



**ĐỒ ÁN CUỐI KỲ**  
CS532.M21.KHCL  
GVHD: ThS. Đỗ Văn Tiến

**TRÍCH XUẤT THÔNG TIN BÌA SÁCH**

Ngày 2 tháng 7 năm 2022

**Nhóm thực hiện:**  
Lương Phạm Bảo 19521242  
Nguyễn Gia Thông 19520993  
Phạm Ngọc Dương 19521412

# MỤC LỤC

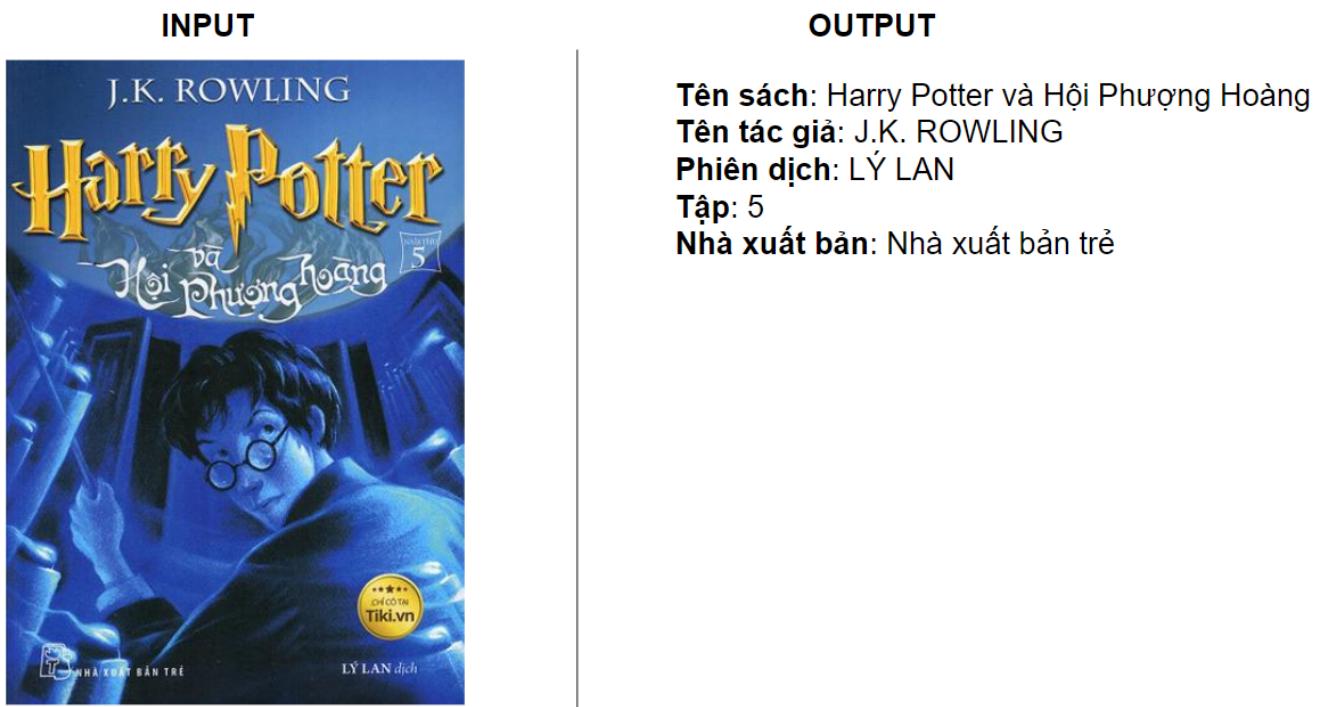
<b>1 Giới thiệu</b>	<b>2</b>
1.1 Dataflow . . . . .	2
1.2 Thách thức . . . . .	2
<b>2 Phương pháp đề xuất</b>	<b>3</b>
2.1 Hướng tiếp cận đề xuất ban đầu . . . . .	3
2.2 Hạn chế của hướng tiếp cận thứ nhất . . . . .	4
2.3 Hướng tiếp cận đề xuất tiếp theo . . . . .	5
<b>3 Giới thiệu bộ dữ liệu</b>	<b>5</b>
3.1 Ảnh Input đầu vào . . . . .	5
3.2 Object Detection - YOLOv5 . . . . .	6
3.3 Text Recognition - VietOCR . . . . .	6
<b>4 Giới thiệu các Model</b>	<b>7</b>
4.1 Giới thiệu về Model YOLO cho object detection . . . . .	7
4.1.1 Cấu trúc mạng của YOLO . . . . .	7
4.1.2 Cách YOLO hoạt động . . . . .	8
4.1.3 Loss Function . . . . .	8
4.2 Các Model sử dụng cho Text Localization . . . . .	9
4.2.1 Paddle OCR . . . . .	9
4.2.2 ABCNet . . . . .	10
4.2.3 CRAFT Text Detector . . . . .	10
4.3 Text Recognition-VietOCR . . . . .	11
4.3.1 Giới thiệu mô hình VietOCR . . . . .	11
4.3.2 Kiến trúc Network . . . . .	12
<b>5 Thiết kế hệ thống</b>	<b>13</b>
<b>6 Đánh giá pipeline</b>	<b>13</b>
6.1 Giới thiệu các độ đo đánh giá mô hình . . . . .	13
6.1.1 Intersection over Union (IoU) . . . . .	13
6.1.2 True/False Positive/Negative . . . . .	13
6.1.3 Precision . . . . .	14
6.1.4 Recall . . . . .	14
6.1.5 Average Precision (AP) . . . . .	14
6.1.6 Mean Average Precision (mAP) . . . . .	14
6.1.7 Accuracy sequence và character . . . . .	14
6.2 Đánh giá chung . . . . .	15
6.2.1 Tốc độ xử lý . . . . .	15
6.3 Kiến trúc hệ thống . . . . .	16
6.3.1 Tổng quan . . . . .	16
6.3.2 Chi tiết . . . . .	17
6.3.3 Giao diện hệ thống . . . . .	17
6.3.4 Screenflow . . . . .	17
6.4 Demo . . . . .	18
<b>7 Cài đặt &amp; Kiểm thử</b>	<b>19</b>
<b>8 Kết luận</b>	<b>19</b>
8.1 Pros . . . . .	19
8.2 Cons . . . . .	19
8.3 Kết luận . . . . .	19

# 1 Giới thiệu

Ngày nay theo sự phát triển CNTT nói chung và AI nói riêng, việc chúng ta chuyển đổi, số hóa thông tin đã trở thành nhu cầu cấp bách, thiết yếu nhất đối với xã hội ngày nay. Trên thực tế có rất nhiều ứng dụng số hóa thông tin như : nhận diện biển số xe, hỗ trợ đọc văn bản cho người khiếm thị, tự động rút trích thông tin các giấy tờ tùy thân như chứng minh thư, CCCD... Với việc các thư viện lớn của các viện nghiên cứu ,trường đại học ngày càng có nhiều tài liệu hơn, việc trích xuất thông tin từ các tài liệu, thông tin từ bìa sách trở thành yêu cầu cấp thiết để phục vụ cho việc quản lý, chuyển đổi lên các thư viện sách điện tử....

## 1.1 Dataflow

- Đầu vào: Ảnh bìa sách.
- Đầu ra: Các thông tin trên bìa sách của ảnh: Tên sách, Tên tác giả, phiên dịch, Tập, Nhà xuất bản,Tái bản .



**Hình 1:** Mô tả bài toán

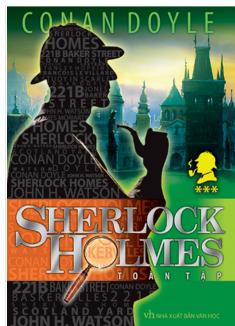
## 1.2 Thách thức

- Một trong những thách thức đối với bài toán này là các dạng Art-text, trên thực tế các nhà xuất bản luôn tìm cách để bìa sách họ phát hành trong đẹp mắt và cuốn hút người đọc. Chính vì thế đa số các bìa sách hiện nay đều có các dạng chữ được cách điệu và điều đó sẽ ảnh hưởng đến mô hình nhận diện chữ.
- Vị trí của các trường thông tin không cố định. Không có một chuẩn mực nào cho việc đặt vị trí của các trường thông tin. Các nhà thiết kế bìa sách tùy vào mục đích mà đặt các thông tin ở vị trí khác nhau gây khó khăn cho việc xác định vị trí của chúng.
- Ngoài ra, các bìa sách còn chứa các thông tin gây nhiễu, những thông tin không thật sự cần thiết.

## Art Text



Nhiều thông tin  
gây nhiễu



## Đa dạng vị trí các trường thông tin



## 2 Phương pháp đề xuất

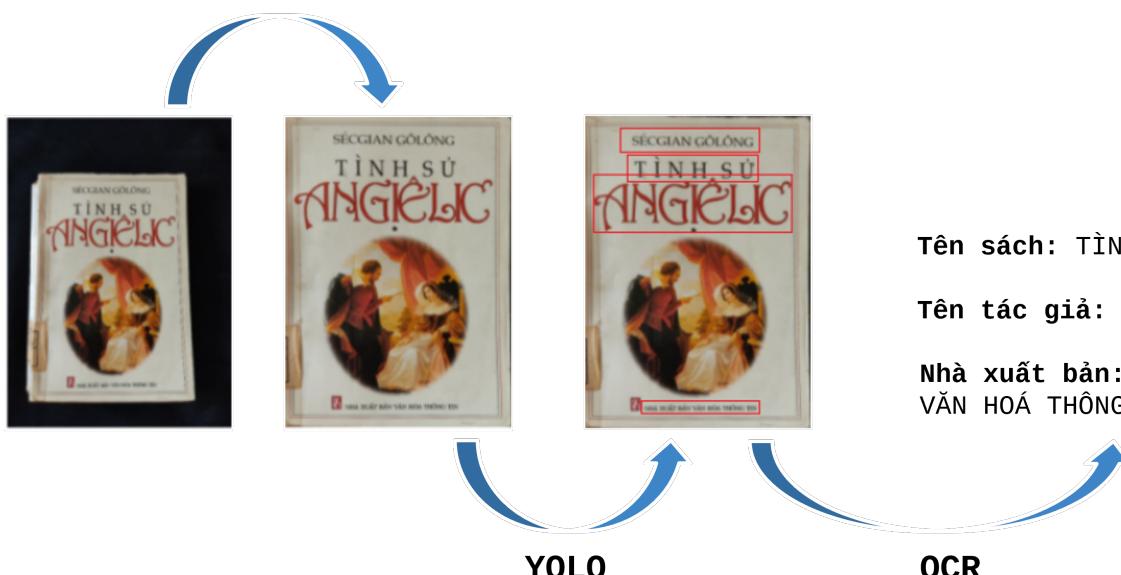
### 2.1 Hướng tiếp cận đề xuất ban đầu

**Preprocessing:** Ở bước này là giai đoạn tiền xử lý nhằm loại bỏ các vùng không thuộc sách nhằm giảm thiểu các yếu tố nhiễu để tăng hiệu suất huấn luyện

**Object detection:** Ở bước này, ứng với mỗi trường thông tin ta có thể xem như là một đối tượng cần nhận diện và áp dụng một số phương pháp tiêu biểu cho bài toán Object detection như Yolov4, Yolov5,...

**Text recognition:** Sau khi crop các vùng chứa text thành các ảnh chứa text, các bức ảnh trên sẽ được đưa qua mô hình nhận diện chữ như VietOCR, SRN, DeepText,...

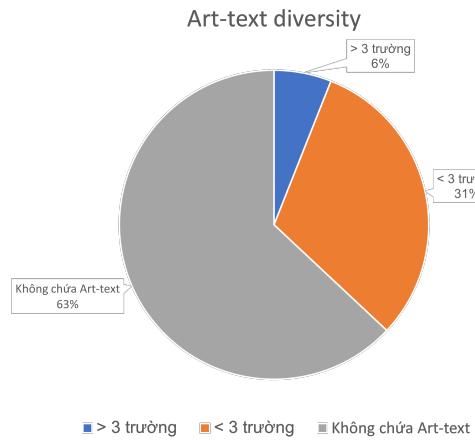
### Preprocess



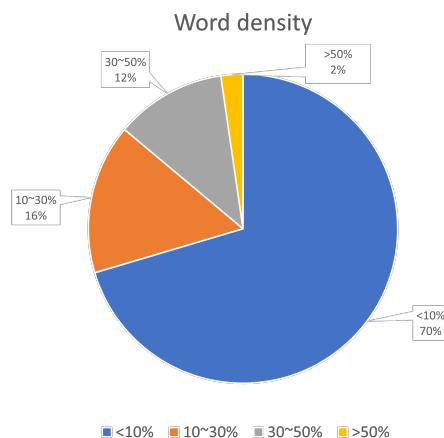
Tên sách: TÌNH SỬ ANGIÊLIC

Tên tác giả: SÉCGIAN GÔLÔNG

Nhà xuất bản: NHÀ XUẤT BẢN  
VĂN HÓA THÔNG TIN



**Hình 2:** Thống kê số lượng ảnh có Art text



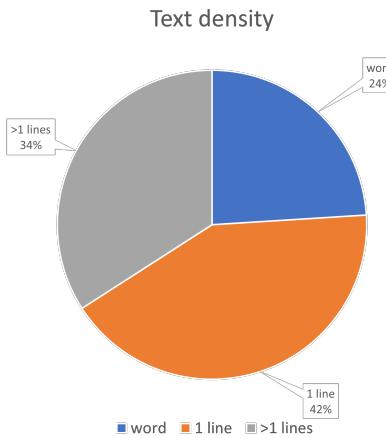
**Hình 3:** Thống kê ảnh dựa trên diện tích text

Việc nhóm không sử dụng trực tiếp mô hình text detection trên ảnh Input là bởi vì theo nhóm thống kê có khá nhiều ảnh bìa sách có chứa Art text (các chữ trang trí, có hình dạng đặc thù và phức tạp) cũng như là một số các ảnh có chứa nhiều text không cần thiết như thống kê từ bộ dữ liệu của nhóm nên nhóm quyết định sẽ sử dụng mô hình Object detection như trên để lọc bỏ các vùng không chứa các trường thông tin và có thể detect các text một cách dễ dàng hơn (Số lượng các ảnh chứa Art Text khá nhiều hơn 40 phần trăm và số lượng các ảnh có diện tích text chiếm hơn 10 phần trăm cũng là gần 40 phần trăm)

## 2.2 Hạn chế của hướng tiếp cận thứ nhất

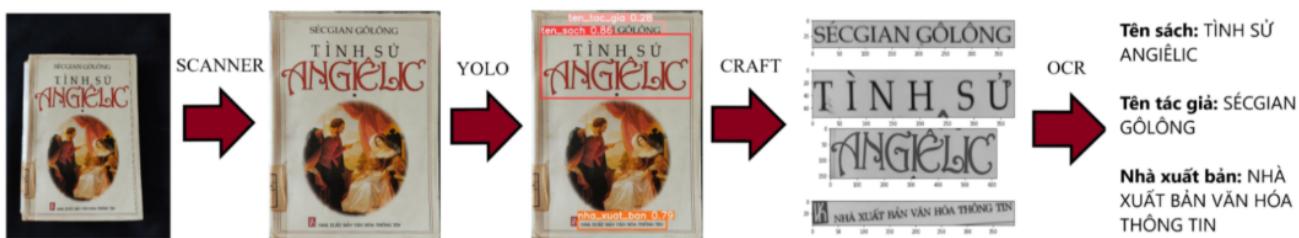


Ở hình minh họa trên ta có thể thấy rằng khi các vùng chứa các trường thông tin có nhiều line thì mô hình text recognition của chúng ta sẽ hoạt động không tốt. Vì dữ liệu đó không phù hợp với dữ liệu mà các mô hình text recognition học là các line text hoặc các word text (số lượng multi line text hơn 30 phần trăm nên cần phải xử lý luôn cả TH này )



Hình 4

### 2.3 Hướng tiếp cận đề xuất tiếp theo



Hình 5: Pipeline

- **Preprocessing:** Ở bước này là giai đoạn tiền xử lý nhằm loại bỏ các vùng không thuộc sách nhằm giảm thiểu các yếu tố nhiễu để tăng hiệu suất huấn luyện
- **Object detection:** Ở bước này, ứng với mỗi trường thông tin ta có thể xem như là một đối tượng cần nhận diện và áp dụng một số phương pháp tiêu biểu cho bài toán Object detection như Yolov4, Yolov5,...
- **Text detection:** Ở bước này sau khi detect được bounding box chứa các trường thông tin, ta cần nhận biết các text nằm trong các bbox trên, vì thế ta sẽ sử dụng các mô hình phát hiện text như CRAFT, ABCNet, DB net,....
- **Text recognition:** Sau khi crop các vùng chứa text thành các ảnh chứa text, các bức ảnh trên sẽ được đưa qua mô hình nhận diện chữ như VietOCR, SRN, DeepText,...

## 3 Giới thiệu bộ dữ liệu

Do bài toán gồm nhiều bước và ngoài trừ giai đoạn thu thập dữ liệu bìa sách thô là không yêu cầu phải label dữ liệu vì vậy, nhóm sẽ chuẩn dữ liệu cho 3 phần: dữ liệu tự thu thập để kiểm thử phần mềm, dữ liệu dùng để huấn luyện Object detection, dữ liệu dùng để huấn luyện mô hình Text recognition.

### 3.1 Ảnh Input đầu vào

Gồm 400 ảnh bìa sách được chụp bằng camera smartphone, chất lượng ảnh tối thiểu 720x960, 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.



### 3.2 Object Detection - YOLOv5

Data dành cho model YOLOv5 gồm 6982 ảnh bìa sách (do nhóm crawl) và file.txt (do nhóm dán nhãn). Ảnh crawl chứa đầu sách từ nhiều nhà xuất bản khác nhau như Nhà xuất bản Trẻ, Nhà xuất bản Kim Đồng, Nhà xuất bản ĐHQG-TPHCM, Nhà xuất bản Hà Nội,... và sử dụng online tool makesense.ai để dán nhãn gần 7000 ảnh bìa sách này, thu được các file .txt chứa tọa độ các bounding boxes của các vùng có thông tin cần thu thập trên bìa sách. Trong đó:

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



Hình 6

### 3.3 Text Recognition - VietOCR

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.



## 4 Giới thiệu các Model

- 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. Ở bài toán trích xuất thông tin nhóm chúng em sẽ sử dụng mô hình YOLO để detect các vùng chứa 6 trường thông tin chứa thông tin về sách
- Text localization: dùng để detect các line text trong ảnh
- Text recognition: sử dụng model VietOCR cho bài toán text recognition chữ tiếng Việt.

### 4.1 Giới thiệu về Model YOLO cho object detection

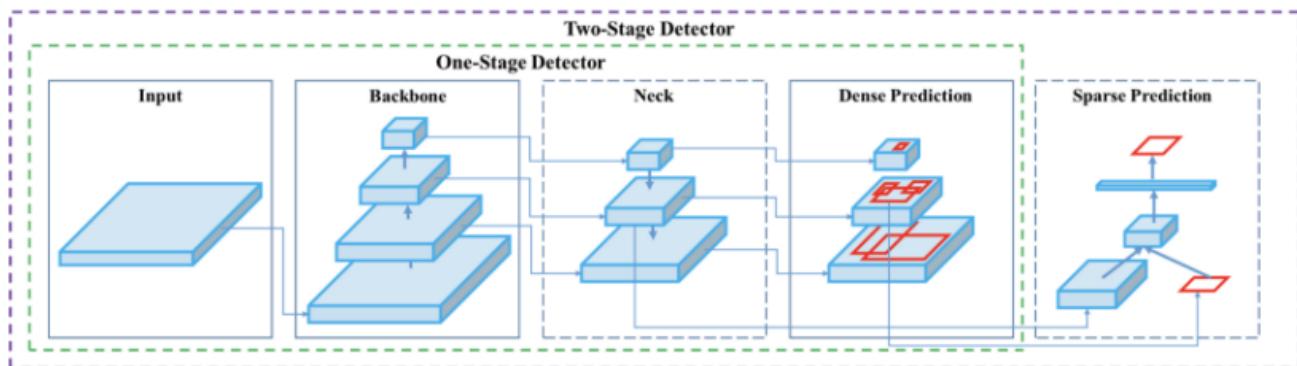
YOLO là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng



Hình 7

#### 4.1.1 Cấu trúc mạng của YOLO

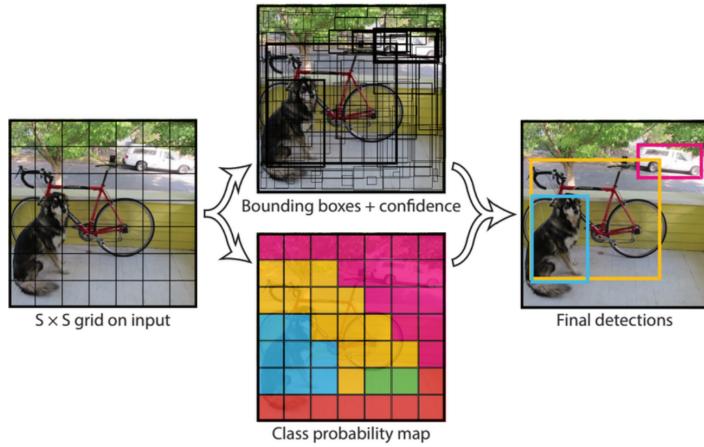
Kiến trúc mạng của YOLO gồm 4 phần:



Hình 8: Mô hình You Only Look Once

- Mạng Backbone: Trong YOLOv5, mạng CSP được dùng để trích xuất đặc trưng từ ảnh đầu vào. Sự thay đổi đã giúp tăng tốc thời gian xử lý đáng kể đối với các cấu trúc mạng nhiều tham số.
- Mạng Neck: Mạng Neck hoạt động dựa trên khái niệm pyramid (FPN) giúp cải thiện độ chính xác của mô hình đối với cái vật có kích thước nhỏ.
- Mạng Head: Nhiệm vụ chính của mạng này là phân nhận diện vật thể. Đầu ra của mạng sẽ là xác suất của lớp, độ tự tin của box và 4 kích thước chính của bounding box.

#### 4.1.2 Cách YOLO hoạt động



Ý tưởng chính của YOLOv1 là chia ảnh thành một lưới các ô (grid cell) với kích thước  $S \times S$  (mặc định là  $7 \times 7$ ). Với mỗi grid cell, mô hình sẽ đưa ra dự đoán cho  $B$  bounding box. Ứng với mỗi box trong  $B$  bounding box này sẽ là 5 tham số  $x, y, w, h, \text{confidence}$ , lần lượt là tọa độ tâm ( $x, y$ ), chiều rộng, chiều cao và độ tự tin của dự đoán. Với grid cell trong lưới  $S \times S$  kia, mô hình cũng dự đoán xác suất rơi vào mỗi class. Độ tự tin của dự đoán ứng với mỗi bounding box được định nghĩa là  $p(\text{Object})IoU$  trong đó  $p(\text{Object})$  là xác suất có vật trong cell và  $IoU$  là intersection over union của vùng dự đoán và ground truth. Xác suất rơi vào mỗi class cho một grid cell được ký hiệu  $p(\text{Class}|\text{Object})$ . Các giá trị xác suất cho  $C$  class sẽ tạo ra  $C$  output cho mỗi grid cell. Lưu ý là  $B$  bounding box của cùng một grid cell sẽ chia sẻ chung một tập các dự đoán về class của vật, đồng nghĩa với việc tất cả các bounding box trong cùng một grid cell sẽ chỉ có chung một class.

Vậy tổng số output của mô hình sẽ là  $SS(5B + C)$ .

#### 4.1.3 Loss Function

Hàm lỗi trong YOLO được tính trên việc dự đoán và nhãn mô hình để tính. Cụ thể hơn nó là tổng độ lỗi của 3 thành phần sau:

Độ lỗi của việc dự đoán loại nhãn của object - Classification loss, hàm lỗi này chỉ tính trên những ô vuông có xuất hiện object, còn những ô vuông khác ta không quan tâm. Classification loss được tính bằng công thức sau:

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

Trong đó:

$\mathbb{I}_i^{obj}$ : bằng 1 nếu ô vuông đang xét có object ngược lại bằng 0

$\hat{p}_i(c)$ : là xác suất có điều kiện lớp  $c$  tại ô vuông tương ứng mà mô hình dự đoán

Độ lỗi của dự đoán tọa độ tâm, chiều dài, rộng của boundary box ( $x, y, w, h$ ) (Localization loss) là hàm lỗi dùng để tính giá trị lỗi cho boundary box được dự đoán bao gồm tọa độ tâm, chiều rộng, chiều cao của so với vị trí thực tế từ dữ liệu huấn luyện của mô hình. Lưu ý rằng chúng ta không nên tính giá trị hàm lỗi này trực tiếp từ kích thước ảnh thực tế mà cần phải chuẩn hóa về  $[0, 1]$  so với tâm của bounding box. Việc chuẩn hóa này kích thước này giúp cho mô hình dự đoán nhanh hơn và chính xác hơn so với để giá trị mặc định của ảnh. Giá trị hàm Localization loss được tính trên tổng giá trị lỗi dự đoán tọa độ tâm ( $x, y$ ) và ( $w, h$ ) của predicted bounding box với ground-truth bounding box. Tại mỗi ô có chưa object, ta chọn 1 boundary box có IOU (Intersection over Union) tốt nhất, rồi sau đó tính độ lỗi theo các boundary box này. Giá trị hàm lỗi dự đoán tọa độ tâm ( $x, y$ ) của predicted bounding box và ( $,$ ) là tọa độ tâm của truth bounding box được tính như sau:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$$

Giá trị hàm lỗi dự đoán ( $w, h$ ) của predicted bounding box so với truth bounding box được tính như sau :

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$$

Độ lỗi của việc dự đoán bounding box đó chứa object so với nhãn thực tế tại ô vuông đó - Confidence loss Là độ lỗi giữa dự đoán boundary box đó chứa object so với nhãn thực tế tại ô vuông đó. Độ lỗi này tính trên cả những ô vuông chứa object và không chứa object.

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

Total loss Tổng lại chúng ta có hàm lỗi là tổng của 3 hàm lỗi trên:

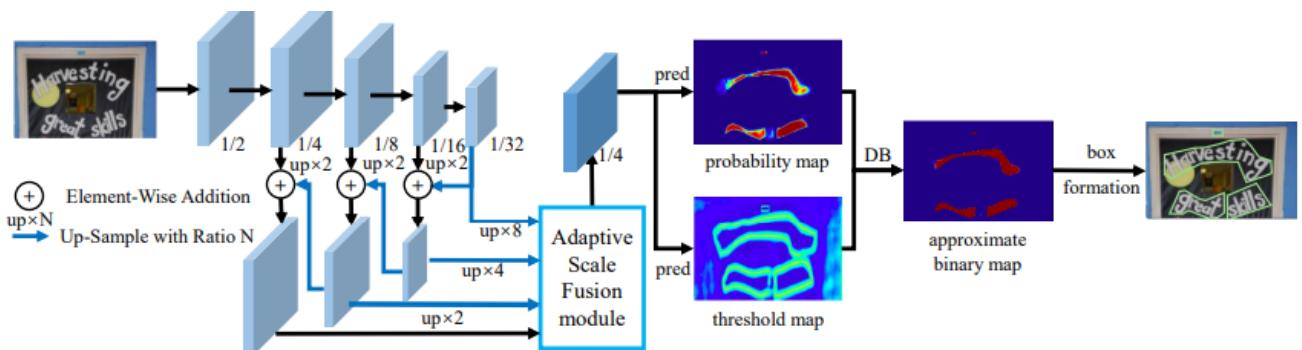
$$L_{total} = L_{classification} + L_{localization} + L_{confidence}$$

Models	MAP@0.5	MAP@0.5:0.95	Inference Time
YOLOv4	0.927	0.678	0.3
YOLOv5	0.908	0.653	0.25

## 4.2 Các Model sử dụng cho Text Localization

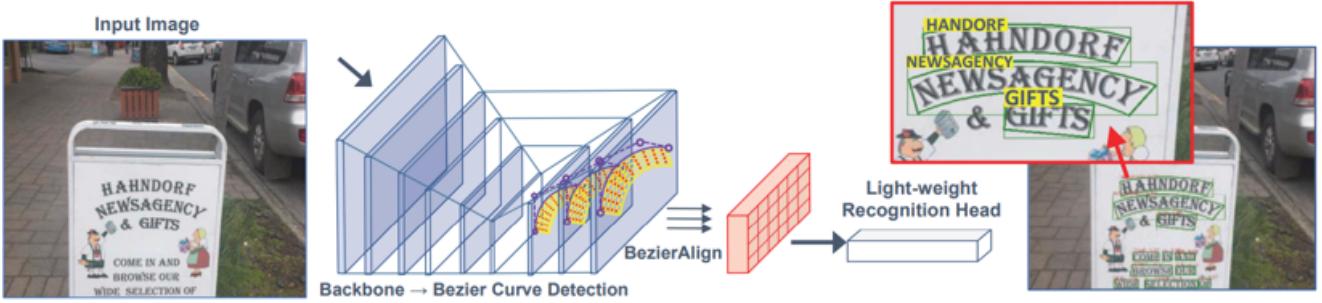
### 4.2.1 Paddle OCR

Nhóm chúng em sử dụng một frame work rất hiệu quả cho các bài toán chung là Paddle OCR ở đây mô hình text detection nhóm chọn là mô hình DB net với backbone là resnet 50 và được train trên tập dữ liệu CTW1500 (tập dữ liệu theo line).DB net đề xuất một mô-đun Binarization có thể phân biệt (DB) tích hợp quy trình encoding binary , một trong các bước quan trọng nhất trong quy trình xử lý của bài toán thành một mạng segmentation. Được tối ưu hóa cùng với mô hình module segmentation network có thể tạo ra kết quả chính xác hơn, giúp tăng cường độ chính xác của việc phát hiện văn bản với chỉ một pipeline duy nhất . Hơn nữa kết hợp với một module Dung hợp Quy mô Thích ứng (ASF) hiệu quả được đề xuất để cải thiện hiệu quả cách kết hợp các tính năng với các kích thước khác nhau một cách tương ứng.



Hình 9: Text Localization

#### 4.2.2 ABCNet



Hình 10: ABCNet

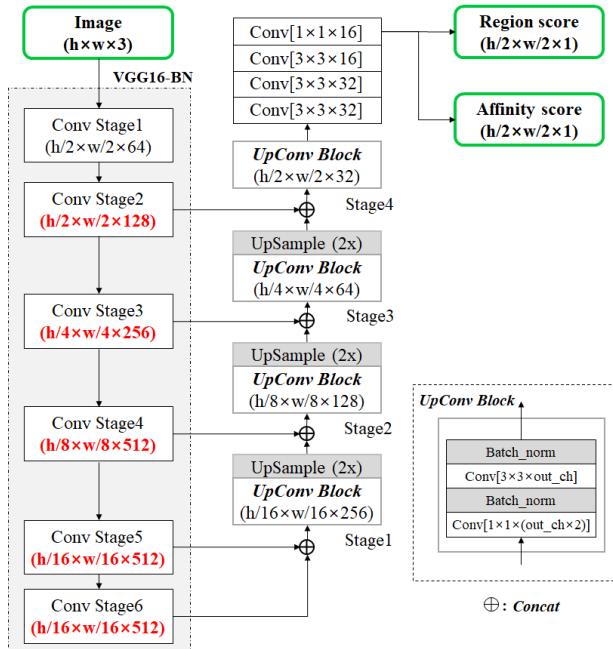
ABCNet là một framework End-to-end dùng để phát hiện văn bản có hình dạng tùy ý (xiên, cong, hoặc có dạng gợn sóng,...). Mô hình sử dụng single-shot, anchor-free CNN làm detection framework. Việc loại bỏ các anchor boxes giúp đơn giản hoá và giảm thời gian đáng kể trong việc thực hiện detect → phù hợp với bài toán Real-time Scene Text spotting. Dưới đây là pipeline của mô hình, trong đó Bezier Curve Detection dùng để detect scene text ở phần Detection Head và Bezier-Align dùng để chuẩn hoá các text cong về dạng thẳng để hỗ trợ cho việc nhận dạng nội dung text đó ở phần Recognition Head. Còn ở phần recognition sẽ sử dụng một mô hình rất hiệu quả cho các bài toán OCR là mô hình CRNN+CTC loss. Ở đây nhóm sẽ tách phần text detection của ABC riêng để dự đoán và sử dụng pretrained trên tập dữ liệu CTW1500

#### 4.2.3 CRAFT Text Detector

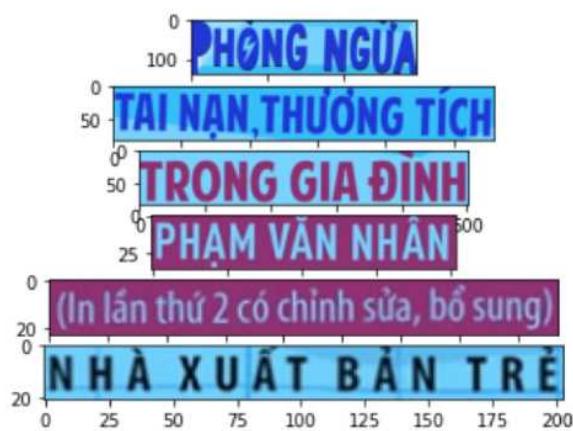
Nhóm sử dụng model deep-learning có sẵn trên pypi/craft-text-detector 0.4.2: CRAFT: Character-Region Awareness For Text detection để thực hiện locate các text trên bìa sách. Đây là một PyTorch dùng cho craft text detection, nó detect được khá hiệu quả bằng cách tìm ra phân vùng của từng từ chữ cái và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

Kiến trúc network: ứng dụng này hoạt động với mục đích chính là locate chính xác từng ký tự trong ảnh. Nhóm tác giả đã 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:



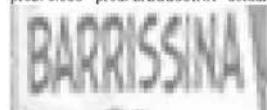
Hình 11: Cấu trúc model Craft.



Model	Precision	Recall	H-Mean
CRAFT	86	81.1	83.5
ABC net	85.4	80.7	83.0
PaddleOCR	84.2	77.1	80.6

### 4.3 Text Recognition-VietOCR

prob: 0.933 - pred: BARRISSINA - actual: BARRISSINA



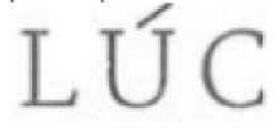
prob: 0.929 - pred: NHỈ? - actual: Nhỉ?



prob: 0.916 - pred: MÂY - actual: mây



prob: 0.935 - pred: LÚC - actual: LÚC



Hình 12

#### 4.3.1 Giới thiệu mô hình VietOCR

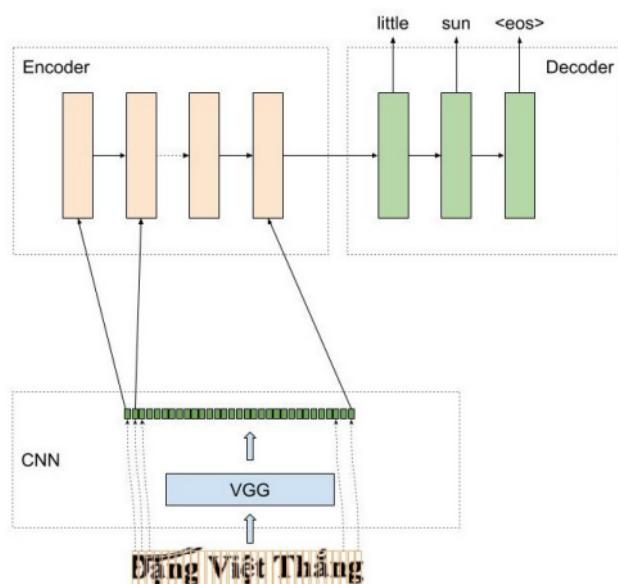
Thư viện này kết hợp CNN cùng hai mô hình khá nổi tiếng trong việc xử lý ngôn ngữ tự nhiên (cũng như về mặt hình ảnh) là: Transformer và Attention của seq2seq. Đây đều là những mô hình nổi tiếng, hiệu quả, đã được khắc phục nhiều hạn chế của các

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. Tuy nhiên Transformer lại predict khá chậm (cụ thể là so với Attention).

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).

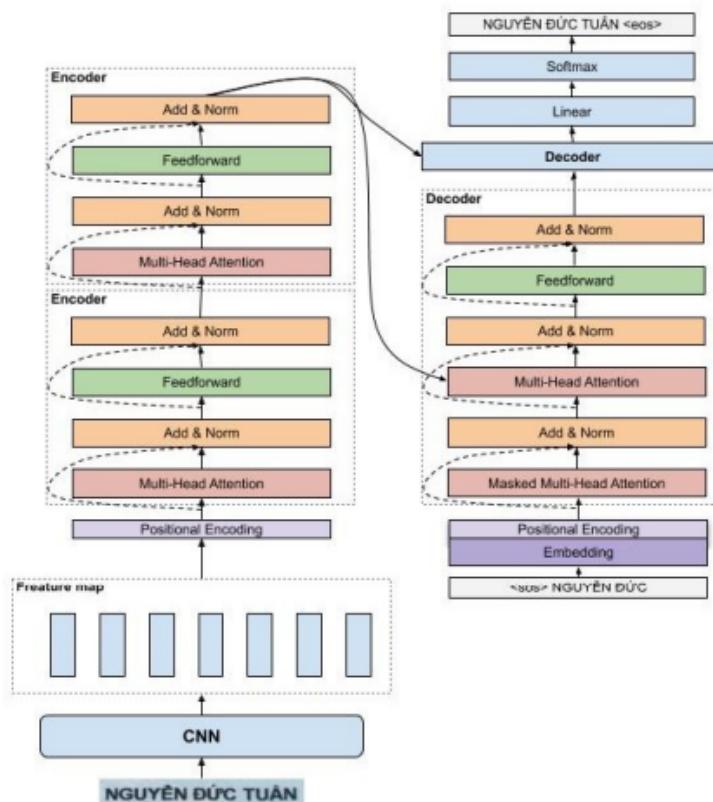
#### 4.3.2 Kiến trúc Network

##### AttentionOCR



Hình 13: Mô hình dùng CNN để trích xuất đặc trưng sau đó đi qua seq2seq sử dụng cơ chế attention

##### TransformerOCR



Hình 14: Mô hình sử dụng CNN để trích xuất đặc trưng sau đó đi qua transformer

Đầu tiên 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.

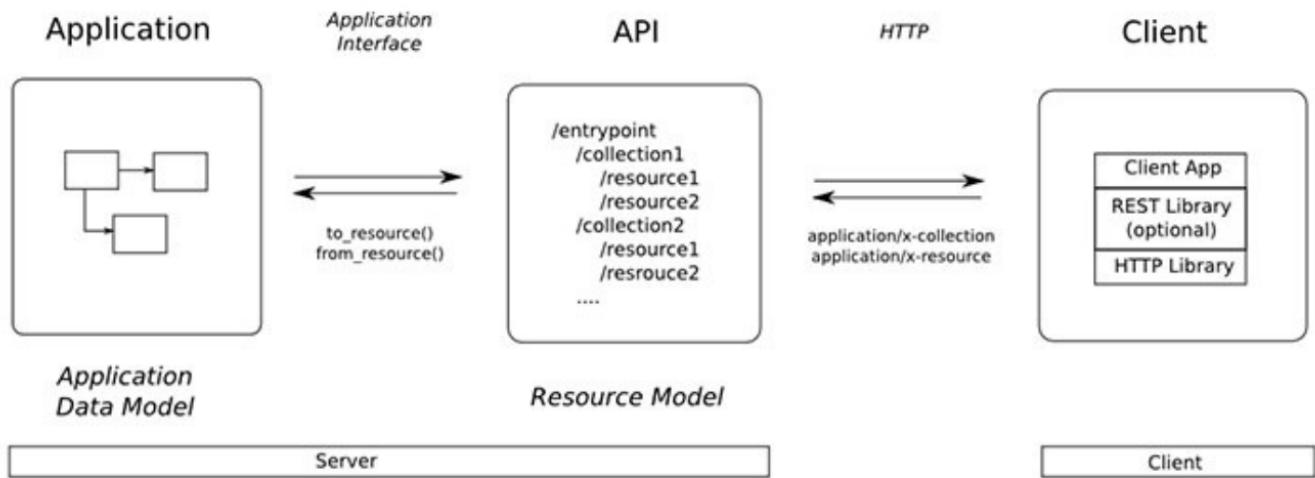
- **Model Zoo:** Mô hình này được huấn luyện trên tập dữ liệu gồm 10m ảnh, bao gồm nhiều loại ảnh khác nhau như ảnh tự phát sinh, chữ viết tay, các văn bản scan thực tế. Pretrain model được cung cấp sẵn. Model này có vẻ thích hợp với các tài liệu scan, đánh máy trên giấy... Mô hình được train bằng 2 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	0.93	60ms @ 2080
VGG19-bn - Seq2Seq	vgg_seq2seq	0.88	10ms @ 2080

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

## 5 Thiết kế hệ thống

Trong đồ án này, nhóm chúng em xây dựng một API bằng Flask web framework cho phía server.



Hình 15: RESTful API

Một số yêu cầu cơ bản:

- Hệ thống có thể upload file từ người dùng (định dạng file ảnh jpg/png/jpeg).
- Sau khi upload file thì khi muốn upload tiếp thì giao diện sẽ cho phép upload tiếp (upload -> extract -> re-upload)

## 6 Đánh giá pipeline

### 6.1 Giới thiệu các độ đo đánh giá mô hình

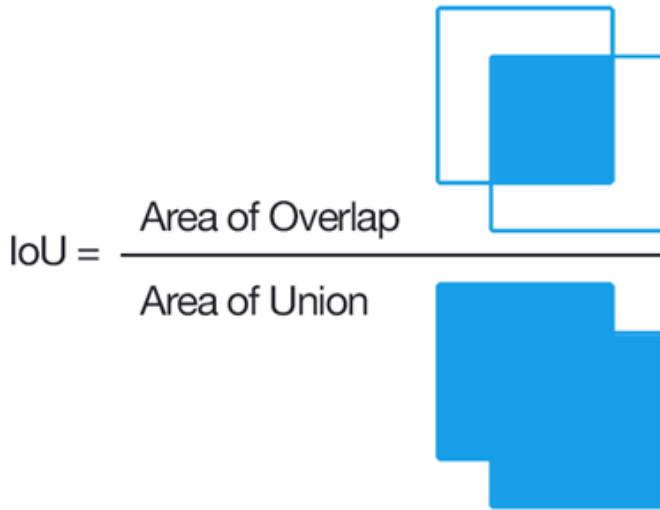
#### 6.1.1 Intersection over Union (IoU)

Intersection over Union (IoU) là tỷ lệ diện tích giữa phần giao và phần hợp của bounding box dự đoán và bounding box thực tế.

Với Area of Overlap là phần diện tích mà hai bounding box dự đoán và bounding box thực tế giao nhau và Area of Union là phần diện tích mà cả hai bounding box bao phủ trên ảnh như hình 16.

#### 6.1.2 True/False Positive/Negative

Kết quả của IoU là những giá trị trong khoảng (0,1) mỗi dự đoán sẽ có một giá trị IoU riêng. Để xác định liệu đó là dự đoán sai hay dự đoán đúng, chúng ta dựa vào một ngưỡng (threshold) cho trước (có thể là 0.5, 0.75, 0.95 tùy vào bài toán), nếu IoU lớn hơn hoặc bằng ngưỡng thì đó là dự đoán đúng, còn lại là dự đoán sai. Dựa vào những khái niệm trên chúng ta định nghĩa True/false positive/negative như sau:



**Hình 16:** Hình biểu diễn cho độ đo Intersection over Union (IoU)

- True Positive (TP): các bounding box dự đoán với IoU lớn hơn hoặc bằng 1 giá trị threshold (thường là 0.5).
- False Positive (FP): các bounding box dự đoán với IoU nhỏ hơn threshold.
- False Negative (FN): mô hình không bắt được đối tượng trong ảnh (ứng với ground truth tương ứng).
- True Negative (TN): Đây là thông số ít được quan tâm đến. Có thể hiểu là những phần của ảnh không chứa đối tượng và thực tế thì đúng là như vậy.

### 6.1.3 Precision

Precision là thang đo độ chính xác của dự đoán, được định nghĩa là tỉ lệ số điểm Positive mà mô hình dự đoán đúng trên tổng số điểm mà mô hình dự đoán là Positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detections}} \quad (1)$$

### 6.1.4 Recall

Recall là thang đo độ nhạy của khả năng tìm thấy các dự đoán đúng, được định nghĩa là tỉ lệ số điểm Positive mà mô hình dự đoán đúng trên tổng số điểm thật sự là Positive (hay tổng số điểm được gán nhãn là Positive ban đầu).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truth}} \quad (2)$$

### 6.1.5 Average Precision (AP)

Average Precision (AP) là một độ đo dùng để xấp xỉ phần diện tích phía dưới precision-recall curve. AP được tính bằng tích của precision ở mức  $k$  và sự chênh lệch của recall ở hai mức recall thứ  $k$  và  $k+1$ :

$$\text{AP} = \sum_{k=0}^{k=n-1} [\text{Recall}(k) - \text{Recall}(k+1)] \times \text{Precision}(k) \quad (3)$$

Với  $n$  là số lượng mức threshold, recall( $n$ ) bằng 0 và precision( $n$ ) = 1.

### 6.1.6 Mean Average Precision (mAP)

Mean average precision (mAP) là một độ đo dùng để tính AP trung bình trên nhiều lớp khác nhau và nhiều mức threshold khác nhau:

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}_k \quad (4)$$

Với  $n$  là số lớp,  $\text{AP}_k$  là AP của lớp thứ  $k$

### 6.1.7 Accuracy sequence và character

Ở bài toán Text Recognition, chúng ta sẽ đánh giá dựa trên hai độ đo là độ chính xác theo từng kí tự và độ chính xác cho từng chuỗi

## 6.2 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 400 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.

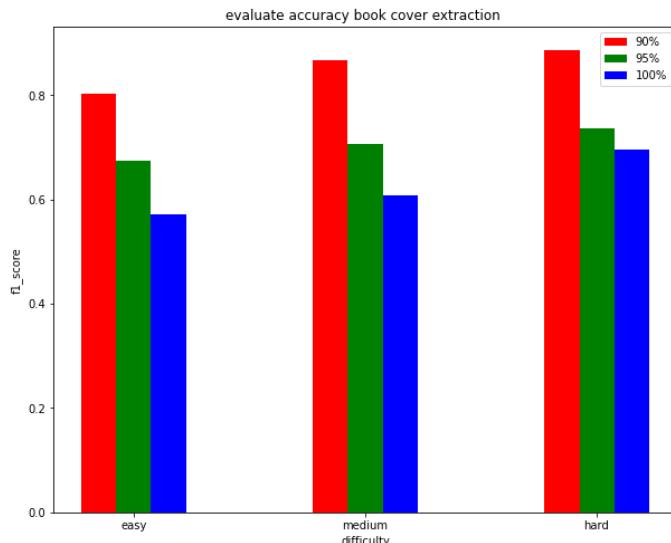
Kết quả:

- Easy:
  - 100%: 0.80355
  - >= 95%: 0.86644
  - >= 90%: 0.88694
- Medium:
  - 100%: 0.67382
  - >= 95%: 0.70687
  - >= 90%: 0.73651
- Hard:
  - 100%: 0.57132
  - >= 95%: 0.60857
  - >= 90%: 0.6948

### 6.2.1 Tốc độ xử lí

Model có thể linh hoạt sử dụng CPU lẫn GPU để phục vụ cho quá trình trích xuất, và dưới đây là thông kê thời gian trích xuất trung bình với mỗi ảnh:

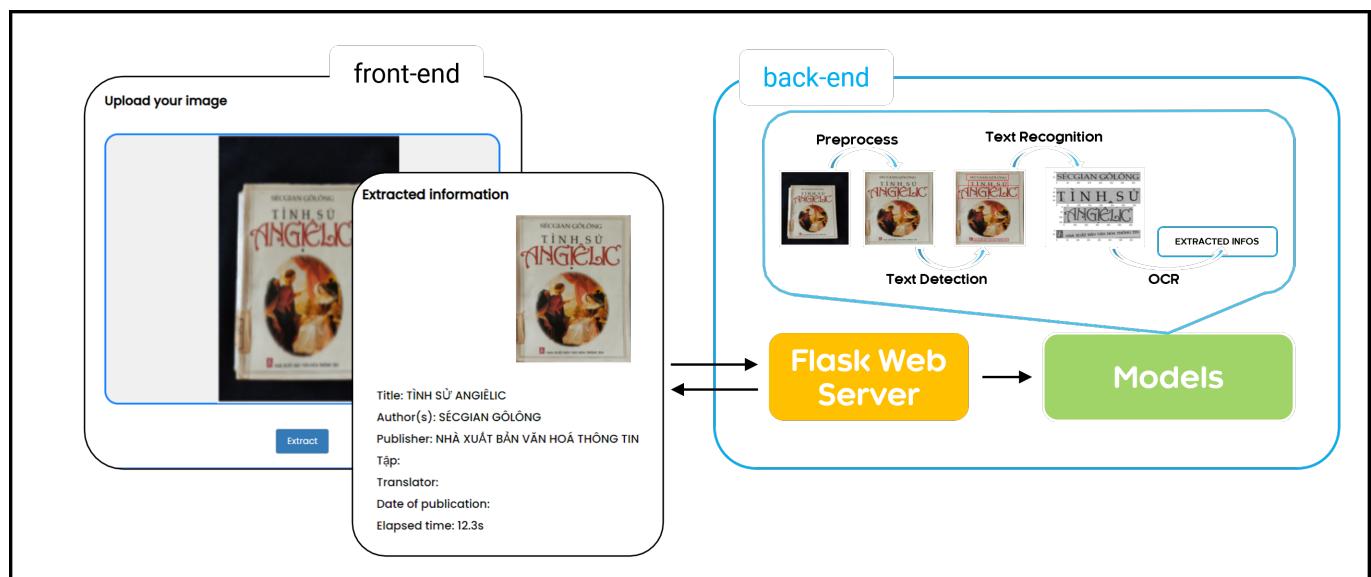
CPU i7-9750H	10 - 15s
GTX 1650 4GB VRAM	7 - 10s
GPU Google Colab Pro	1 - 2s



Hình 17: Thống kê đánh giá kết quả

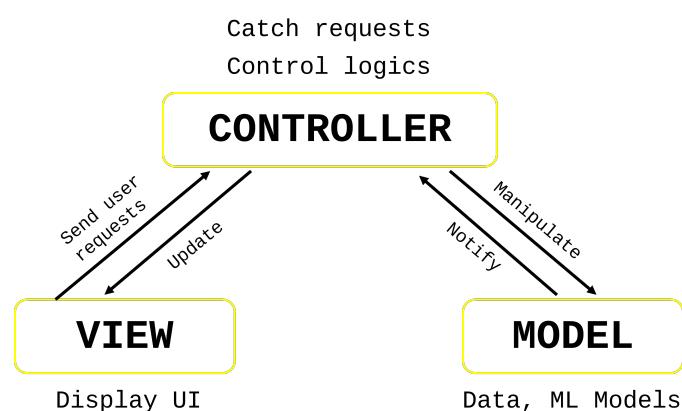
### 6.3 Kiến trúc hệ thống

#### 6.3.1 Tổng quan



Hình 18: Tổng quan kiến trúc hệ thống

Ứng dụng được xây dựng dựa trên mô hình MVC:



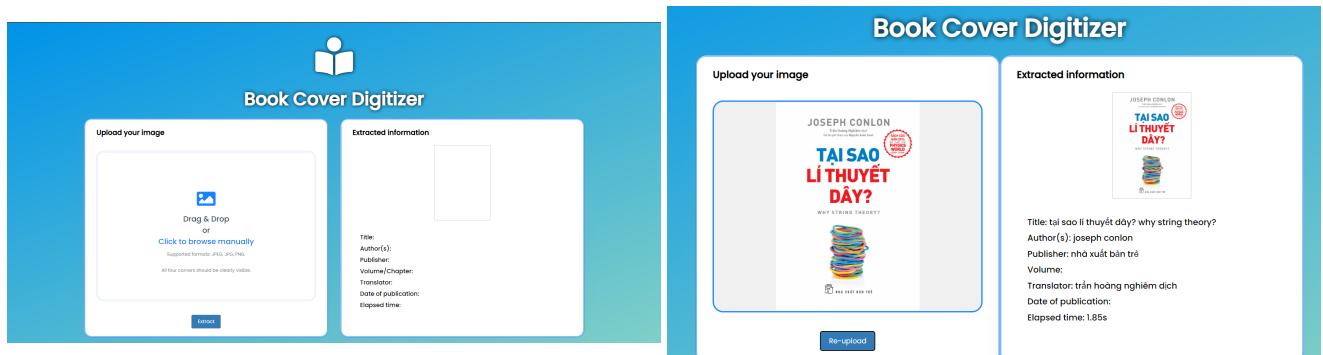
Hình 19: MVC

### 6.3.2 Chi tiết

- Model:** chứa các model trích xuất thông tin.
- View:** chứa và hiển thị giao diện người dùng, đảm nhận việc giao tiếp với người dùng, nhận input từ người dùng và gửi đến Controller và sau đó nhận lại thông tin trích xuất và hiển thị cho người dùng.
- Controller:** nhận yêu cầu từ View, kiểm tra và xử lý logic. Giao tiếp với Model để update View.

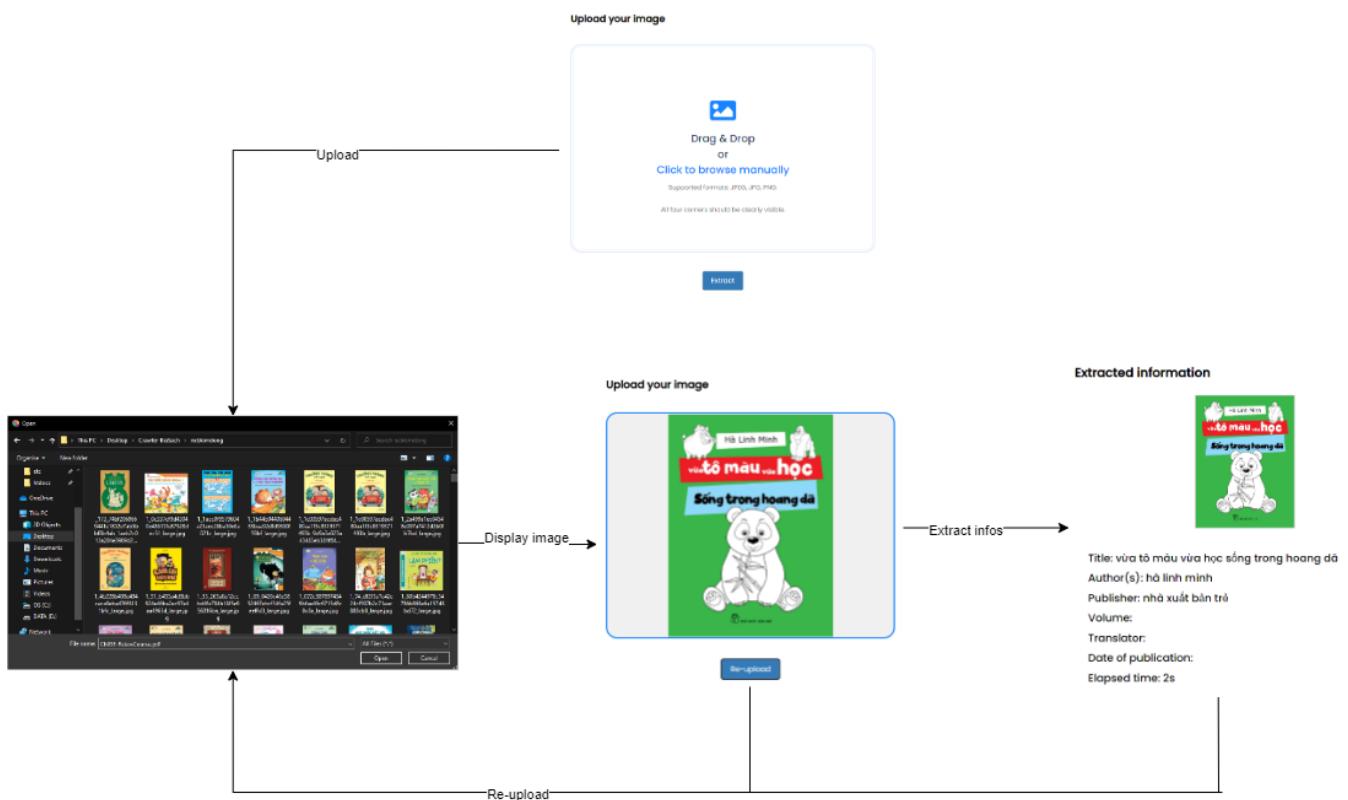
### 6.3.3 Giao diện hệ thống

Ứng dụng của nhóm bao gồm duy nhất một màn hình là trang chủ để nhận ảnh đầu vào từ người dùng và hiện thị kết quả trích xuất.



- Hành động xử lý:** Hiện thị kết quả trích xuất thông tin.
- Ý nghĩa:** Nhận kết quả trích xuất trả về của các models của và hiển thị lên màn hình.
- Điều kiện:** Sau khi upload ảnh và nhấn button "Extract".

### 6.3.4 Screenflow



Hình 20: Screenflow

## 6.4 Demo

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.

The figure consists of three separate screenshots of a web-based application titled "Book Cover Digitizer". Each screenshot shows a "Upload your image" section on the left and an "Extracted information" section on the right.

- Screenshot 1:** Shows a book cover for "CÔNG NGHỆ SẢN XUẤT RƯỢU VANG" (Technology of Wine Production). The extracted information includes: Title: công nghệ sản xuất rượu vang; Author(s): lê văn việt man; Publisher: nhà xuất bản đại học quốc gia tp.hồ chí minh; Volume: ; Translator: ; Date of publication: ; Elapsed time: 1.16s.
- Screenshot 2:** Shows a children's book cover for "vừa tò mò...vừa học Sống trong hoang dã" (Just Curious...Just Learn Living in the Wild). The extracted information includes: Title: vừa tò mò vừa học sống trong hoang dã; Author(s): hà linh minh; Publisher: nhà xuất bản trẻ; Volume: ; Translator: ; Date of publication: ; Elapsed time: 2s.
- Screenshot 3:** Shows a book cover for "TẠI SAO LÍ THUYẾT ĐÂY?" (Why String Theory?). The extracted information includes: Title: tại sao lí thuyết dây? why string theory?; Author(s): joseph conlon; Publisher: nhà xuất bản trẻ; Volume: ; Translator: trần hoàng nghiêm dịch; Date of publication: ; Elapsed time: 1.85s.

Hình 21: Kết quả predict đúng

The figure consists of three separate screenshots of the same web-based application, showing failed predictions for three different book covers.

- Screenshot 1:** Shows a book cover for "TÓ HOAI Dế Mèn phiêu lưu ký" (Tortoise and the Hare). The extracted information is mostly blank or incorrect, including: Title: tó hoai dế mèn phiêu lưu ký minh họa: tạ huy long; Author(s): ; Publisher: nhà xuất bản kim đồng; Volume: ; Translator: ; Date of publication: ; Elapsed time: 1.45s.
- Screenshot 2:** Shows a book cover for "VŨ TRỌNG PHUNG làm đi" (Vu Trong Phung's Stories). The extracted information is mostly blank or incorrect, including: Title: làm; Author(s): vũ trọng phung; Publisher: nhà xuất bản; Volume: ; Translator: ; Date of publication: ; Elapsed time: 1.16s.
- Screenshot 3:** Shows a book cover for "SING TO IT" by AMY HEMPEL. The extracted information is mostly blank or incorrect, including: Title: cinc g to t amy hempel; Author(s): new storisey by; Publisher: ; Volume: ; Translator: hempel; Date of publication: ; Elapsed time: 1.75s.

Hình 22: Kết quả predict chưa tốt

## 7 Cài đặt & Kiểm thử

Chức năng	Mức độ hoàn thành	Ghi chú
Upload ảnh	100%	
Crop ảnh trước khi truy xuất	70%	Đã implement crop tự động nhưng chưa cho phép người dùng crop thủ công
Lưu trữ	50%	Lưu trữ kết quả trả về cho View dưới dạng JSON nhưng chưa thiết kế Database để phục vụ việc lưu trữ lịch sử trích xuất

## 8 Kết luận

### 8.1 Pros

- Ứng dụng đáp ứng được nhu cầu cơ bản trích xuất thông tin bìa sách.
- Giao diện đơn giản và cực kì dễ sử dụng.
- Có thể linh hoạt sử dụng giữa CPU và GPU.
- Lưu trữ thông tin

### 8.2 Cons

- Thời gian trích xuất bằng CPU còn khá chậm.
- Kết quả trích xuất vẫn còn chưa chính xác hoàn toàn đối với một vài trường hợp.
- Khả năng áp dụng thực tế chưa cao.

### 8.3 Kết luận

Ứng dụng tuy đã đáp ứng được việc trích xuất thông tin nhưng vì chưa thiết kế Database nên khả năng áp dụng thực tế là chưa cao. Các định hướng tương lai:

- Thêm tính năng:
  - Cho phép user crop trên ảnh sau khi upload.
  - Lưu lại lịch sử trích xuất + tải về dưới dạng CSV.  
→ thiết kế database + các vấn đề về bảo mật.
- Lưu lại ảnh + kết quả trích xuất từ người dùng phục vụ cho việc cải thiện độ chính xác của model.
- Xây dựng API tích hợp cho bên thứ ba (nếu có nhu cầu).

## TÀI LIỆU THAM KHẢO

- [1] Medium. 2022. Can a neural network train other networks?.  
<https://towardsdatascience.com/can-a-neural-network-train-other-networks-cf371be516c6>.
- [2] Subramani, N., Matton, A., Greaves, M. and Lam, A., 2022. A Survey of Deep Learning Approaches for OCR and Document Understanding. arXiv.org.  
<https://arxiv.org/abs/2011.13534>
- [3] Scene text detection and recognition: a survey.  
<https://link.springer.com/article/10.1007/s11042-022-12693-7>
- [4] vietocr. Github  
<https://github.com/pbcquoc/vietocr>
- [5] vnese-id-extractor. Github  
<https://github.com/ntvuonggg/vnese-id-extractor>
- [6] MC-OCR Challenge 2021: An end-to-end recognition framework for Vietnamese Receipts  
<https://ieeexplore.ieee.org/document/9642121>
- [7] You Only Look Once: Unified, Real-Time Object Detection  
<https://arxiv.org/abs/1506.02640>
- [8] Character Region Awareness for Text Detection  
<https://arxiv.org/abs/1904.01941>