Report - Sino-nom character localization - Group 12

Mục lục

1 Giới thiệu

- Bài toán localization chữ Nôm là một thách thức quan trọng trong lĩnh vực xử lý ảnh và nhận dạng ký tự. Chữ Nôm, từng được sử dụng phổ biến trong văn học và hành chính của Việt Nam cổ đại, là một hệ thống chữ viết có cấu trúc phức tạp và đa dạng. Việc xác định chính xác vị trí của các ký tự chữ Nôm trong các tài liệu cổ không chỉ giúp bảo tồn và phục hồi các tác phẩm văn học quý giá mà còn hỗ trợ các nhà nghiên cứu trong việc nghiên cứu và hiểu rõ hơn về lịch sử, văn hóa và ngôn ngữ của dân tộc. Bài toán này đòi hỏi sự kết hợp của nhiều kỹ thuật hiện đại như xử lý ảnh, học sâu và trí tuệ nhân tạo, để có thể đối phó với các thách thức về độ phức tạp của ký tự, chất lượng hình ảnh kém và sự đa dạng trong cách viết. Việc giải quyết thành công bài toán localization chữ Nôm sẽ mở ra những cơ hội mới trong việc số hóa và khai thác kho tàng văn hóa phong phú của Việt Nam.

- YOLO (You Only Look Once) là một trong những mô hình hàng đầu trong lĩnh vực phát hiện đối tượng nhờ vào khả năng xử lý nhanh chóng và độ chính xác cao. YOLO nổi bật với cách tiếp cận đơn giản nhưng hiệu quả, chia ảnh đầu vào thành các ô lưới và dự đoán các hộp chứa đối tượng cùng với xác suất của chúng chỉ trong một bước duy nhất. Điều này giúp YOLO thực hiện phát hiện đối tượng trong thời gian thực, lý tưởng cho các ứng dụng yêu cầu tốc độ cao.

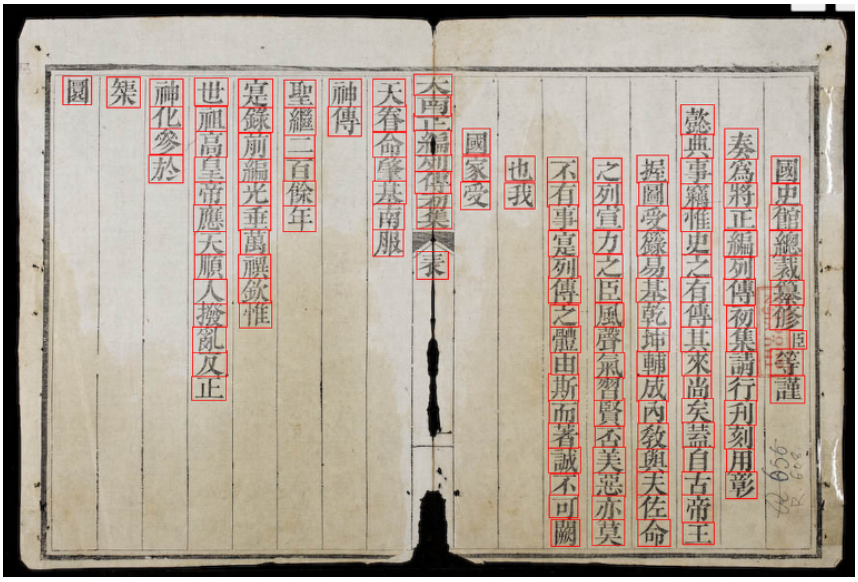
- Áp dụng YOLO vào bài toán localization chữ Nôm mang lại nhiều lợi ích đáng kể. Với khả năng nhận diện và định vị các đối tượng trong ảnh, YOLO có thể xác định chính xác vị trí của các ký tự chữ Nôm trong các tài liệu cổ. Quá trình này bắt đầu bằng việc huấn luyện mô hình YOLO trên một tập dữ liệu chứa các ảnh tài liệu và nhãn tương ứng với vị trí của từng ký tự chữ Nôm. Khi đã được huấn luyện, mô hình có thể quét qua các tài liệu chưa được chú thích, phát hiện và đánh dấu các ký tự chữ Nôm một cách nhanh chóng và hiệu quả.

2 Bộ dữ liệu và tiền xử lý ảnh

2.1 Bộ dữ liệu

- Bộ dữ liệu cho dự án định vị ký tự Sino-nom bao gồm 70 hình ảnh độ phân giải cao của các tài liệu Sino-nom cổ. Mỗi hình ảnh đều được chú thích với vị trí ký tự chính xác để làm dữ liệu cho việc huấn luyện.
- Nhãn dãn trong ảnh được cấu trúc theo định dạng:

  label_id x_center y_center bbox_width bbox_height
- Các hình ảnh có chất lượng tốt, rõ ràng và văn bản trông khá nổi bật so với nền
- Kích thước hình ảnh: các hình ảnh có kích thước không đồng nhất 900x610, 750x640, 800x632...
- Các hình ảnh có độ tương phản và độ sáng khác nhau
- Một số hình ảnh có chứa nhiễu



Hình 1: Visualize Data

2.2 Tiền xử lý ảnh

- Resize hình ảnh về kích thước cố định 416x416 phù hợp với mô hình yolo

```
def resize_image(img, size=(416, 416)):
    return cv2.resize(img, size)
```

- Điều chỉnh độ sáng và độ tương phản cải thiện tính nhất quán của dữ liệu

```
def adjust_brightness_contrast(img, brightness=20, contrast=20)
    if brightness != 0:
        shadow = brightness if brightness > 0 else 0
        highlight = 255 if brightness > 0 else 255 + brightness
        alpha_b = (highlight - shadow) / 255
        gamma_b = shadow
        img = cv2.addWeighted(img, alpha_b, img, 0, gamma_b)

    if contrast != 0:
        f = 131 * (contrast + 127) / (127 * (131 - contrast))
        alpha_c = f
        gamma_c = 127 * (1 - f)
        img = cv2.addWeighted(img, alpha_c, img, 0, gamma_c)

    return img
```

- Sử dụng các bộ lọc Gaussian Blur và Bilateral Filter giúp giảm nhiễu mà không làm mất đi các đặc trưng quan trọng của hình ảnh.

```
def apply_gaussian_blur(img, kernel_size=(3, 3), sigma=0.5):
    return cv2.GaussianBlur(img, kernel_size, sigma)

def apply_bilateral_filter(img, d=9, sigma_color=75, sigma_spac
    return cv2.bilateralFilter(img, d, sigma_color, sigma_space
```

2.3 Tăng cường dữ liệu

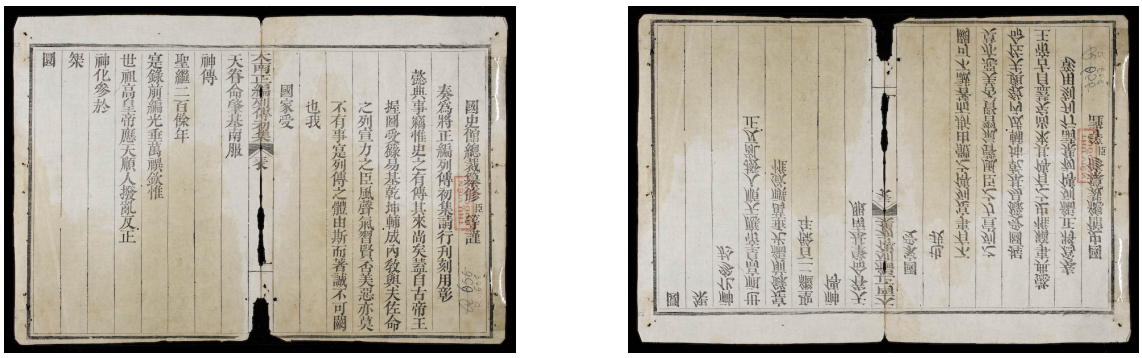
Chúng tôi tăng cường dữ liệu bằng các phương pháp:

- Lật ngang ảnh%. Phép biến đổi này lật ảnh từ trái sang phải (phản chiếu ngang) với xác suất 0.5. Điều này giúp mô hình nhận dạng các đối tượng bất kể chúng nằm ở phía nào của ảnh.

(a) Ảnh gốc (b) Ảnh lật ngang

Hình 2

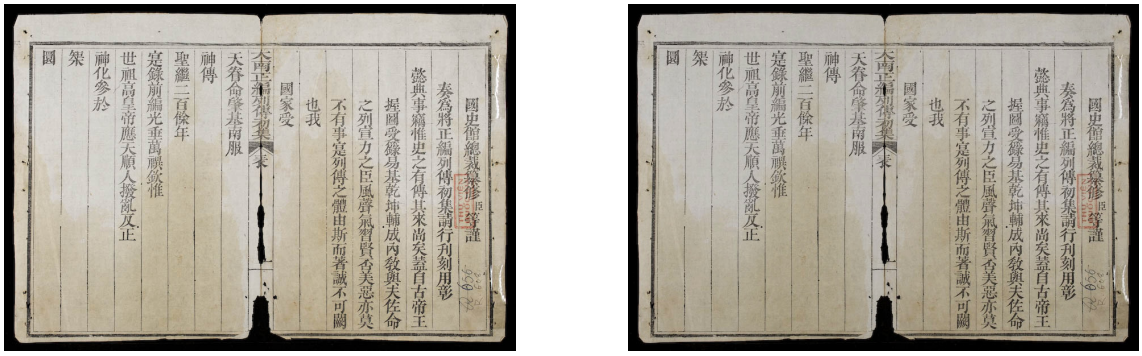
- Lật dọc ảnh %. Phép biến đổi này lật ảnh từ trên xuống dưới (phản chiếu dọc) với xác suất 0.5. Điều này giúp mô hình nhận dạng các đối tượng bất kể chúng nằm ở vị trí trên hay dưới của ảnh.



(a) Ảnh gốc (b) Ảnh lật dọc

Hình 3

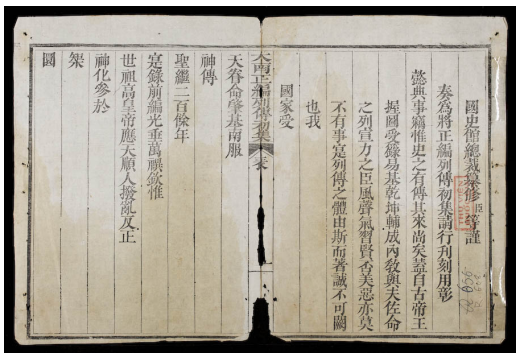
- Thay đổi mức độ bão hòa (Saturation) của ảnh. Phép biến đổi này thay đổi độ bão hòa của ảnh, làm cho ảnh trở nên sáng hơn hoặc tối hơn. Giá trị 0.7 giảm độ bão hòa của ảnh xuống 30%, trong khi giá trị 1.3 tăng độ bão hòa của ảnh lên 30%. Điều này giúp mô hình học cách xử lý các mức độ bão hòa khác nhau của ảnh.



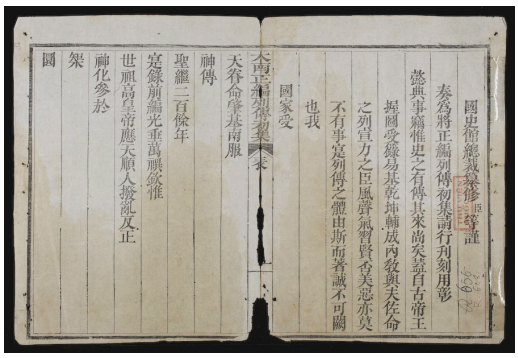
(a) Ảnh gốc (b) Ảnh thay đổi độ bão hòa

Hình 4

- Tăng giảm mức độ phơi sáng (Exposure) của ảnh. Phép biến đổi này thay đổi độ tương phản của ảnh, làm cho ảnh trở nên sáng hơn hoặc tối hơn. Giá trị 0.85 giảm độ tương phản của ảnh xuống 15%, trong khi giá trị 1.15 tăng độ tương phản của ảnh lên 15%. Điều này giúp mô hình học cách xử lý các mức độ phơi sáng khác nhau của ảnh.



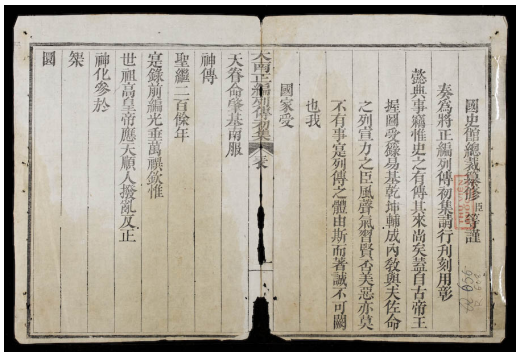
(a) Ảnh gốc



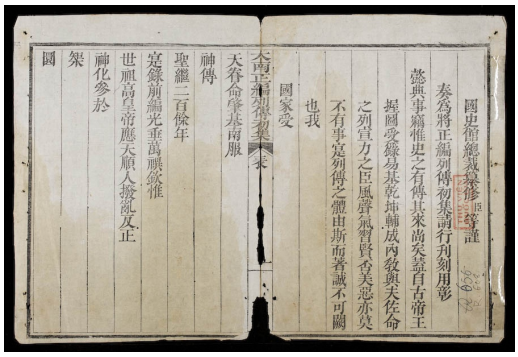
(b) Ảnh tăng giảm mức độ phơi sáng

Hình 5

- Thêm nhiễu vào ảnh.Phép biến đổi này thêm nhiễu Gaussian vào ảnh, với độ lớn của nhiễu thay đổi từ 0 đến 1.5% của giá trị pixel tối đa (255). Điều này giúp mô hình học cách xử lý các ảnh có nhiễu, làm cho mô hình trở nên bền vững hơn với các nhiễu loạn trong dữ liệu thực tế.



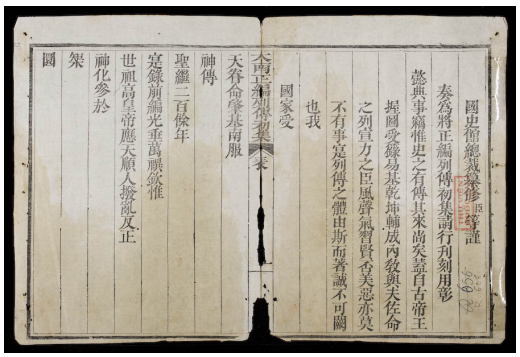
(a) Ảnh gốc



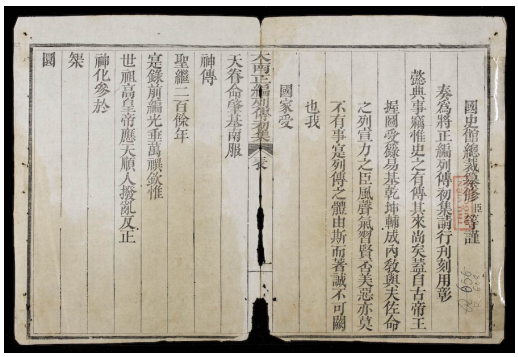
(b) Ảnh thêm nhiễu

Hình 6

- Thêm Gaussian Blur cho ảnh. Phép biến đổi này áp dụng hiệu ứng làm mờ Gaussian cho ảnh, với độ mờ được xác định bởi tham số sigma trong khoảng từ 0 đến 1.0. Sigma lớn hơn tạo ra độ mờ mạnh hơn. Điều này giúp mô hình học cách nhận dạng các đối tượng ngay cả khi ảnh bị mờ.



(a) Ảnh gốc

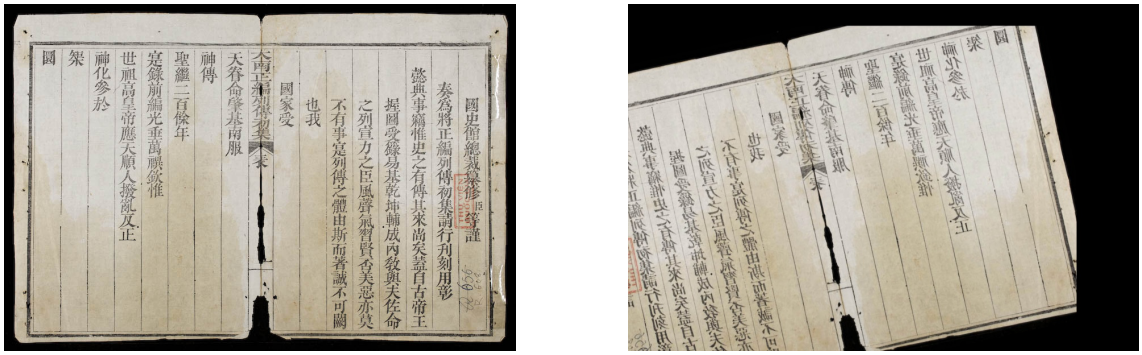


(b) Ảnh thêm Gaussian Blur

Hình 7

2.4 Tăng cường dữ liệu kết hợp

- Lật ngang ảnh với xác suất 50/
- Lật dọc ảnh với xác suất 50/
- Thay đổi mức độ bão hòa (Saturation) của ảnh trong khoảng từ -30% đến 30%. iaa.LinearContrast((0.85, 1.15)): Tăng giảm mức độ phơi sáng (Exposure) của ảnh trong khoảng từ -15% đến 15%.
- Thêm nhiễu vào ảnh với tỉ lệ nhiễu tối đa là 1.5% tổng số pixels.
- Thêm Gaussian Blur cho ảnh với kernel size từ 1x1 đến 5x5.



(a) Ảnh gốc

(b) Ảnh sau khi tăng cường

Hình 8

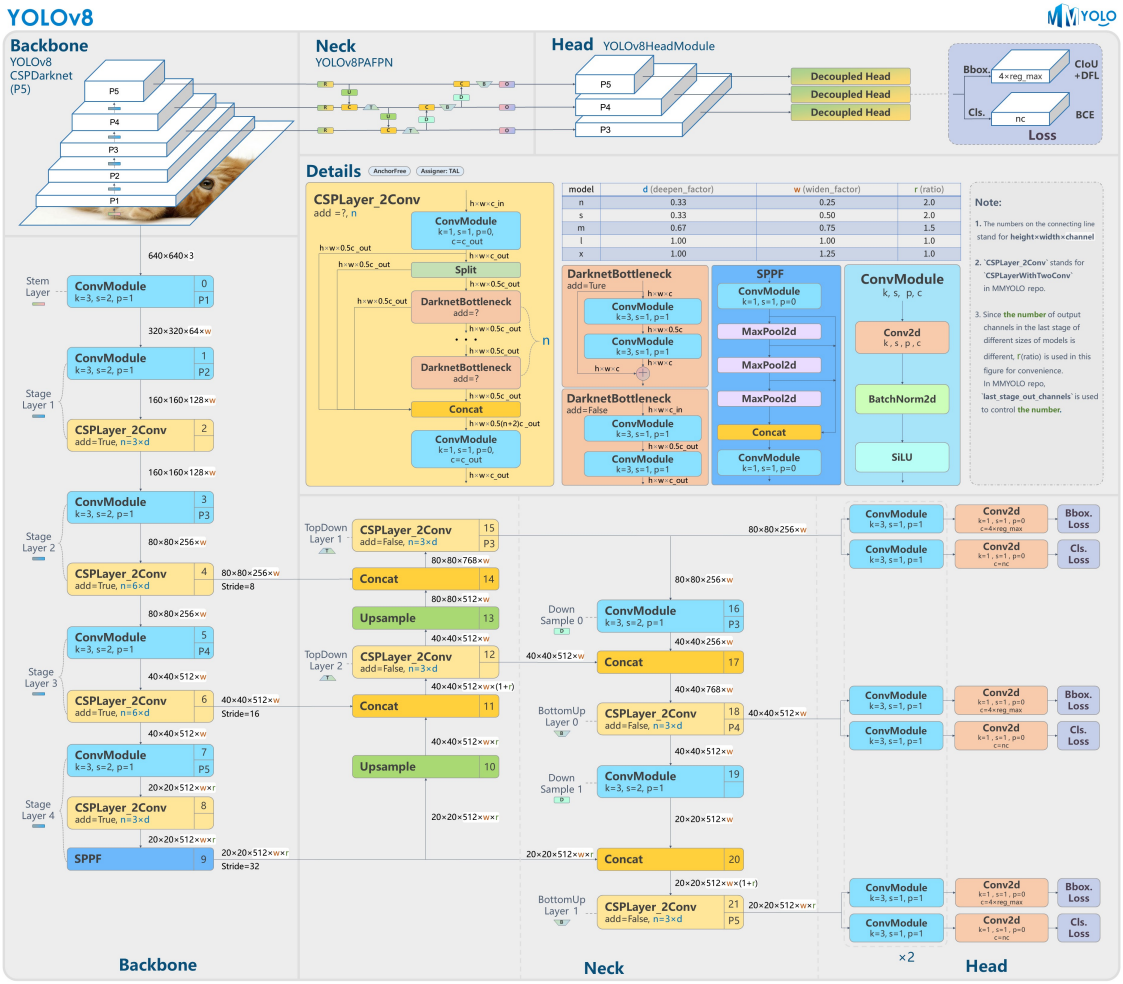
3 Các mô hình sử dụng

3.1 Yolov5

YOLOv5 là một phiên bản của dòng mô hình YOLO (You Only Look Once), nổi tiếng trong lĩnh vực phát hiện đối tượng (object detection). YOLOv5 được phát triển bởi Ultralytics và đã trở thành một trong những mô hình phổ biến nhất cho nhiệm vụ phát hiện đối tượng nhờ tính hiệu quả và dễ sử dụng của nó.

3.2 Yolov8

YOLOv8 là một mô hình nhận dạng đối tượng dựa trên mạng convolutional neural network (CNN). Mô hình YOLOv8 sử dụng một mạng neural kiến trúc darknet-53 để trích xuất đặc trưng của hình ảnh và áp dụng thuật toán nhận dạng đối tượng YOLOv8 trên các đặc trưng đó.



Hình 9

3.2.1 Kiến trúc cơ bản gồm

- Backbone: Mạng trích xuất đặc trưng từ ảnh, thường là các mạng CNN sâu như CSPDarknet.
- Neck: Mạng kết hợp và khuếch đại các đặc trưng từ backbone, thường là các FPN (Feature Pyramid Networks) hoặc PAN (Path Aggregation Networks).
- Head: Mạng thực hiện dự đoán bounding boxes, xác suất của các lớp (ở đây là các ký tự), và các điểm trọng yếu nếu cần.

3.2.2 Quy trình hoạt động của Yolov8

1. Trích xuất đặc trưng
   - Backbone: Đây là phần chịu trách nhiệm trích xuất các đặc trưng từ ảnh đầu vào. YOLOv8 sử dụng một mạng CNN (Convolutional Neural Network) hiệu quả, thường là các kiến trúc hiện đại như CSPDarknet hoặc các biến thể của nó, để tạo ra các đặc trưng cấp thấp và cao từ ảnh.
2. Tổng hợp đặc trưng với Neck
   - Neck: Phần này bao gồm các cấu trúc như FPN (Feature Pyramid Network) hoặc PAN (Path Aggregation Network), giúp tổng hợp và kết hợp các đặc

trưng từ các cấp độ khác nhau của backbone. Neck giúp mô hình hiểu rõ hơn về cấu trúc của đối tượng và cải thiện khả năng phát hiện đối tượng ở các kích thước khác nhau.

3. Dự đoán với Head

   - Head: Phần cuối của mô hình, nơi các dự đoán được thực hiện. Head sử dụng các đặc trưng từ neck để dự đoán các bounding box (hộp giới hạn), nhãn đối tượng (class labels), và độ tin cậy (confidence scores) cho mỗi ô lưới (grid cell).

4. Quy trình dự đoán

   - Grid Cells: Ảnh đầu vào được chia thành các ô lưới (grid cells). Mỗi ô lưới chịu trách nhiệm phát hiện các đối tượng trong phạm vi của nó.
   - Bounding Box Predictions: Mỗi ô lưới dự đoán một số lượng nhất định các bounding box. Mỗi bounding box đi kèm với các thông tin như tọa độ trung tâm, chiều rộng, chiều cao, và độ tin cậy (confidence score).
   - Class Predictions: Mỗi bounding box cũng dự đoán các nhãn đối tượng (class labels) với các xác suất tương ứng.

5. Hậu xử lý (Post-processing)

   - Non-Maximum Suppression (NMS): Sau khi các bounding box được dự đoán, NMS được sử dụng để loại bỏ các bounding box trùng lặp và giữ lại những bounding box có độ tin cậy cao nhất. Quá trình này giúp loại bỏ các dự đoán dư thừa và cải thiện độ chính xác của mô hình.
   - Thresholding: Áp dụng ngưỡng (threshold) để chỉ giữ lại các dự đoán có độ tin cậy vượt qua một mức nhất định, giúp giảm số lượng các phát hiện sai.

6. Đầu ra cuối cùng

   - Danh sách các bounding box: Các bounding box sau khi được lọc qua NMS và thresholding sẽ được trả về dưới dạng danh sách các đối tượng được phát hiện cùng với tọa độ và nhãn của chúng.

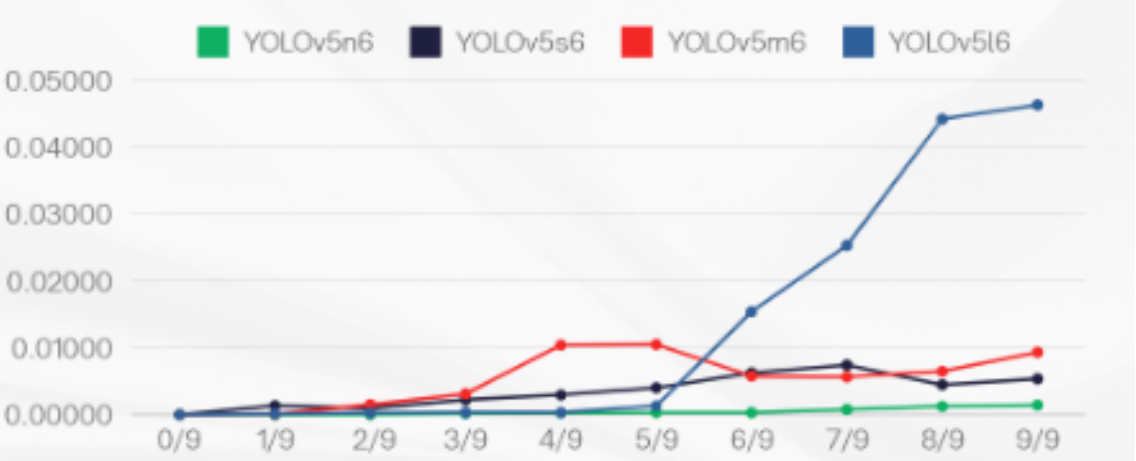
3.2.3 Fine-tuning với mô hình pretrained trong bài toán Localization character

- Tải mô hình pretrained: Sử dụng mô hình YOLOv8 đã được huấn luyện trước trên các tập dữ liệu lớn.
- Huấn luyện lại (Fine-tuning): Sử dụng tập dữ liệu của bạn để huấn luyện lại mô hình, giúp nó học được các đặc trưng cụ thể của ký tự cần nhận diện.
- Thay đổi các siêu tham số như batch size, learning rate, và epochs để đạt được hiệu suất và độ chính xác cao cho mô hình.

4 Thử nghiệm

4.1 Yolov5

Nhóm chúng tôi thử nghiệm bộ dữ liệu trên nhiều mô hình Yolov5 khác nhau



Hình 10: Thử nghiệm trên nhiều mô hình

4.2 Yolov8

- Huấn luyện mô hình Yolov8 với Yolov8x cùng với các siêu tham số và 2 bộ dữ liệu riêng biệt:
  - Bộ dữ liệu gốc
  - Bộ dữ liệu được tăng cường
- Siêu tham số

Siêu tham số	**Giá trị**
Epochs	100
Batch	2
lfr	0.0001

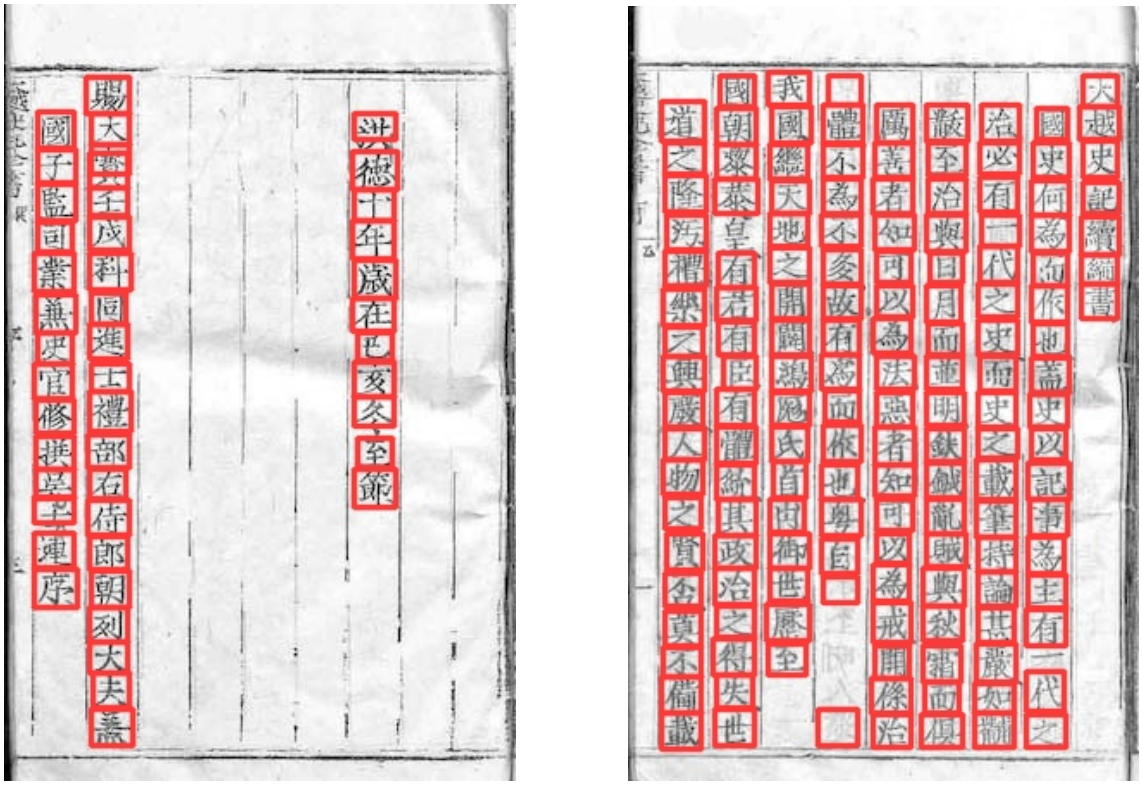
Bảng 1: Bảng các siêu tham số

- Thử nghiệm kết quả

Mô hình	**Giá trị**
Dữ liệu gốc	0.82
Dữ liệu được tăng cường	0.85

Bảng 2: Bảng các siêu tham số

- Kết quả



(a) (b)

Hình 11

5 Kết luận

- Việc sử dụng mô hình YOLOv8 kết hợp với các kỹ thuật tăng cường dữ liệu đã chứng minh hiệu quả vượt trội trong việc địa phương hóa chữ Nôm. Mô hình YOLOv8, với kiến trúc cải tiến và khả năng xử lý nhanh chóng, đã cho phép xác định vị trí chữ Nôm một cách chính xác và hiệu quả. Các kỹ thuật tăng cường dữ liệu như xoay, lật, và thay đổi độ sáng đã giúp làm phong phú tập dữ liệu huấn luyện, từ đó cải thiện khả năng nhận diện và giảm thiểu lỗi.
- Qua các thử nghiệm, mô hình YOLOv8 đã đạt được độ chính xác cao trong việc nhận diện chữ Nôm trên các tài liệu văn bản, thậm chí trong các điều kiện ánh sáng và góc nhìn khác nhau.