INT3404E 20 - Xử lý ảnh: Bài tập lớn

Nhóm 4 - Khoanh vùng ký tự Hán Nôm

1 Bài toán khoanh vùng ký tự Hán Nôm:

- Bài toán khoanh vùng ký tự Hán Nôm là một phần quan trọng trong việc xử lý, phân tích và số hóa tài liệu, văn bản viết tay hay in bằng mộc bản bằng chữ Nôm cổ.
- Mục tiêu của bài toán này là từ ảnh đầu vào là bản scan của văn bản cần xác định, khoanh vùng được các ký tự Hán Nôm có trong văn bản và đưa ra được vị trí tương đối của các kí tự này.
- Môt trong những khó khăn của bài toán này là sự đa dạng về cấu trúc và hình dáng của các ký tự Hán Nôm, chất lượng của văn bản viết tay gốc hay bản scan.

2 Phân tích và xử lí bộ dữ liệu:

2.1 Phân tích bộ dữ liệu gốc:

- Bộ dữ liệu gốc do thầy Thương cung cấp bao gồm 80 ảnh đã được gán nhãn, trong đó 70 ảnh được chia vào tập **train** và 10 ảnh được chia vào tập **validation**.
- Mỗi nhãn trong ảnh được cấu trúc theo định dạng sau:

label_id x_center y_center bbox_width bbox_height confident_score

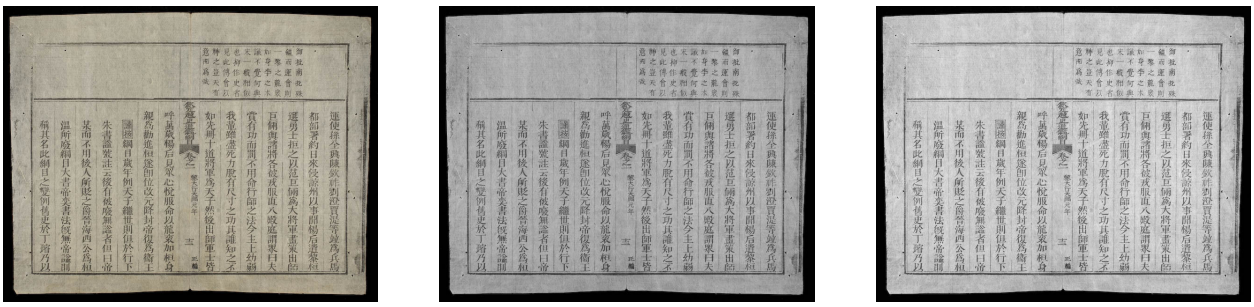
2.1.1 Tiền xử lí dữ liệu:

- Trong quá trình tìm hiểu các ảnh khắc in trên website của Dự án số hóa kho tàng thư tịch cổ văn hiến Hán Nôm[1], nhóm nhận thấy có những ảnh sẽ có màu sắc khác nhau như chữ viết bằng mực đỏ như ở hình 1 hay mực đen. Để đồng nhất về mặt màu sắc, tất cả ảnh đầu vào sẽ được chuyển từ ảnh màu thành ảnh **Grayscale**.

Hình 1: Ảnh có chữ được viết bằng mực đỏ.

- Sau khi chuyển thành ảnh **Grayscale**, nhóm sử dụng phương pháp **Contrast Stretching** nhằm tăng chất lượng hình ảnh đầu vào như thể hiện ở hình 2 dưới đây.

[1] https://lib.nomfoundation.org.



(a) Original Image (b) Grayscale Image (c) Contrast Stretching Image

Hình 2: Hình ảnh theo quy trình tiền xử lí dữ liệu của nhóm

2.1.2 Tăng cường dữ liệu:

Các phương pháp mà nhóm sử dụng bao gồm:

- Lật ảnh theo 2 trục ngang và dọc.
- Tăng giảm mức độ bão hòa (**Saturation**) của ảnh trong khoảng từ **-30%** đến **30%**.
- 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**.
- Trong quá trình tăng cường dữ liệu, nhóm sử dụng công cụ **Roboflow** với các thông số tăng cường được gợi ý đạt kết quả ổn định nhất từ công cụ này.

2.2 Bộ dữ liệu bổ sung:

2.2.1 Gán nhãn bằng công cụ Roboflow:

- Khi tiếp cận với bài toán, nhóm đánh giá thu thập và xử lí dữ liệu là một khía cạnh có thể mang ảnh hưởng quyết định đến kết quả huấn luyện của mô hình. Do đa số các bộ dữ liệu đều không có sẵn nhãn phù hợp, nhóm chọn phương án gán nhãn dữ liệu tự động kết hợp với thủ công để tạo ra một bộ dữ liệu bổ sung nhỏ. Ảnh gốc của tập dữ liệu lấy từ Dự án số hóa kho tàng thư tịch cổ văn hiến Hán Nôm[2].
- Nhóm sử dụng **Roboflow** để gán nhãn một tập dữ liệu gồm 30 ảnh được chọn lọc với những tính chất đặc trưng phù hợp với bộ dữ liệu gốc. Do quá trình gán nhãn này tốn rất nhiều thời gian nên số lượng 30 ảnh là mức công sức tối đa nhóm có thể đầu tư để xây dựng bộ dữ liệu bổ sung. Bộ dữ liệu này dự kiến được sử dụng để huấn luyện mô hình của nhóm.
- Nhóm không tiếp tục sử dụng tập dữ liệu này trong các mô hình và bộ dữ liệu huấn luyện sau khi việc gán nhãn dữ liệu thủ công chính thức bị cấm.

2.2.2 Bộ dữ liệu tiếng Hán:

- Trong quá trình tìm hiểu những bộ dữ liệu có sẵn nhãn phù hợp bài toán đặt ra, nhóm nhận thấy các bộ dữ liệu Hán tự có nhiều điểm tương đồng cả về bố cục lẫn tính chất đặc trưng của kí tự.

[2] https://lib.nomfoundation.org.

- Nhóm lựa chọn bộ dữ liệu Hán tự **MTHv2**[3] với 3199 ảnh chứa các kí tự tiếng Hán với nhãn dựa trên tọa độ pixel để thêm vào tập dữ liệu bổ sung. Bộ dữ liệu bao gồm 3 bộ dữ liệu con: **MTH1000**, **MTH1200** và **TKH**. Bộ dữ liệu **MTH1000** và **MTH1200** lần lượt chứa 1000 và 1200 ảnh phần lớn bao gồm ảnh scan bằng máy scan nên không được nhóm sử dụng. Bộ dữ liệu **TKH** bao gồm 999 ảnh đều là ảnh chụp từ máy ảnh nên được sử dụng để làm bộ dữ liệu bổ sung.

- Bộ dữ liệu **TKH** đã có tập nhãn theo tọa độ pixel của các bounding box. Để chuyển hóa tập nhãn này sang định dạng của YOLOv8 để sử dụng, nhóm đã xây dựng chương trình thông minh phục vụ chuyển đổi định dạng của nhãn từ định dạng gốc qua định dạng YOLOv8.

- Do bộ dữ liệu con **TKH** vẫn có số lượng ảnh quá lớn nên dữ liệu bổ sung được chọn lọc thủ công từ những ảnh có nhiều tính chất phù hợp với bộ dữ liệu gốc. Bộ dữ liệu bổ sung cuối cùng được sử dụng là 100 ảnh sau khi đã lọc. Các phương pháp tiền xử lí cho bộ dữ liệu này được áp dụng tương tự như bộ dữ liệu gốc.

3 Các mô hình sử dụng:

3.1 Mô hình YOLOv8n:

- YOLO[4] (You Only Look Once) là một thuật toán nhận diện đối tượng tiên tiến được giới thiệu vào năm 2016 bởi Joseph Redmon và cộng sự.

- YOLOv8 là phiên bản thứ 8 của mô hình YOLO. Mô hình này đã được đánh giá cao về tốc độ và độ chính xác trong bài toán nhận diện đối tượng và phân vùng hình ảnh trong thời gian thực. Cấu trúc mô hình YOLOv8 được thể hiện ở hình 3.

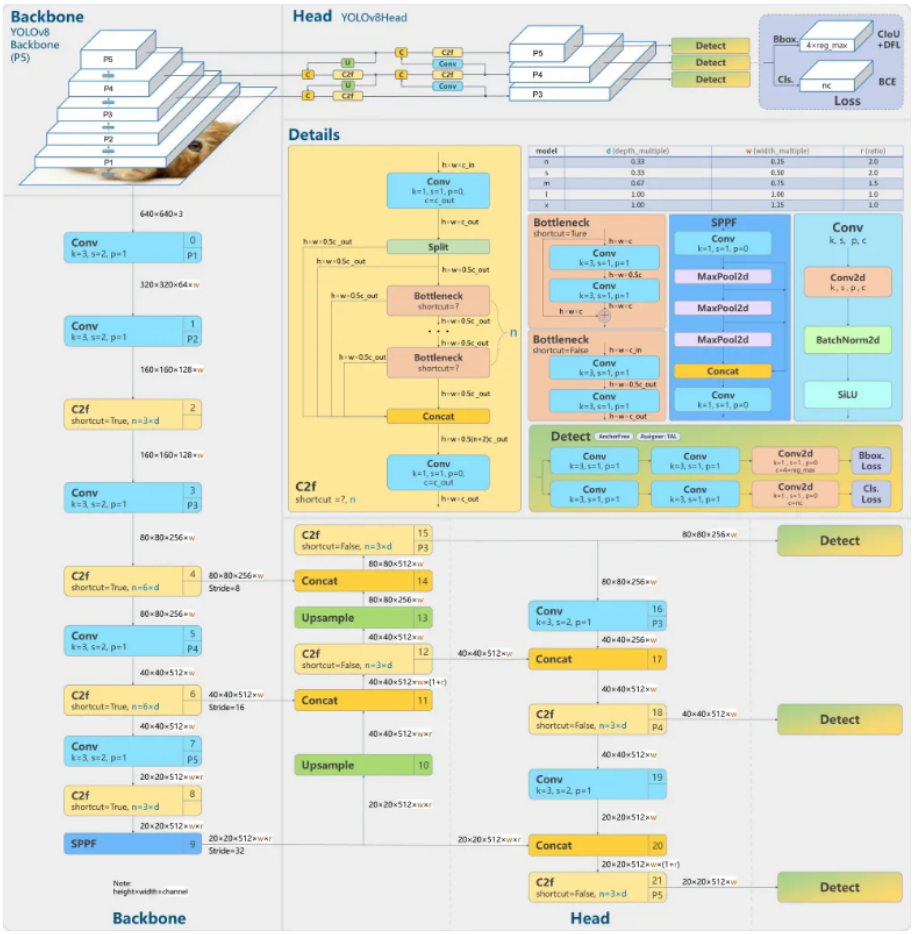
- YOLOv8n là một mô hình YOLOv8 nhưng có số lượng tham số nhỏ hơn. Do đó, YOLOv8n có thể được huấn luyện nhanh hơn trên Google Colab nhưng vẫn giữ được độ chính xác tốt. Đặc tính này giúp nhóm có thể thử nghiệm cùng một mô hình YOLOv8n nhưng với nhiều phương pháp xử lý dữ liệu khác nhau để đưa ra kết quả tốt nhất.

3.2 Các mô hình trong thư viện MMDetection:

- **MMDetection** là thư viện thị giác máy bao gồm các mô hình cho các bài toán nhận diện vật thể, phân vùng ngữ nghĩa (semantic segmentation) và phân vùng toàn cảnh (panoptic segmentation). Trong bài tập lớn này, nhóm đã sử dụng các mô hình cho bài toán nhận diện vật thể do bài toán này bao gồm một bài toán con là khoanh vùng kí tự (localization). Cụ thể hơn, qua phân tích, so sánh và tổng hợp, nhóm đã chọn được 4 mô hình phù hợp là Side-Aware Boundary Localization (SABL)[1], CentripetalNet [2], TOOD [3] và Deformable DETR [4].

- **Side-Aware Boundary Localization (SABL)** là một cách tiếp cận mới cho bài toán khoanh vùng (localization). Thay vì dự đoán bounding box dựa trên dự đoán tâm và kích thước (chiều dài và chiều rộng), SABL tiến hành dự đoán 4 cạnh của bounding box qua 4 nhánh riêng biệt trong mạng nơ-ron. Điều này được lấy ý tưởng từ việc khi ta đánh bounding box bằng tay, ta thường điều chỉnh 4 cạnh để tăng độ chính xác. Cụ thể hơn, SABL tiến hành 2 bước chính để xử lý bài toán khoanh vùng như mô tả trong hình 4. Đầu tiên, SABL chia chiều dài và chiều rộng của ảnh thành các khoảng (bucket), rồi tìm ra khoảng chính xác (bucket estimation). Sau

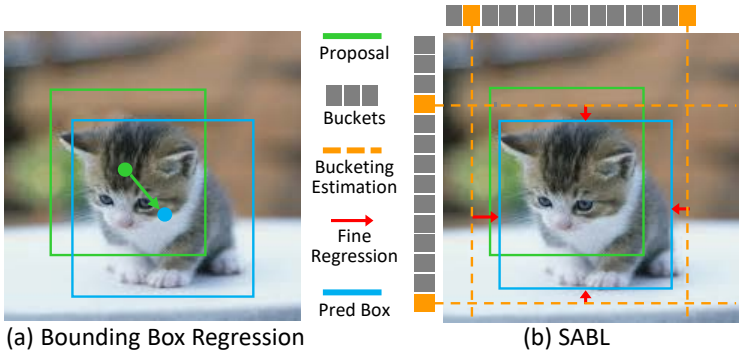
[3] https://github.com/HCIILAB/MTHv2_Datasets_Release.
[4] https://docs.ultralytics.com/.



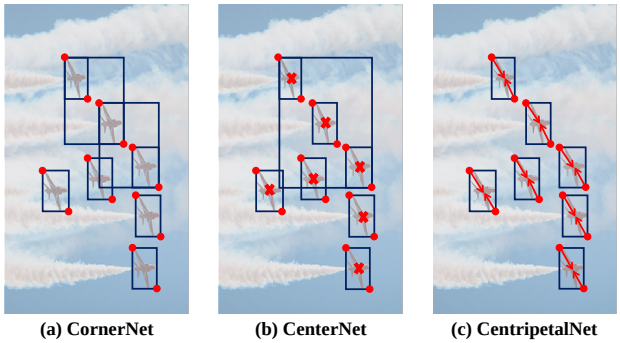
Hình 3: Cấu trúc mô hình YOLOv8.

đó, dựa vào đầu ra của bước thứ nhất là những đường nằm giữa của các khoảng này (các hình nét đứt trong hình), SABL tiến hành bước dự đoán sự sai khác với bounding box thực (fine regression). Hai bước này giúp đảm bảo độ chính xác của bài toàn khoanh vùng vật thể qua việc giải quyết vấn đề có sự sai khác lớn giữa anchor box và bounding box trong phương pháp truyền thống. Trong báo cáo này, nhóm sử dụng SABL dựa trên RetinaNet với backbone là ResNet-50 để giải quyết bài toán.

- **CentripetalNet** là một phương pháp nhận dạng vật thể dựa trên CornerNet [5]. CornerNet dự đoán bounding box dựa trên 2 bước: phát hiện các góc của bounding box và kết hợp 2 góc của cùng một bounding box để được một bounding box hoàn chỉnh. Bước kết hợp 2 góc được thực hiện qua việc học các biểu diễn nhúng (embedding) của các góc. Các biểu diễn nhúng nào giống nhau sẽ thuộc cùng một bounding box. Tuy nhiên, cách làm này có một số hạn chế như dễ bị tác động bất thường (outlier) hay các biểu diễn nhúng không có khả năng phân biệt các vật thể giống nhau những ở vị trí xa nhau. Do đó, CentripetralNet dự đoán độ dịch chuyển hướng tâm (centripetal shift), một vector 2D biểu diễn khoảng cách từ 2 góc đến tâm. Mỗi góc sẽ được dùng để dự đoán một tâm tương ứng. Hai góc nào có hai tâm dự đoán tương ứng gần nhau sẽ thuộc cùng một bounding box. Hình 5 so sánh phương pháp CentripetalNet và một số phương pháp tương tự.

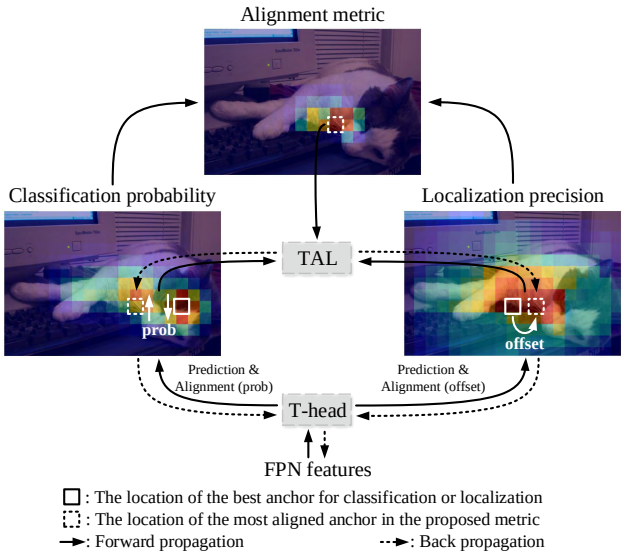


Hình 4: So sánh SABL và phương pháp thường dùng cho bài toàn khoanh vùng vật thể. Hình lấy từ bài báo gốc.

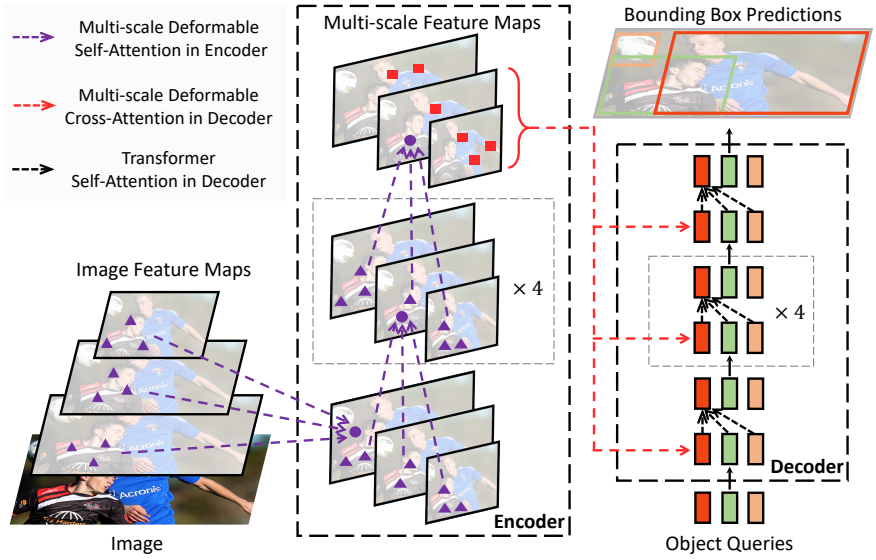


Hình 5: So sánh CentripetalNet và các phương pháp nhận dạng vật thể dựa trên keypoint khác. Hình lấy từ bài báo gốc.

- **TOOD: Task-aligned One-stage Object Detection** là một phương pháp nhận dạng vật thể trong đó các tác giả đã đề xuất 2 khái niệm mới là task-aligned head và task-alignnment learning. Task-aligned head tính toán các đặc trưng trong sự tương tác của tác vụ: khoanh vùng (localization) và phân loại (classification). Điều này giúp giải quyết vấn đề không có sự liên kết giữa hai tác vụ này. Task-alignment learning tăng cường khả năng thống nhất (alignment) giữa hai tác vụ này qua việc đánh giá độ alignment trên từng anchor, sau đó sử dụng task-aligned loss để hợp nhất những anchor được chọn để đưa ra kết quả cho cả hai tác vụ. Hình 6 minh họa sự kết hợp giữa task-aligned head và task-alignment learning để nhận dạng vật thể.
- **Deformable DETR**: Deformable DETR là một sự cải tiến dựa trên mô hình DETR [6]. DETR là mô hình nhận dạng vật thể đầu cuối đầu tiên dựa trên mạng nơ-ron tích chập và transformer. Tuy nhiên, DETR gặp phải hai vấn đề chính: hội tụ chậm và có hiệu suất thấp trong nhận dạng vật thể nhỏ. Deformable DETR giải quyết vấn đề này bằng cách kết hợp ý tưởng từ deformable convolution và transformer. Các tác giả đã đề xuất deformable attention module thay cho cơ chế attention thông thường, trong đó chỉ "chú ý" đến các vị trí quan trọng trong bản đồ đặc trưng (feature map). Quá trình nhận dạng vật thể chi tiết được mô tả trong hình 7.



Hình 6: Quá trình nhận dạng vật thể của TOOD. Hình lấy từ bài báo gốc.



Hình 7: Quá trình nhận dạng vật thể của Deformable DETR. Hình lấy từ bài báo gốc.

4 Thực nghiệm:

4.1 Mô hình YOLOv8n

Nhóm đã tiến hành thực nghiệm với mô hình YOLOv8n với các siêu tham số trong bảng 1 trên các tập dữ liệu như sau:

- **Baseline**: Sử dụng tập train gốc gồm 70 ảnh, tập validation gốc gồm 10 ảnh do thầy Thương cung cấp.
- **Kết hợp Tiền xử lí và Tăng cường dữ liệu**: Sử dụng tập train đã qua tiền xử lí và tăng cường dữ liệu.
- **Sử dụng bộ dữ liệu bổ sung**: Bổ sung 100 ảnh từ bộ dữ liệu tiếng Hán MTHv2 vào tập train gốc.

Siêu tham số	**Giá trị**
Epoch	100
Batch size	7
Learning rate	0.01
Conf	0.5

Bảng 1: Các siêu tham số được sử dụng trong mô hình YOLOv8n.

4.2 Các mô hình trong thư viện MMDetection

Nhóm chỉ tiến hành thực nghiệm các mô hình trong thư viện MMDetection trong tập dữ liệu đã được tiền xử lí và tăng cường dữ liệu do giới hạn của tài nguyên tính toán. Các siêu tham số sử dụng trong huấn luyện được trình bày trong bảng 2.

Mô hình	**Epoch**	**Learning rate**	**Batch size**	**Optimizer**
SABL RetinaNet R50	20	0.01	4	SGD
TOOD R50	20	0.01	2	SGD
CentripetalNet	20	0.0005	4	Adam
Deformable DETR	20	0.00008	1	AdamW

Bảng 2: Các siêu tham số sử dụng trong MMDetection.

5 Kết quả:

Mô hình	**mAP@[.5:.95]**
Baseline	0.771
YOLOv8n (Tiền xử lí & Tăng cường)	**0.872**
YOLOv8n sử dụng bộ dữ liệu bổ sung (Base)	0.863
YOLOv8n sử dụng bộ dữ liệu bổ sung (Tiền xử lí dữ liệu)	0.863
YOLOv8n sử dụng bộ dữ liệu bổ sung (Tiền xử lí & Tăng cường dữ liệu)	0.821

Bảng 3: Kết quả thử nghiệm mô hình YOLOv8n với các bộ dữ liệu khác nhau. Điểm mAP được tính bằng code do thầy Thương cung cấp.

Mô hình	**mAP@[.5:.95]**
SABL RetinaNet	0.841
TOOD R50	0.842
CentripetalNet	**0.873**
Deformable DETR	0.669

Bảng 4: Kết quả thực nghiệm các mô hình trong thư viện MMDetection. Điểm mAP được tính bằng code do thầy Thương cung cấp.

- Mô hình YOLOv8n được tối ưu cho bài toán **Real-time Detection**, tuy nhiên các mô hình được chọn trong thư viện MMDetection có phương pháp đánh bounding box khác và tối ưu hơn cho bài toán **Localization**. Do vậy mô hình cuối cùng mà nhóm sử dụng để đánh giá kết quả bài tập lớn này là mô hình **CentripetalNet** của thư viện **MMDetection**.

- Mô hình Deformable DETR cho ra kết quả thấp nhất so với các mô hình còn lại, chỉ đạt được 0.669 trong khi các mô hình còn lại đều đạt được 0.7+. Điều này có thể là do việc huấn luyện chưa đạt đủ epoch do hạn chế về mặt tài nguyên tính toán. Ngoài ra, các siêu tham số khác cũng chưa được tối ưu, ảnh hưởng đến kết quả cuối cùng.

- Trong quá trình huấn luyện các mô hình, khi sử dụng dữ liệu bổ sung đã qua tăng cường, cả mô hình YOLOv8n và các mô hình trong thư viện MMDetection đều bị vượt quá giới hạn tài nguyên trên 2 nền tảng nhóm sử dụng là **Google Colab** và **Lightning AI**. Do vậy kết quả cuối cùng nhóm đạt được chưa phải là kết quả hoàn thiện nhất với những mô hình này và nhóm **không sử dụng các mô hình được huấn luyện với bộ dữ liệu bổ sung**.

- Trong buổi đánh giá kết quả bài tập lớn trên lớp, nhóm sử dụng mô hình **CentripetalNet** và đạt kết quả **mAP@[.5:.95] = 0.8585** trên tập test của thầy Thương.

- Sau khi trao đổi với các nhóm có kết quả mô hình tốt hơn, nhóm nhận thấy quá trình tăng cường dữ liệu của nhóm chưa phủ được nhiều trường hợp trong tập test của thầy Thương dẫn tới kết quả không ổn định ở ngưỡng điểm **mAP@[.8:.95]**.

6 Kết luận:

Trong quá trình phân tích và giải quyết bài toán, nhóm đã đưa ra được nhiều phương án để tiếp cận và cải tiến kết quả của mô hình. Bằng việc so sánh các phương án dựa trên thực nghiệm, những phương án được nhóm đào sâu tìm hiểu đã có những kết quả nổi bật so với các cách tiếp cận trực diện khác như sử dụng mô hình YOLO đơn thuần. Tuy nhiên, với số lượng dữ liệu và mô hình cần được xử lí lớn, nhóm đã bỏ sót một vài hướng tiếp cận có thể ảnh hưởng tích cực đến kết quả của mô hình cuối cùng được sử dụng. Dù vậy, những phương án mà nhóm đã triển khai đã thể hiện được tiềm năng qua quá trình tự đánh giá lẫn kết quả đánh giá cuối cùng và hoàn toàn có thể được cải thiện để đạt kết quả tốt hơn trong tương lai.

References

[1] Jiaqi Wang, Wenwei Zhang, Yuhang Cao, Kai Chen, Jiangmiao Pang, Tao Gong, Jianping Shi, Chen Change Loy, and Dahua Lin. *Side-Aware Boundary Localization for More Precise Object Detection*. 2020. arXiv: 1912.04260 [cs.CV].

[2] Zhiwei Dong, Guoxuan Li, Yue Liao, Fei Wang, Pengju Ren, and Chen Qian. *CentripetalNet: Pursuing High-quality Keypoint Pairs for Object Detection*. 2020. arXiv: 2003.09119 [cs.CV].

[3] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R. Scott, and Weilin Huang. *TOOD: Task-aligned One-stage Object Detection*. 2021. arXiv: 2108.07755 [cs.CV].

[4] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. 2021. arXiv: 2010.04159 [cs.CV].

[5] Hei Law and Jia Deng. *CornerNet: Detecting Objects as Paired Keypoints*. 2019. arXiv: 1808.01244 [cs.CV].

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].