá trình training

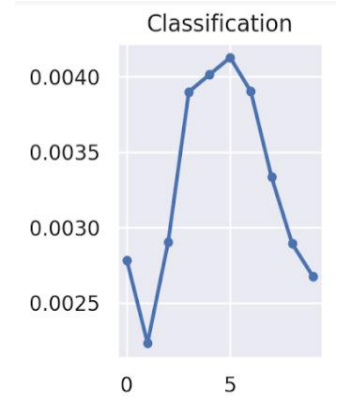
- Model sử dụng: YOLOv5x6 (extra large), 606 layers
- Batch size: 8
- Số lượng epoch train qua: 50 epoch
- Kết quả:
  - Loss



Localization loss



Confidence loss



Classification loss

- mAP0.5(all class): 0.9268

## 2.3 Đánh giá kết quả trên tập val

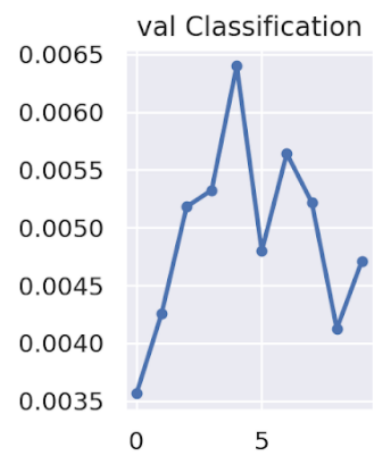
- Loss:



Localization loss

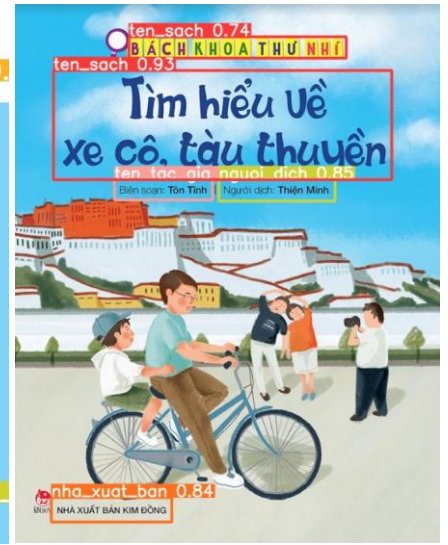
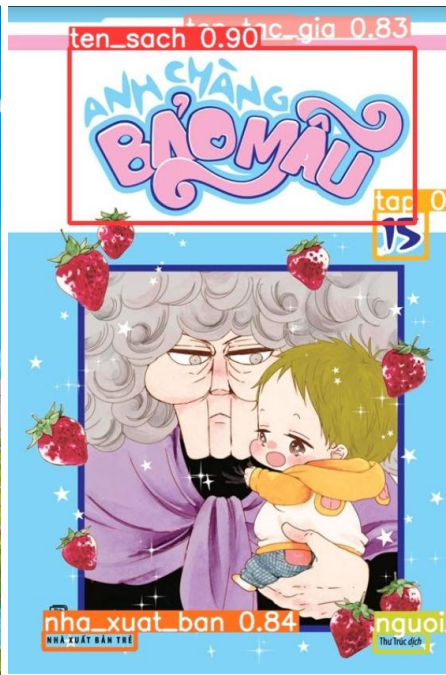


Confidence loss



Classification loss

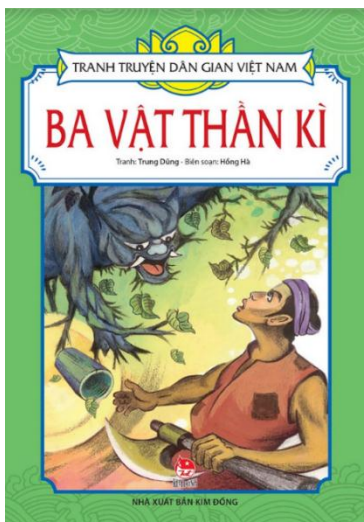
- mAP0.5: 0.935
- Giá trị mAP0.5 trên all class là 0.935 có thể nói là khá tốt
- Detect thử một vài tấm:



- Model detect đúng phần lớn các mẫu. Các mẫu detect đúng bao gồm từ các mẫu đơn giản chỉ gồm vài label cho đến những mẫu phức tạp với nhiều label hơn

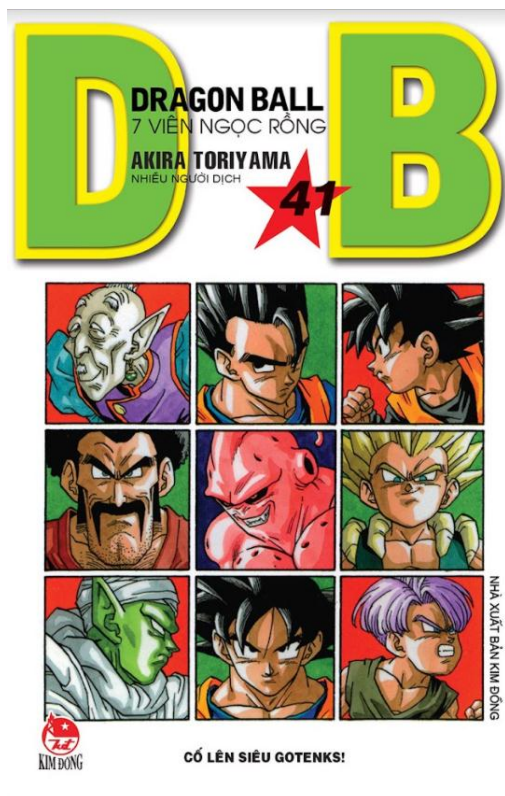
### 3. Text localization

- Sau khi đã train model yolov5 để detect được các thành phần trên một cuốn sách, việc còn lại 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 0.4.2: [CRAFT: Character-Region Awareness For Text detection](#)
- Model này nhóm chỉ sử dụng mô hình đã pretrain để tiết kiệm thời gian:

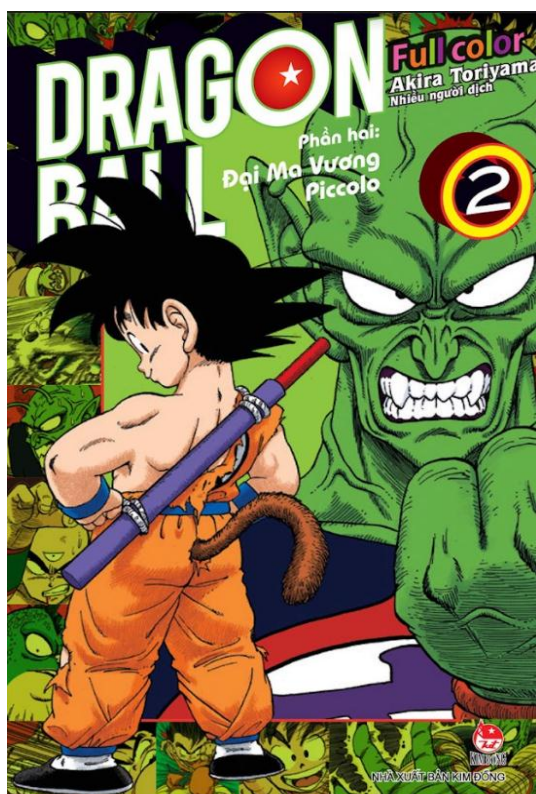


- Phông chữ đơn giản, dễ nhìn thì craft vẫn ra kết quả tốt





- Với những mẫu có chữ đề lên chữ như trên thì craft không được tốt lắm



- Mẫu này thì bị craft sót mất 1 phần tên sách do bị che

## 4. 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 là thư viện có sẵn VietOCR
- Đây là [colab](#) mà nhóm đã thực hiện

### 4.1 Training

- Tổng cộng 20241 ảnh chứa dòng text từ bìa sách và 100000 ảnh chứa dòng text đã được gán nhãn sẵn. Trong đó:
  - Training data: 16218 ảnh được nhóm gán nhãn + 100000 ảnh đã được gán nhãn sẵn
  - Validation: 4023 ảnh được nhóm gán nhãn

### 4.2 Quá trình training

- Iter: 10000
- Device: 'cuda:0'
- Thời gian train: Gần 4 tiếng
- Đối với các dòng text dọc thì ta sẽ xoay 90 độ theo chiều kim đồng hồ và ngược chiều kim đồng hồ. Từ đó ta sẽ có 3 ảnh: ảnh gốc, ảnh xoay trái và xoay phải. Ta tiếp tục dự đoán chữ trên 3 ảnh đó và lấy ra kết quả dự đoán có điểm dự đoán cao nhất.

### 4.3 Đánh giá

- Training loss: 0.544
- Valid loss: 0.562 – acc full seq: 0.878 – acc per char: 0.9644
- Tập train precision đạt: 0.96482104
- Tập valid precision đạt: 0.873114224137931

87% có thể nói là khá tốt, tuy nhiên thì vẫn còn sai sót, đặc biệt là các font chữ đặc biệt. Với các font chữ đơn giản, cách bố trí rõ ràng, dễ nhìn thì model nhận diện khá tốt

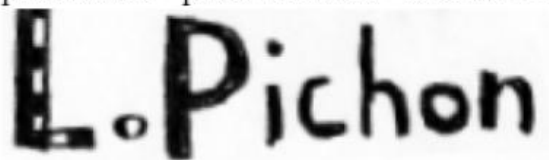
prob: 0.925 - pred: đều đặn! - actual: đều đặn!



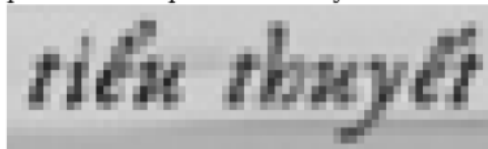
prob: 0.916 - pred: sao xẹt - actual: sao xẹt



prob: 0.926 - pred: L.Pichon - actual: L.Pichon

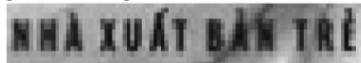


prob: 0.931 - pred: tiểu thuyết - actual: tiểu thuyết



- Có thể thấy model nhận diện khá tốt các mẫu trên. Mặc dù ảnh có độ phân giải thấp nhưng model vẫn nhận diện đúng với prob rất cao

prob: 0.829 - pred: ĐẾN HÀ XUẤT BẢN TRẺ - actual: NHÀ XUẤT BẢN TRẺ



prob: 0.795 - pred: BÙ CHÍ VINH - actual: BÙI CHÍ VINH



prob: 0.919 - pred: Nhalno - actual: Nhà trọ



prob: 0.917 - pred: Lilidolle - actual: Lilidoll



prob: 0.898 - pred: CÙNG NGẠI NẾU BẠN KHÔNG GIỎI TIẾNG ANH - actual: ĐỪNG NGẠI NẾU BẠN KHÔNG GIỎI TIẾNG ANH



- Các mẫu nhận diện sai trên phần lớn là do font chữ đặc biệt, các chữ được thiết kế riêng cho sách hoặc những chữ nằm lẫn trong hình minh họa. Một số mẫu bị nhận diện sai do dòng text bị viết xéo, cong

## 5. Đánh giá chung

- Để đánh giá kết quả dự đoán với kết quả thực, nhóm sử dụng thư viện fuzzywuzzy, cụ thể là fuzzywuzzy.fuzz.ratio(). Thay vì cố gắng định dạng các chuỗi để khớp với nhau, Fuzzywuzzy sử dụng một số tỷ lệ tương tự giữa hai chuỗi và trả về tỷ lệ phần trăm tương tự. Với 5 tiêu chí:

- Tương đồng 80%
- Tương đồng 85%
- Tương đồng 90%
- Tương đồng 95%
- Tương đồng 100%

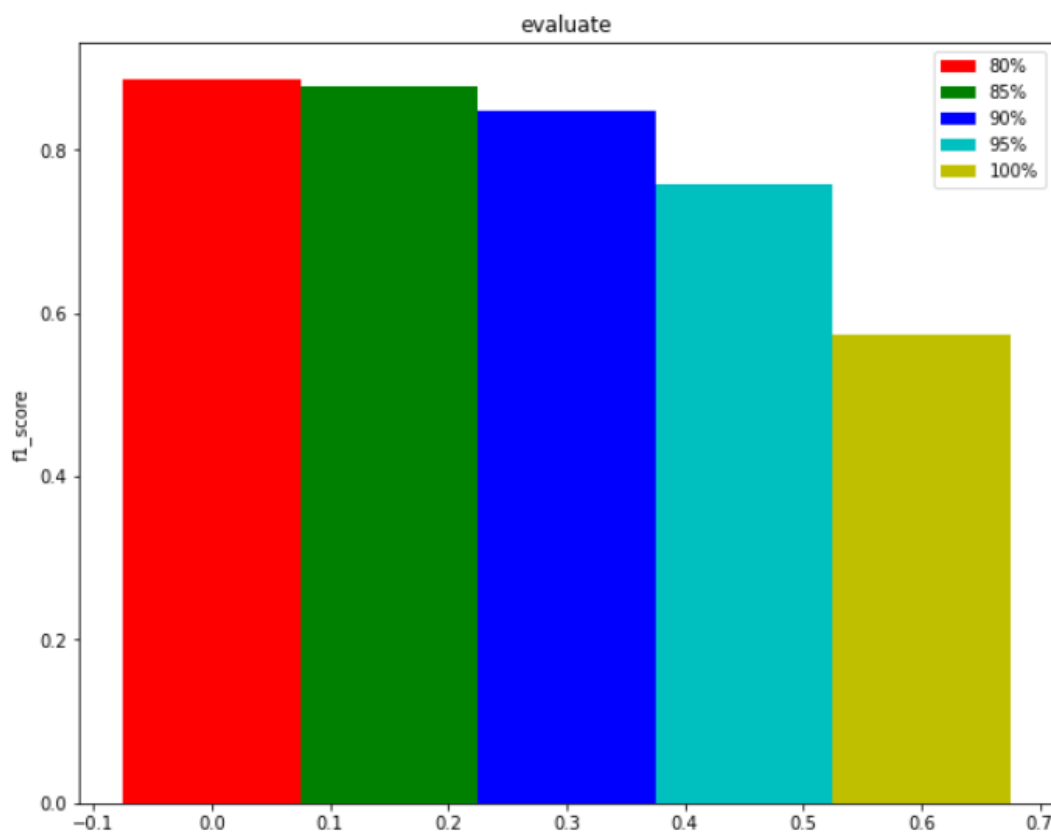
- Cách đánh giá: Dựa trên F1-score, với:

- TP (True positive): Biểu thị những thuộc tính thực tế có mang giá trị, dự đoán có mang giá trị và đúng
- TN (True negative): Biểu thị những thuộc tính thực tế không mang giá trị, dự đoán không

mang giá trị

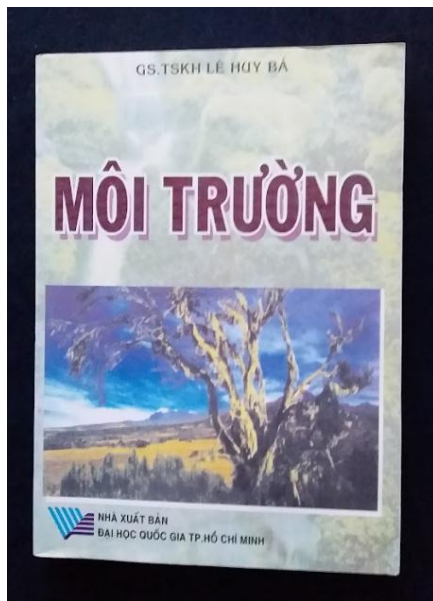
- FP (False positive): Biểu thị những thuộc tính thực tế có mang giá trị, dự đoán không mang giá trị hoặc sai
  - FN (False negative): Biểu thị những thuộc tính thực tế không mang giá trị, dự đoán mang giá trị
- Đánh giá: Tập dữ liệu gồm 236 ảnh được chụp thực tế chưa hề được dùng để training YOLOv5 hay OCR:

- $\geq 80\%$ : 0.8874493927125505
- $\geq 85\%$ : 0.8774509803921569
- $\geq 90\%$ : 0.8482816429170159
- $\geq 95\%$ : 0.7588075880758807
- 100%: 0.5746887966804979



Biểu đồ thống kê đánh giá kết quả

- Đây là [colab](#) mà nhóm đã thực hiện
- Demo output một vài tấm:



Tên sách: MÔI TRƯỜNG  
 Tên tác giả: GS.TSKH LÊ HUY BÀ  
 NXB: NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
 Tập:  
 Người dịch:  
 Tái bản:

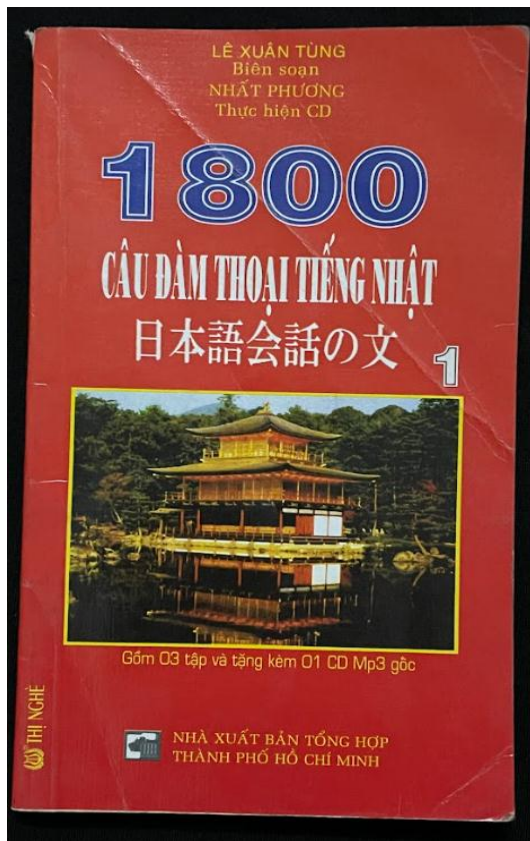
- Với những cuốn dễ nhìn và font chữ phổ biến như hình trên thì việc nhận diện rất tốt, các thành phần trong sách đều được lấy ra rất chính xác



Tên sách: Cách mạng công nghiệp lần thứ tư  
 Tên tác giả: Klaus Schwab  
 NXB:  
 Tập:  
 Người dịch:  
 Tái bản:

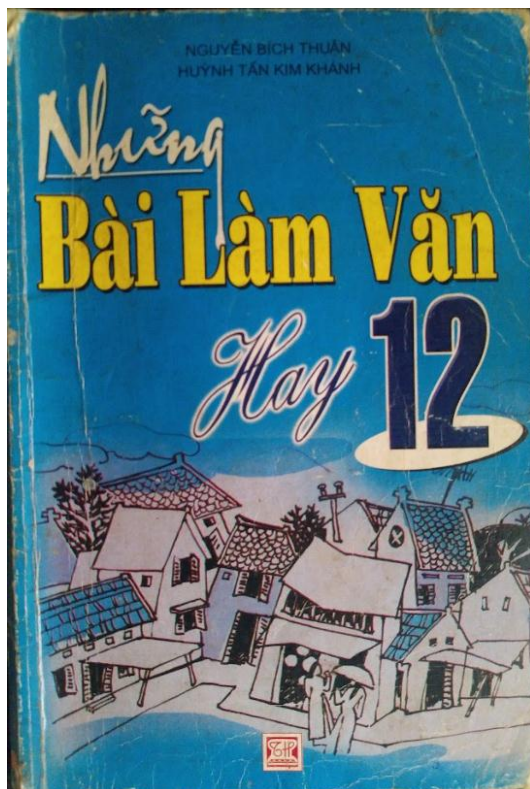
- Đối với hình trên, model nhận diện sai tên sách do dấu của tên sách quá nhỏ





Tên sách: 1 30 CÂU ĐÀM THOẠI TIẾNG NHẬT  
 Tên tác giả:  
 NXB: NHÀ XUẤT BẢN TỔNG HỢP THÀNH PHỐ HỒ CHÍ MINH  
 Tập:  
 Người dịch:  
 Tái bản:

- Đối với hình này thì model không nhận diện được tên tác giả và nhận diện sai tên sách



Tên sách: Nhưng Văn Bài Làm  
 Tên tác giả: NGUYỄN BÍCH THUẬN HUỲNH TÂN KIM KHÁNH  
 NXB:  
 Tập:  
 Người dịch:  
 Tái bản:

- Đối với mẫu trên thì do font chữ đặc biệt nên model nhận diện sai tên sách



## 6. Nhận xét

- Ưu điểm: Có thể giảm bớt thời gian nhập liệu đối với quy mô lớn
- Nhược điểm:
  - Do thời gian có hạn nên nhóm không tiến hành training model Craft nên kết quả không được tốt.
  - Độ chính xác chưa được cao
  - Phụ thuộc vào phần cứng của Colab quá nhiều, cần phần cứng mạnh
  - Việc thu nhập data test vẫn còn hạn chế do ảnh hưởng của dịch Covid và không có nhiều thời gian
- Nhìn chung thì nhóm quản lý thời gian chưa được tốt dẫn tới kết quả đồ án không được như mong muốn, nhưng vẫn hoàn thành được những yêu cầu tối thiểu trong đồ án này

## CHƯƠNG VI. ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN

### 1. Ứng dụng:

- Với nhu cầu đọc sách của mọi người ngày càng đa dạng, đòi hỏi các thư viện cần có một hệ thống để lưu trữ cũng như quản lý sách với số lượng nhiều hơn. Lúc này, các hệ thống quản lý thư viện số ra đời.
- Các hệ thống này đã phát huy tốt vai trò quản lý sách trong thư viện và nhà sách dựa trên những thông tin cơ bản của một cuốn sách như tên sách, tác giả, nhà xuất bản,... Tuy nhiên, một trong số những nhược điểm của đa số các hệ thống quản lý thư viện ngày nay là các thông tin của sách chỉ có thể nhập bằng phương pháp thủ công, tức là sử dụng nhân công để nhập liệu. Điều này sẽ rất tốn thời gian và nhân lực nếu như số lượng sách cần nhập liệu có thể lên đến hàng trăm hoặc thậm chí hàng nghìn cuốn.
- Mô hình máy học của bọn em có thể khắc phục được nhược điểm này bằng cách quét một hình ảnh chụp một bìa sách và trả về những thông tin cơ bản của cuốn sách đó được in trên bìa sách. Cách làm này mang lại đồng thời nhiều lợi ích, nhất là đối với người sử dụng thư viện có thể dễ dàng sử dụng các thiết bị máy tính, điện thoại thông minh để tìm kiếm, truy cập, sử dụng tri thức một cách rộng rãi, nhanh gọn.

### 2. Hướng phát triển:

- Trong đồ án này dữ liệu mà nhóm dùng để huấn luyện mô hình chỉ là một phần nhỏ trong số vô vàn sách đang được lưu hành và lưu trữ tại các nhà sách và thư viện. Vì vậy nên nhóm sẽ cố gắng cải thiện code và thu thập thêm dữ liệu đa dạng và phong phú hơn để có thể xử lý nhiều cuốn sách hơn với độ chính xác cao. Có thể nhóm sẽ tìm kiếm những phương pháp khác để giải quyết vấn đề này.
- Nếu có thể, nhóm sẽ phát triển mô hình trở thành một ứng dụng để có thể sử dụng hoặc tích hợp với các hệ thống quản lý sách hiện đang được sử dụng tại các thư viện và nhà sách.

## BẢNG PHÂN CÔNG CÔNG VIỆC

Họ tên	Mô tả công việc	Đánh giá
Phan Anh Lộc	Crawl data phần Object detection, gán label, làm file báo cáo pdf, làm code & chạy code phần Object detection, text recognition và đánh giá chung đồ án, lên ý tưởng	90%
Lê Đình Đức	Thu nhập data và chạy code phần text localization, làm file powerpoint, gán label	90%
Lưu Anh Dũng	Thu nhập data và chạy code phần text localization, làm file powerpoint, gán label	90%

## TÀI LIỆU THAM KHẢO

- <https://pypi.org/project/craft-text-detector/>
- <https://github.com/pbcquoc/vietocr>
- <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>
- <https://pbcquoc.github.io/transformer/>
- <https://viblo.asia/p/transformers-nguoi-may-bien-hinh-bien-doi-the-gioi-nlp-924IJPOXKPM>
- <https://github.com/ultralytics/yolov5>
- <https://www.kaggle.com/ultralytics/yolov5>
- <https://viblo.asia/p/yolov5-detect-lua-mi-chi-trong-vai-phut-GrLZDawglk0>