

# ĐẠI HỌC QUỐC GIA HÀ NỘI

## TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



# BÁO CÁO BÀI TẬP LỚN XỬ LÝ ẢNH INT3404E 20 NĂM HỌC 2023 – 2024

## Tên báo cáo:

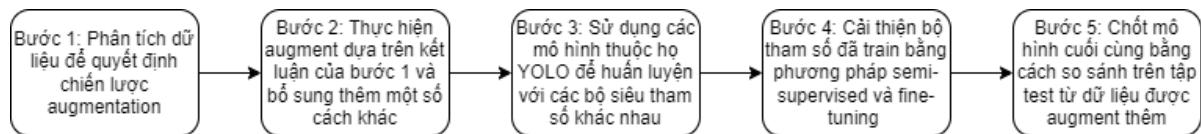
# OBJECT LOCALIZATION FOR SINO NOM'S CHARACTER WITH YOLO FRAMEWORK

Nhóm 2: Tăng Vĩnh Hà - K67-CA-CLC2  
Lê Thị Hải Anh - K67-CA-CLC2  
Vũ Nguyệt Hằng - K67-CA-CLC2  
Lê Xuân Hùng - K67-CA-CLC2

Giảng viên hướng dẫn: GS. Lê Thanh Hà  
ThS. Nguyễn Công Thương

HÀ NỘI - 2024

# 1 Tóm tắt



Hình 1: Pipeline cơ bản của nhóm

## 2 Phương pháp

### 2.1 Phân tích dữ liệu

Trước khi train model, nhóm đã thực hiện phân tích qua về dữ liệu của label trong tập train để đưa ra các chiến lược augmenting data. Nhóm đã thực hiện 2 hướng chính để phân tích dữ liệu được tóm tắt như sau:

- **Hướng tiếp cận 1:** Khảo sát các thông tin chung của bộ dữ liệu nhận được: số lượng ảnh, kích cỡ ảnh, đặc điểm chung các bức ảnh.
- **Hướng tiếp cận 2:** Khảo sát các tham số của các bounding box: chiều dài - rộng và tọa độ. Thực hiện bằng cách: vẽ các biểu đồ histogram từng aspect (tọa độ x - y, chiều dài - rộng của bounding box) trên toàn bộ bộ tập ảnh để rút ra 2 insight: đặc điểm của từng tập dữ liệu val - train và so sánh distribution giữa các tham số đó giữa tập train - test.
- **Hướng tiếp cận 3:** Tập trung vào các ảnh đặc biệt khiến kết quả model kém. Trong quá trình train model thành nhiều lần, nhóm nhận ra model ở các giai đoạn đầu (khi được train với số lượng epoch ít) thì làm kém trên những bức ảnh "mất cân bằng" về mức độ tập trung của ký tự trong ảnh. Nên nhóm đã đưa ra giả thuyết là: model sẽ được cải thiện hơn khi được "học" trên nhiều bức ảnh như vậy.

#### 2.1.1 Kết quả phân tích 1: Khảo sát các thông tin chung của bộ dữ liệu

Kết quả phân tích như sau:

- Số lượng ảnh: 70 ảnh tập train, 10 ảnh tập val.
- Kích cỡ ảnh: quan sát các ảnh trên tập dữ liệu train, kích cỡ các ảnh có khác nhau nhưng chủ yếu có 3 nhóm kích cỡ dựa theo tên của các ảnh (Ví dụ tên ảnh là "nlpnf-0137-01-001" thì ảnh thuộc nhóm kích cỡ "0137"). Cụ thể hơn 3 nhóm có kích cỡ W x H như sau:
  - Nhóm 0137: 900 x 608
  - Nhóm 0140: 750 x 640
  - Nhóm 0174: 800 x 632

Mỗi nhóm có chung chỉ số W còn H giao động quanh số đã nêu ở trên.

- Đặc điểm chung các bức ảnh: nhìn chung các bức ảnh rõ nét - không bị nhiễu - không bị xoay (được chụp thẳng đứng), bình thường và cùng có chung format là 2 trang giấy.
- Đặc điểm của file label:
  - Tọa độ x, y của các bounding box. Các tọa độ đã được chuẩn hóa nên  $x, y \in [0, 1]$ .
  - Chiều dài, rộng w, h của các bounding box. Chiều dài, rộng cũng đã được chuẩn hóa nên  $w, h \in [0, 1]$ .

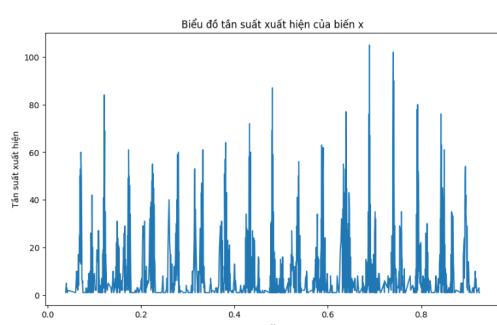
### Kết luận rút ra sau khi phân tích:

- Số lượng ảnh train và val còn quá ít, nhóm quyết định sẽ thêm vào tập train với số lượng ảnh được augment là: 70 (bằng số lượng ảnh trong tập train ban đầu).
- Các bức ảnh có các kích cỡ khác nhau theo từng cách đặt tên khác nhau, nhóm quyết định sẽ thử nghiệm 2 cách cùng resize về 1 kích cỡ (thực hiện bởi framework Ultralytics khi train mô hình YOLO): 640 x 640 và 1280 x 1280.
- Thực hiện các chiến lược augment đa dạng được nhắc đến phía sau với thư viện Albumentation để bổ sung vào tập train.

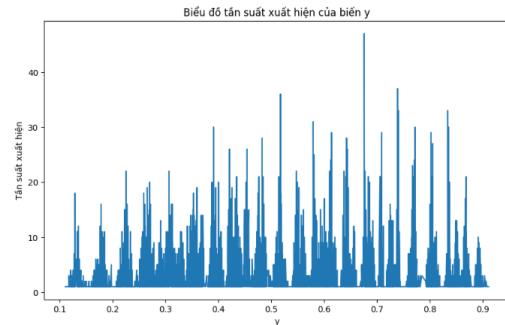
#### 2.1.2 Kết quả phân tích 2: Khảo sát các tham số của các bounding box

**Kết quả phân tích như sau:** các tham số của bounding box trên toàn bộ 70 ảnh tập train được thể hiện qua 4 biểu đồ histogram như sau: trục Ox là giá trị của các tham số và trục Oy là giá trị tần suất xuất hiện (số lần xuất hiện từng thuộc tính đang xét trên toàn bộ 70 ảnh):

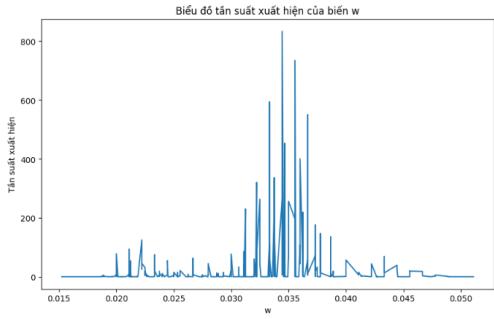
- Phân bố tọa độ x, y: tọa độ  $x, y$  của các kí tự phân bố khá đều, nhìn biểu đồ ta thấy tập trung nhiều nhất trong tọa độ  $x \in [0.6, 0.7]; y \in [0.65, 0.7]$  thể hiện qua hình 2 và hình 3.
- Phân bố của chiều dài và chiều rộng của tất cả các bounding box: tập trung từ 0.03 đến 0.045 thể hiện qua hình 4 và hình 5.
- So sánh các tham số nói trên giữa tập train và val: nhóm đã vẽ các biểu đồ Histogram của các tham số trên của tập labels train và tập labels val, và thấy 2 tập có distribution không quá khác nhau (so sánh phân bố của tọa độ x qua hình 2 và hình 6; phân bố của tọa độ y qua hình 3 và hình 7).



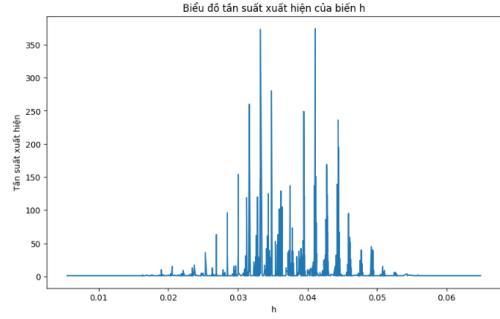
Hình 2: Phân bố của tọa độ x của tập train



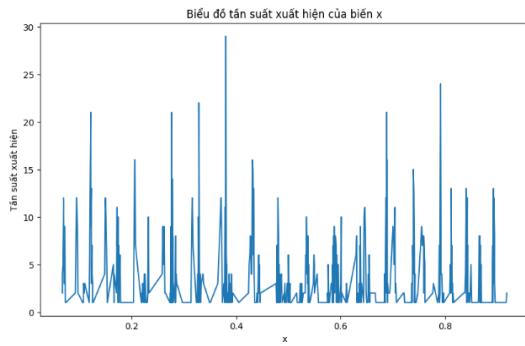
Hình 3: Phân bố của tọa độ y của tập train



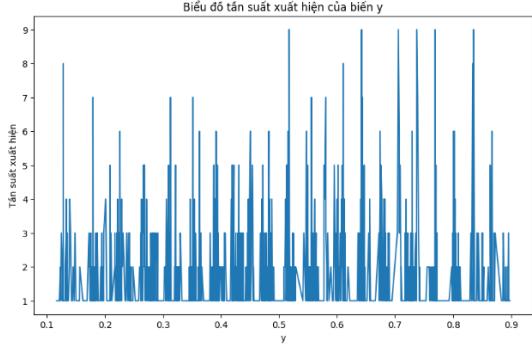
Hình 4: Phân bố của chiều rộng các bounding box trong tập train



Hình 5: Phân bố của chiều dài các bounding box trong tập train



Hình 6: Phân bố của tọa độ x của tập validation



Hình 7: Phân bố của tọa độ y của tập validation

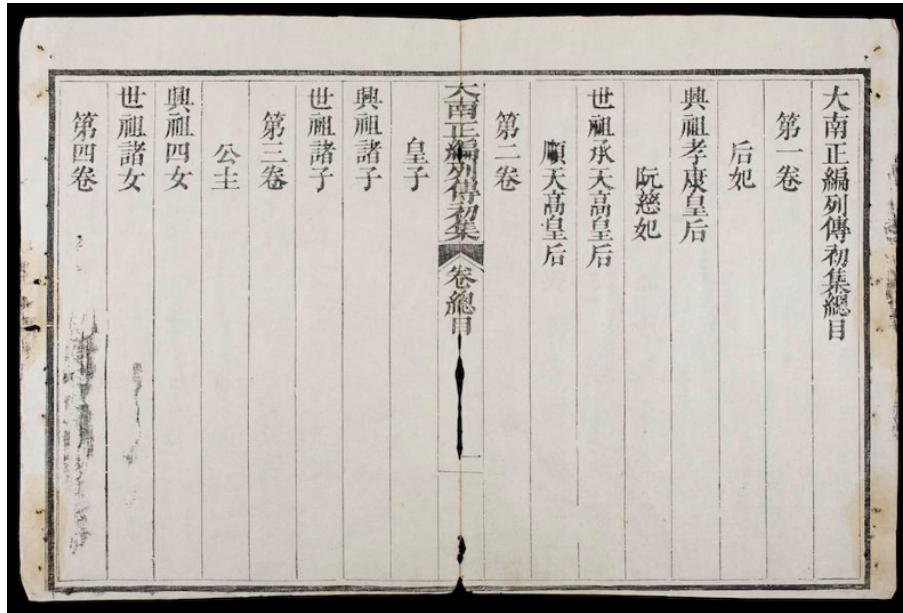
**Kết luận rút ra sau khi phân tích:** các tham số của bounding box vừa phân tích ổn, phân bố đều, không phải tác động quá nhiều bằng các phương pháp augment như: crop một phần của ảnh.

### 2.1.3 Kết quả phân tích 3: Tập trung vào các bức ảnh có khả năng khiến model kém

Như đã giải thích ở trên, nhóm đã nghiên cứu và đưa ra giải pháp sau:

- **Ý tưởng chung:** xét trên từng ảnh xem các bounding box của ảnh có tập trung quá nhiều vào 1 nửa của ảnh hay không. Nếu có thì khả năng model "học kém" trên ảnh đó sẽ cao hơn.
- **Cụ thể hơn,** đầu tiên ta chia ảnh thành 2 nửa (có 2 cách chia: chia theo chiều dọc và chiều ngang) đồng thời ta xây dựng 1 ngưỡng *rate*.
- Ở đây, *rate* là tỉ lệ độ thừa hiện nửa ảnh có số ký tự ít hơn *rate%* hay không, miền giá trị của *rate* là [0, 0.5].
- Từ đó, ta xây dựng 1 thuật toán như sau: nếu một nửa ảnh có số ký tự nhỏ hơn *rate%* của ảnh thì ảnh đó sẽ được phân vào danh sách những ảnh có nguy cơ "học kém", ta gọi các ảnh này là các *ảnh bị lệch* - hình 8. Danh sách các *ảnh bị lệch* sẽ tăng khi ta tăng *rate* lên.

- Tuy nhiên, đôi khi *rate* cao quá thì điều kiện lọc sẽ bị quá lỏng. Vậy nên nếu ta không điều chỉnh độ *rate* sao cho hợp lý thì sẽ có rất nhiều ảnh bị phân nhầm vào nhóm ảnh bị *lệch*. Đây cũng là điều lưu ý quan trọng mà nhóm rút ra được sau khi thử phân tích với nhiều độ *rate* khác nhau.



Hình 8: Minh họa cho "ảnh bị lệch"

### Kết quả rút ra được sau khi phân tích:

- Sau khi thử với nhiều độ *rate* khác nhau, nhóm quyết định chọn các rates phù hợp nhất đó là 0.3 và 0.35.
- Sau đó, nhóm quyết định sinh ra 2 bộ dữ liệu được augment bằng các phương pháp phía bên dưới tương ứng với 2 độ *rate* trên kết hợp với bộ dữ liệu gốc tạo ra 2 bộ training data hoàn chỉnh rồi đưa vào train model xem bộ dữ liệu nào cho kết quả tốt hơn.

## 2.2 Tiền xử lý dữ liệu

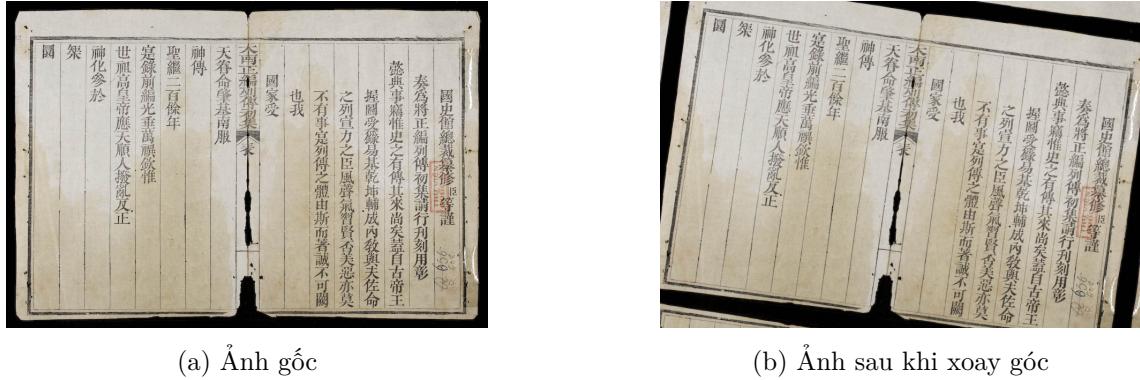
Trong phần này, nhóm xin trình bày về việc tiền xử lý dữ liệu bằng cách tăng cường dữ liệu (augmentation) cho tập dữ liệu ảnh và nhãn. Việc tăng cường dữ liệu ảnh đã được thực hiện trên dữ liệu ảnh ban đầu (*original data*) và dữ liệu ảnh đặc biệt có nguy cơ bị lỗi theo hướng tiếp cận 3 đã đề cập với *rate* = 0.3 và *rate* = 0.35.

### 2.2.1 Các kỹ thuật augmentation cơ bản

Nhóm đã sử dụng thư viện *albumentations* của Python hỗ trợ để thực hiện các kỹ thuật augmentation cơ bản sau đây:

- Xoay ảnh theo nhiều góc  
Để tạo ra các phiên bản mới của ảnh và nhãn, nhóm đã sử dụng phép xoay ảnh với nhiều góc khác nhau trong đó giới hạn là  $15^\circ$ . Quá trình này giúp tăng cường dữ

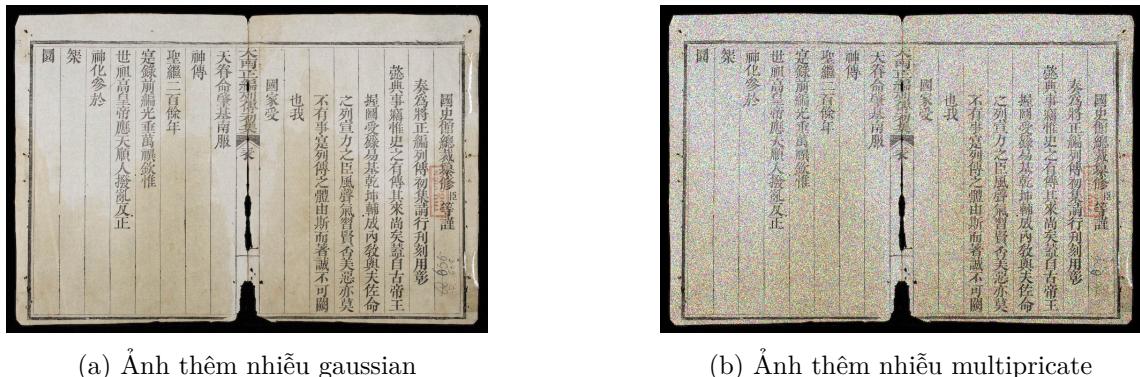
liệu bằng cách tạo ra các ảnh có góc nhìn khác nhau, từ đó làm tăng tính đa dạng của tập dữ liệu.



Hình 9: Kết quả phép xoay ảnh theo góc giới hạn  $15^\circ$

- Add noise

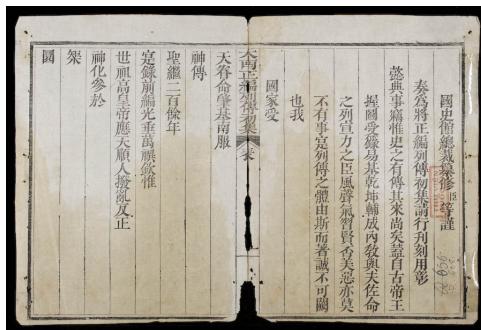
Nhóm đã áp dụng kỹ thuật thêm nhiễu vào ảnh để tạo ra các phiên bản mới với độ nhiễu khác nhau. Việc này giúp tạo ra sự đa dạng trong dữ liệu và làm tăng khả năng tổng quát hóa của mô hình. Nhóm sử dụng 2 loại nhiễu để thêm vào ảnh chính là gaussian noise và multipricate noise



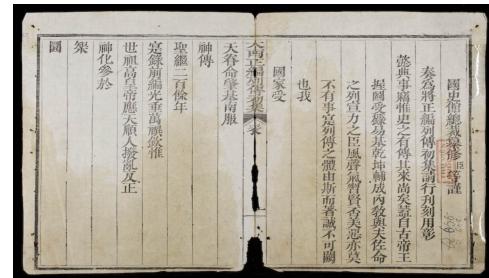
Hình 10: Kết quả phép thêm nhiễu

- Scale ảnh

Nhóm đã thực hiện phép biến đổi scale ảnh để tạo ra các phiên bản mới có kích thước khác nhau. Điều này giúp tạo ra các ảnh có kích thước và tỷ lệ khác nhau, từ đó tăng cường tính đa dạng của tập dữ liệu.



(a) Ảnh gốc

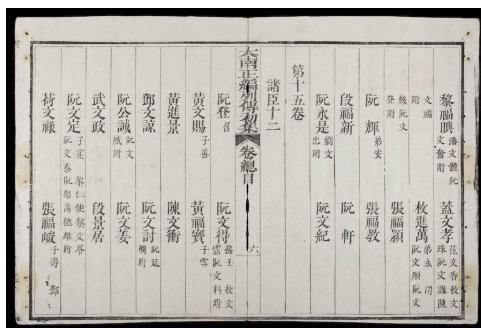


(b) Ảnh sau khi scale

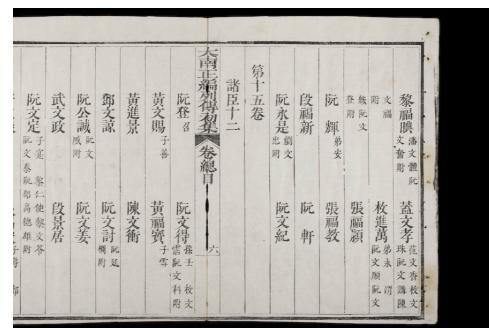
Hình 11: Kết quả phép scale ảnh

- Dịch chuyển ảnh sang trái phải

Nhóm thực hiện dịch ảnh trên trục  $x$  và thiết lập giới hạn là 0.3 - tức là ảnh có thể dịch chuyển tối đa 30% chiều rộng của nó sang trái hoặc phải. Các vùng khoảng trống sau khi dịch ảnh sẽ được lấp đầy bằng màu đen. Điều này giúp mô hình học giảm sự phụ thuộc vào vị trí và tạo ra một tập dữ liệu phong phú hơn.



(a) Ảnh gốc



(b) Ảnh sau khi dịch

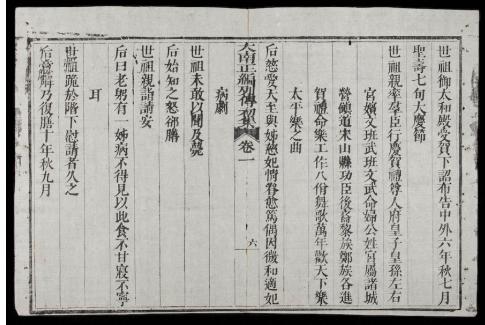
Hình 12: Kết quả phép dịch ảnh

- Crop ảnh chỉ show 1 phần (YOLO có sẵn nêu Hà viết)
- Các kỹ thuật augmen khác được tích hợp sẵn trong framework YOLOv8 (Hà viết)

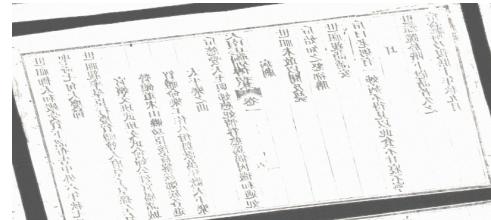
### 2.2.2 Các kỹ thuật augmentation nâng cao

- Kết hợp nhiều kỹ thuật augmentation trong cùng 1 bức ảnh

Nhóm đã kết hợp nhiều kỹ thuật augmentation cơ bản trong cùng một bức ảnh. Trong đó, ngoài 3 phép biến đổi đã đề cập, nhóm đã kết hợp thêm các phép biến đổi như *Horizontal\_flip*, *Blur* và *BrightnessContrast*, trong đó mỗi phép biến đổi con được thực hiện với xác suất  $p = 0.5$



(a) Ảnh gốc



(b) Ảnh sau biến đổi với nhiều kỹ thuật

Hình 13: Kết quả phép biến đổi kết hợp nhiều kỹ thuật

- Chuẩn hóa ảnh (không rõ YOLOv8 có làm hay không để Hà tìm hiểu)

### 2.2.3 Tập dữ liệu được bổ sung thêm

Sau khi áp dụng các kỹ thuật trên, tập dữ liệu mà nhóm bổ sung thêm gồm tập train 70 ảnh như sau:

- Các ảnh được coi là biased theo threshold cho trước là 0.3 bị rotate đi: 9 ảnh
- Các ảnh bị thêm nhiễu Multiplicative: random 9 ảnh trong 70 ảnh gốc ban đầu
- Các ảnh được coi là biased theo threshold cho trước là 0.35 bị scaled, rotated và thêm nhiễu Gaussian: 21 ảnh.

Tập val gồm 15 ảnh như sau:

- 10 ảnh ngẫu nhiên trong toàn 70 ảnh train ban đầu bị scaled, rotated và thêm nhiễu Gaussian.
- 5 ảnh ngẫu nhiên trong 70 ảnh train ban đầu bị thêm nhiễu Multiplicative.

## 2.3 Thuật toán xác định bounding box

### 2.3.1 YOLOv5

Tổng quan kiến trúc

Tổng quan thư viện Ultralytics

Áp dụng vào bài toán detect chữ Nôm

### 2.3.2 YOLOv8

Tổng quan kiến trúc

Áp dụng vào bài toán detect chữ Nôm

## **2.4 Validation**

## **3 Kết quả**

## **4 Kết luận và thảo luận**