

**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**

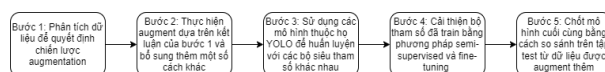
**Tên báo cáo:
OBJECT LOCALIZATION FOR SINO NOM'S
CHARACTER WITH YOLO FRAMEWORK**

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

Giáo 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.

Ở đây, có 3 hướng chính để tiếp cận dữ liệu như sau:

- Hướng tiếp cận 1: Vẽ các biểu đồ histogram từng aspect (tọa độ x , y , chiều dài chiều rộng) của ký tự trên toàn tập ảnh
- Hướng tiếp cận 2: Tính
- Hướng tiếp cận 3: Phân loại các ảnh đặc biệt có nguy cơ bị lỗi

2.1.1 Hướng tiếp cận 1: Vẽ các biểu đồ histogram từng aspect

Ở đây, aspect gồm 4 mục:

- Tọa độ x , y của các ký tự (các Bounding Box)
- Chỉ số chiều dài, rộng của Bounding Box w , h

Đầu tiên, tạo biểu đồ histogram thể hiện sự phân tán của tọa độ x , y của từng ký tự trong tất cả các ảnh

→ Các ảnh phân bố cho thấy tọa độ x , y của các ký tự phân bố khá đều, nhìn biểu đồ ta thấy tập trung nhiều nhất trong tọa độ $x \in [0.6, 0.7]$; $y \in [0.65, 0.7]$.

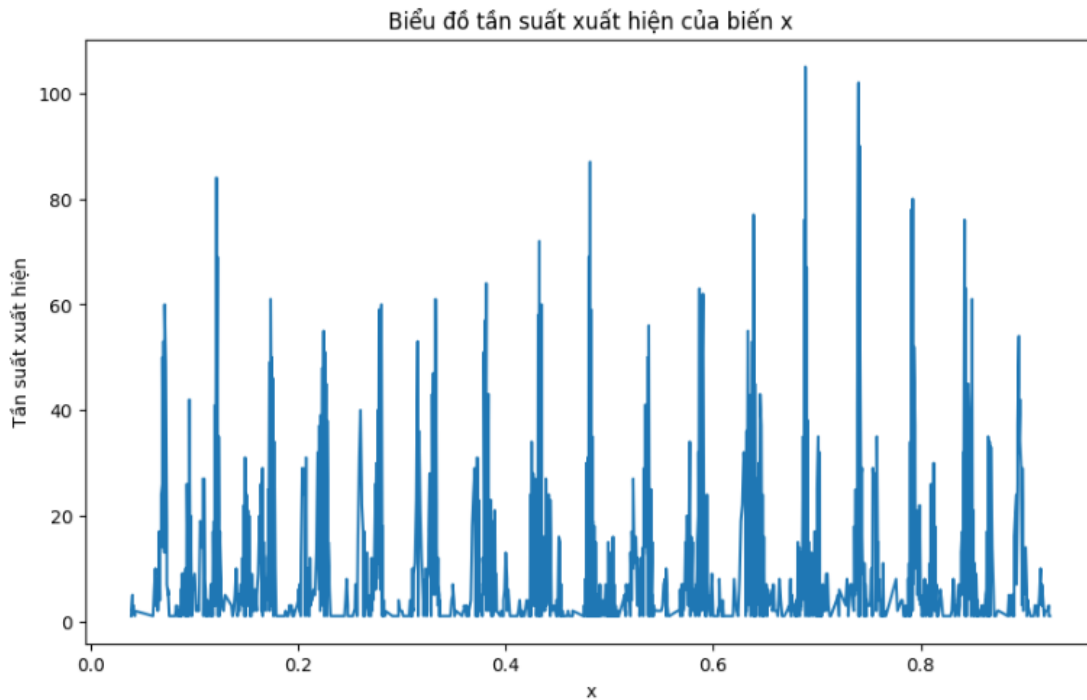
→ Có thể sinh *cropped_imgs* của vùng này trên 70 cái ảnh tập *train* nhưng phân bố của tọa độ x , y trải đều thì không bị coi là tập trung quá nhiều (bất thường).

Tiếp theo, ta quan sát phân bố của chiều dài và chiều rộng của tất cả các Bounding Box → Chiều rộng các Bounding Box phân bố tập trung từ 0.03 đến 0.045.

→ Chiều dài các BB từ 0.03 đến 0.04

2.1.2 Hướng tiếp cận 2: Xét xem variance từng aspect của Bounding Box trong 1 ảnh có bị lệch không

Phần này t định sort các ảnh có variance cao (độ lệch so với giá trị trung cao). Thế nhưng t nghĩ là variance cao có thể do nhiều yếu tố nữa => hướng này tạm để cuối.



Hình 2: Phân bố của tọa độ x

2.1.3 Hướng tiếp cận 3: Phân loại các ảnh đặc biệt có nguy cơ bị lỗi

Trong một lần train với số epoch thấp thì vô tình phát hiện được ra là model làm tốt ở các ảnh chữ thừa và làm không tốt ở các ảnh chữ không thừa.

Ý tưởng chung: Ta sẽ xét xem các Bounding Box của ảnh có tập trung quá nhiều vào 1 nửa của ảnh hay không. Nếu có thì khả năng lỗi ở ảnh đó sẽ cao hơn. Để cụ thể hóa hơn, ta chia ảnh thành 2 nửa (có 2 cách chia: chia theo chiều dọc và chiều ngang) và ta xây dựng 1 ngưỡng rate.

Ở đây, rate là tỉ lệ độ thừa thể hiện nửa ảnh có số ký tự ít hơn $\text{rate}\%$ hay không, miền giá trị của rate là $[0; 0.5]$.

Nếu một nửa ảnh có số ký tự nhỏ hơn $\text{rate}\%$ của ảnh thì ảnh đó sẽ bị liệt vào danh sách những ảnh có nguy cơ bị lỗi, ta gọi các ảnh này là các ảnh bị lệch.

Danh sách các ảnh bị lệch sẽ tăng khi ta tăng rate lên.

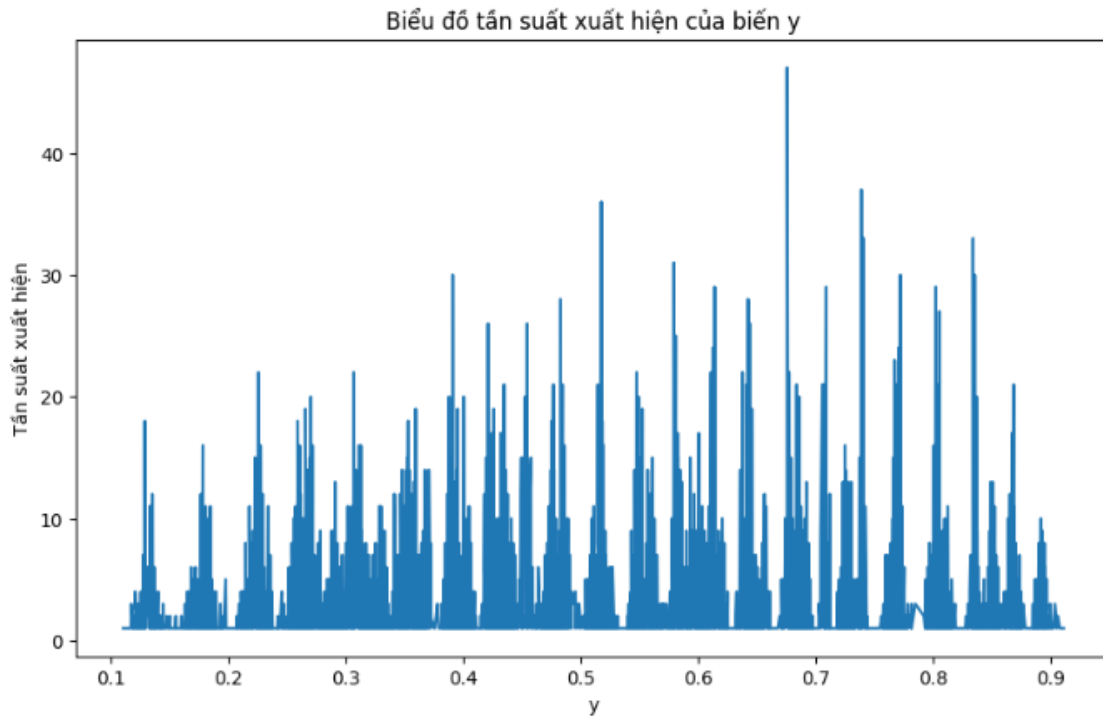
Đôi khi rate cao quá thì điều kiện lọc sẽ bị quá dễ \rightarrow rất nhiều ảnh không đến mức dễ lỗi mà bị lọc là lỗi.

\rightarrow Nhóm thực hiện sinh với các rates là 0.3 và 0.35 vì trước đó đã thử với nhiều rate khác nhau thì quan sát được rate từ 0.3 \rightarrow 0.35 cho dữ liệu tin cậy (nhìn nhận qua folder ảnh).

Sau đó, nhóm quyết định sinh ra nhiều bộ dữ liệu với các rate khác nhau vd bộ 1 gồm original + rate_0.3 và bộ 2 gồm original + rate_0.4 rồi đưa vào train model xem bộ dữ liệu nào tốt hơn.

2.1.4 Phân tích distribution của tập val có gì bất thường so với train không

Nhóm đã vẽ các biểu đồ Histogram (giống Hướng tiếp cận 1 đã nêu) trên tập labels val và không phát hiện bất thường so với tập labels train từ các aspect của Bounding Box.



Hình 3: Phân bố của tọa độ y

2.1.5 Phân tích sự đồng đều kích cỡ của các 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 "nlvnpf-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.

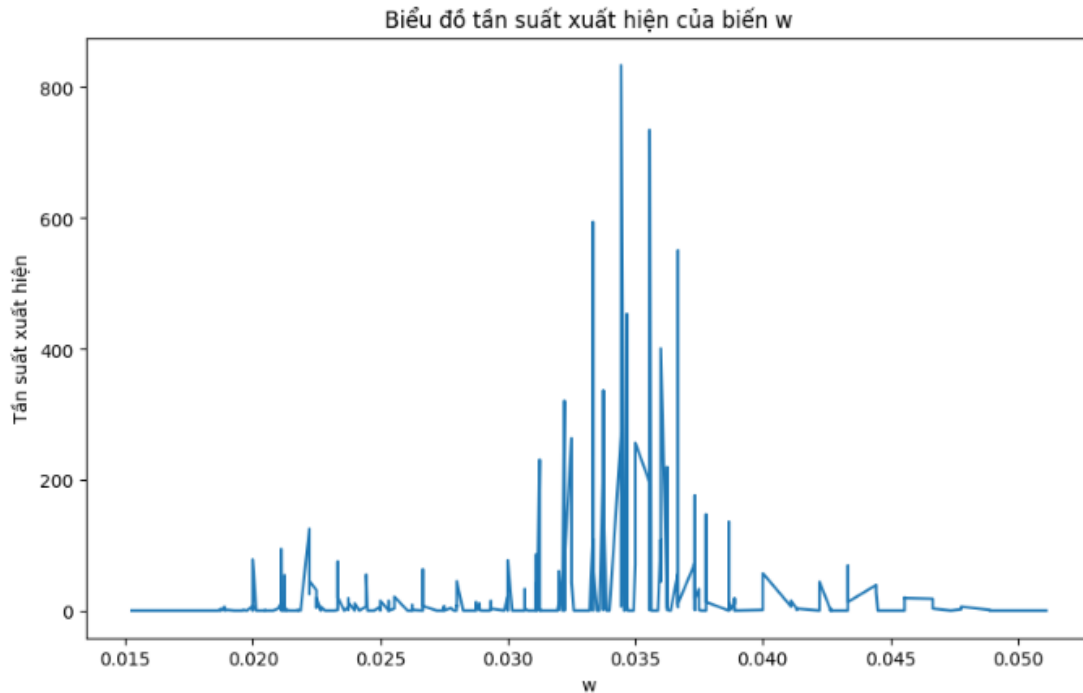
2.2 Tiền xử lý dữ liệu

Trong phần này, nhóm xin trình bày về việc tiền xử lý dữ liệu và tăng cường dữ liệu (augmentation) cho tập dữ liệu ảnh và nhãn.

2.2.1 Các kỹ thuật augmentation cơ bản

Nhóm đã sử dụng thư viện augmentation của Python 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 với giới hạn là 15 độ. 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.

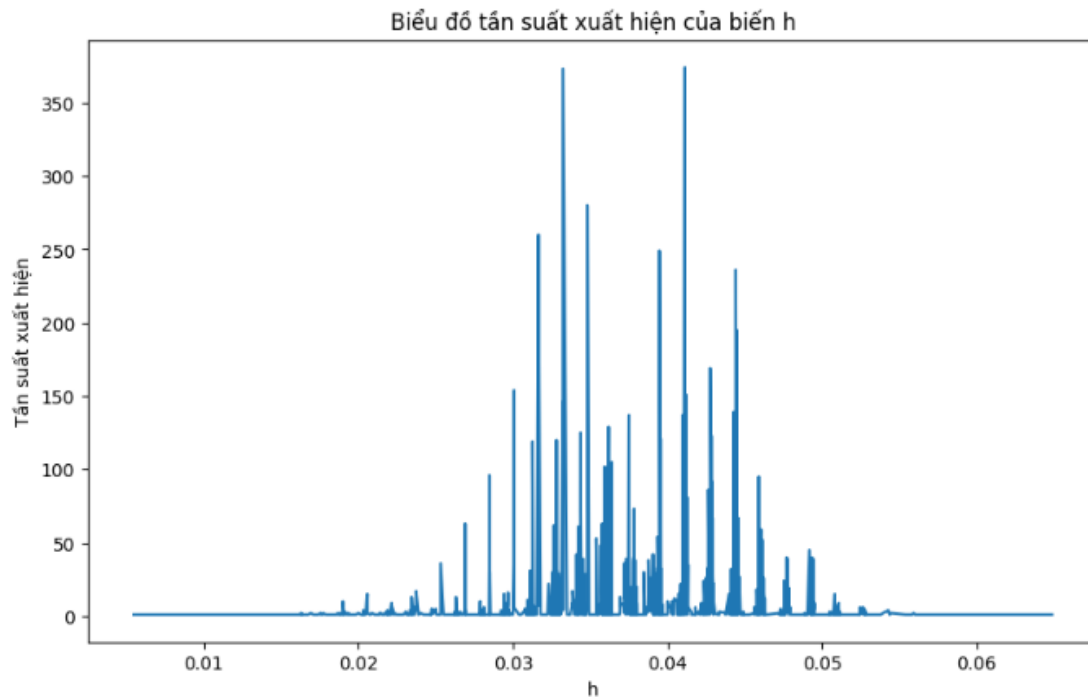


Hình 4: Phân bố của chiều rộng các BB

- Add noise: thực hiện lên bộ ảnh mà Hùng quyết định tập trung. 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 multiplicate noise
- 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.
- Dịch chuyển ảnh sang trái phải
- Crop ảnh chỉ show 1 phần (YOLO có sẵn nên 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 augmen trong cùng 1 bức ảnh Nhóm đã kết hợp nhiều kỹ thuật augmentation cơ bản trong cùng một bức ảnh. Bằng cách này có thể tạo ra các phiên bản mới có sự đa dạng và phong phú hơn.
- Brightness & Contrast Adjustment Nhóm cũng đã thực hiện điều chỉnh độ sáng và độ tương phản của ảnh. Điều này giúp tạo ra các phiên bản mới với độ tương phản và độ sáng khác nhau, từ đó làm tăng sự đa dạng của tập dữ liệu.
- Chuẩn hóa ảnh (không rõ YOLOv8 có làm hay không để Hà tìm hiểu)



Hình 5: Phân bố của chiều dài các BB

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