VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

BÁO CÁO ĐỒ ÁN TỔNG HỢP

TỰ ĐỘNG TRÍCH XUẤT THÔNG TIN VĂN BẢN VÀ SỐ TỪ HÌNH ẢNH BIỂU ĐỒ

Advisor(s): Mai Xuân Toàn
Trần Tuấn Anh
Huỳnh Văn Thống
Trần Hồng Tài

Student(s): Lê Trần Tấn Phát MSSV 2312580
Bùi Ngọc Phúc MSSV 2312665
Nguyễn Hồ Quang Khải MSSV 2352538

HO CHI MINH CITY, DECEMBER 2025

Mục lục

Danh sách hình vẽ

Danh sách bảng

1 Giới thiệu

1.1 Đặt vấn đề

Trong kỷ nguyên dữ liệu lớn (Big Data) hiện nay, thông tin không chỉ tồn tại dưới dạng văn bản thuần túy mà còn được trình bày đa dạng qua các hình thức trực quan hóa, trong đó biểu đồ (chart) đóng vai trò then chốt. Biểu đồ giúp tóm tắt dữ liệu phức tạp, làm nổi bật các xu hướng và hỗ trợ ra quyết định hiệu quả trong nhiều lĩnh vực như tài chính, kinh doanh, y tế và nghiên cứu khoa học.

Tuy nhiên, máy tính thường gặp khó khăn trong việc "đọc hiểu" dữ liệu từ hình ảnh biểu đồ so với văn bản thông thường. Việc trích xuất lại dữ liệu gốc (số liệu, nhãn, chú thích) từ ảnh biểu đồ hiện nay phần lớn vẫn được thực hiện thủ công. Quy trình này không chỉ tốn kém thời gian, công sức mà còn dễ xảy ra sai sót con người, đặc biệt khi phải xử lý một lượng lớn tài liệu lưu trữ hoặc báo cáo định kỳ.

Xuất phát từ nhu cầu thực tiễn đó, việc nghiên cứu và phát triển hệ thống "Tự động trích xuất thông tin văn bản và số từ hình ảnh biểu đồ" là vô cùng cấp thiết. Một hệ thống tự động hóa hiệu quả sẽ giúp số hóa dữ liệu từ các báo cáo dạng ảnh, hỗ trợ người khiếm thị tiếp cận thông tin, và làm tiền đề cho các bài toán phân tích dữ liệu nâng cao.

1.2 Mục tiêu nghiên cứu

Đề tài tập trung giải quyết bài toán chuyển đổi dữ liệu phi cấu trúc (hình ảnh biểu đồ) thành dữ liệu có cấu trúc (bảng, JSON). Các mục tiêu cụ thể bao gồm:

- Trích xuất thành phần cơ sở: Tự động phát hiện và nhận diện chính xác các vùng chứa văn bản (tiêu đề, nhãn trục, chú thích) và các thành phần đồ họa (cột, đường, điểm dữ liệu).
- Phân loại vai trò ngữ nghĩa: Xác định rõ vai trò của từng đoạn văn bản trong biểu đồ (đâu là tiêu đề, đâu là giá trị trục, đâu là chú thích) sử dụng các mô hình học sâu hiện đại.
- Tổng hợp và liên kết dữ liệu: Xây dựng thuật toán để ghép nối thông tin văn bản với thông tin hình học, từ đó khôi phục lại bảng số liệu gốc với độ chính xác cao.

- Tính thực tiễn: Tối ưu hóa hệ thống để có thể ứng dụng vào các quy trình tự động hóa xử lý tài liệu thực tế.

1.3 Thách thức

Việc giải quyết bài toán trích xuất thông tin từ biểu đồ đối mặt với nhiều thách thức kỹ thuật lớn:

- Sự đa dạng của biểu đồ: Có vô số loại biểu đồ (cột, đường, tròn, hỗn hợp) với phong cách thiết kế, màu sắc và bố cục khác nhau.
- Cấu trúc phức tạp: Mối quan hệ giữa các thành phần không cố định (ví dụ: chú thích có thể nằm dưới, nằm phải, hoặc nằm trong biểu đồ).
- Chất lượng hình ảnh: Ảnh đầu vào từ thực tế thường có độ phân giải thấp, bị mờ, nhiễu, hoặc chứa các yếu tố gây nhiễu như watermark, lưới nền (grid lines).
- Độ chồng lấn: Các nhãn dữ liệu hoặc các đường biểu diễn thường xuyên bị chồng lấn lên nhau, gây khó khăn cho việc nhận diện và tách biệt.

1.4 Phạm vi nghiên cứu

Để đảm bảo tính khả thi và tập trung sâu vào chất lượng giải pháp, nhóm giới hạn phạm vi nghiên cứu như sau:

- Dữ liệu đầu vào: Tập trung vào các loại biểu đồ phổ biến nhất trong báo cáo tài chính và khoa học: Biểu đồ cột (Bar chart)
- Dữ liệu đầu ra: Bảng số liệu (CSV/Excel) chứa các thông tin: Tiêu đề biểu đồ, tên các trục, và các cặp giá trị (x, y) hoặc (category, value) tương ứng.
- Yêu cầu kỹ thuật: Hệ thống cần đạt độ chính xác cao về mặt OCR (nhận diện chữ) và Object Detection (phát hiện đối tượng), đồng thời tối ưu hóa thời gian xử lý.

1.5 Phương pháp tiếp cận

Dựa trên các thách thức đã nêu, nhóm đề xuất phương pháp tiếp cận theo hướng Multi-stage Deep Learning (Học sâu đa giai đoạn), kết hợp giữa Computer Vision (Thị giác máy tính) và NLP (Xử lý ngôn ngữ tự nhiên):

- Xử lý ảnh và Object Detection: Sử dụng mô hình `YOLO` để phát hiện các thành phần đồ họa (thanh, điểm, đường) và vùng chứa văn bản.
- Nhận dạng ký tự quang học (OCR): Ứng dụng `PaddleOCR` để chuyển đổi vùng hình ảnh chứa chữ thành văn bản máy tính, đảm bảo khả năng đọc tốt các văn bản đa hướng và kích thước nhỏ.
- Sử dụng mô hình `LayoutLMv3` – một mô hình Transformer đa phương thức (Multimodal) tiên tiến – để phân loại vai trò của văn bản dựa trên cả nội dung, vị trí và hình ảnh.
- Sử dụng các thuật toán hình học và logic để ghép nối các thành phần đã nhận diện, từ đó tái tạo lại dữ liệu số cuối cùng.

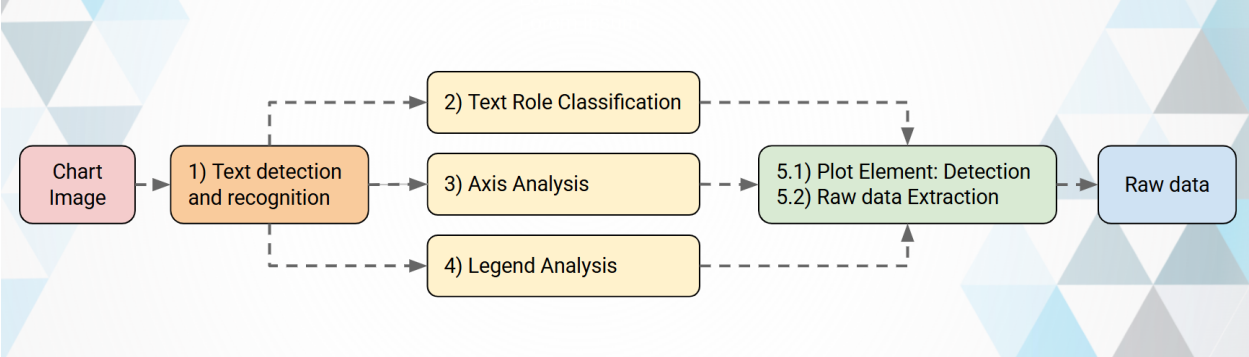
2 Tổng quan kiến trúc hệ thống

Bài toán *trích xuất thông tin từ biểu đồ* (Chart Information Extraction) nhằm mục tiêu chuyển đổi dữ liệu hình ảnh (raster image) thành biểu diễn dữ liệu có cấu trúc. Đầu ra của hệ thống bao gồm ba nhóm thành phần chính: (i) thông tin văn bản (tiêu đề, nhãn trục, nhãn dữ liệu, chú giải); (ii) các đối tượng đồ họa (cột, trục, vùng chú giải); và (iii) dữ liệu số (giá trị định lượng tương ứng với từng cột và chuỗi dữ liệu).

Trong phạm vi nghiên cứu này, hệ thống tập trung xử lý đối tượng **biểu đồ cột** (Bar Chart) với quy trình xử lý được chia thành 5 giai đoạn chính: (1) Text Detection/Recognition, (2) Text Role Classification, (3) Axis Analysis, (4) Legend Analysis, và (5) Data Extraction. Phần này sẽ tổng quan các phương pháp tiếp cận hiện có tương ứng với từng mô-đun chức năng.

2.1 Text Detection and Recognition

Khác với bài toán phát hiện văn bản trong khung cảnh tự nhiên (Scene Text Detection), mục tiêu trong xử lý biểu đồ không phải là phát hiện các từ đơn lẻ, mà là nhận diện các **khối văn bản** (text blocks) mang ý nghĩa ngữ nghĩa trọn vẹn (ví dụ: một tiêu đề trục hoàn chỉnh, một dòng chú giải). Dữ liệu biểu đồ thường có nền đơn giản hơn nhưng đối mặt với thách thức về kích thước vùng chữ rất nhỏ, sự chồng lấn với các phần tử đồ họa, và sự đa



Hình 2.1: Sơ đồ tổng quan quy trình trích xuất thông tin biểu đồ

dạng của các ký tự đặc biệt.

Quy trình xử lý thường được chia thành hai bước hoặc tích hợp trong mô hình End-to-end:

- **Text Detection (Phát hiện)**: Các phương pháp hiện đại tập trung vào việc định vị các vùng văn bản liền mạch (coherent text regions). Thách thức lớn nằm ở sự biến thiên kích thước vùng chữ, từ một ký tự đơn lẻ đến các đoạn văn bản đa dòng.
- **Text Recognition (Nhận dạng)**: Để đảm bảo tính chính xác của dữ liệu khoa học, việc chuyển đổi các ký hiệu đặc biệt (ví dụ: công thức LaTeX) sang mã Unicode là bắt buộc. Các mô hình nhận dạng thông thường thường gặp khó khăn với các chuỗi ký tự quá dài hoặc bị xoay nghiêng. Một giải pháp hiệu quả được áp dụng là sử dụng các quy luật hình học để phát hiện các ứng viên văn bản dài/đa dòng, sau đó áp dụng thuật toán tham lam (greedy algorithms) để cắt ảnh theo chiều ngang/dọc, nhận dạng từng phần và ghép nối kết quả.

2.2 Text Role Classification

Text Role Classification (Phân loại vai trò văn bản) là bước trung gian thiết yếu nhằm gán nhãn ngữ nghĩa cho từng khối văn bản đã được phát hiện. Hệ thống xem xét 9 vai trò cụ thể: `chart title`, `axis title`, `tick label`, `tick grouping`, `legend title`, `legend label`, `value label`, `marker label`, và `other`

Thách thức lớn nhất của tác vụ này bao gồm:

- **Sự phụ thuộc vào ngữ cảnh**: Hai khối văn bản có nội dung giống hệt nhau có thể mang vai trò khác nhau tùy thuộc vào vị trí của chúng trong bố cục biểu đồ.
- **Mất cân bằng dữ liệu nghiêm trọng (Class Imbalance)**: Thống kê cho thấy *tick labels* chiếm tỉ trọng lớn (khoảng 70% số lượng mẫu), trong khi các lớp quan trọng khác như *chart title* hay *legend title* xuất hiện rất thưa thớt (tổng cộng chỉ khoảng 1%). Tuy nhiên, sai số ở bất kỳ lớp nào cũng ảnh hưởng trực tiếp đến độ chính xác của quá trình tái cấu trúc dữ liệu cuối cùng.

2.3 Axis Analysis

Mục tiêu của Axis Analysis là định vị các trục chính (ngang và dọc) và liên kết các điểm dữ liệu trên trục (ticks) với các nhãn văn bản (tick labels) tương ứng. Điểm khác biệt quan trọng trong bài toán này là hệ thống cần xác định được vị trí của các mốc giá trị ngay cả khi biểu đồ `không hiển thị vạch chia` (visual tick marks). Các phương pháp tiếp cận cần suy luận cấu trúc trục dựa trên sự thẳng hàng và phân bố của các *tick labels*, thay vì chỉ dựa vào các đặc trưng hình ảnh của đường kẻ hay vạch chia. Kết quả của bước này là cơ sở để xây dựng thang đo (scale), cho phép chuyển đổi tọa độ pixel sang giá trị thực.

2.4 Legend Analysis

Legend Analysis giải quyết bài toán xác định các cặp thành phần chú giải (Legend Entries), bao gồm: *nhãn chú giải* (Legend Label) và *biểu tượng mẫu* (Legend Symbol/Patch). Quy trình thường bắt đầu bằng việc phát hiện các ứng viên (candidates) cho cả nhãn và biểu tượng, sau đó thực hiện ghép cặp

Do tính chất nhiễu của đầu ra mô hình phát hiện đối tượng (Object Detection), việc áp dụng các thuật toán ghép cặp đồ thị nhị phân (bipartite graph matching) tiêu chuẩn thường không hiệu quả. Một chiến lược ghép cặp dựa trên hình học (Geometric Matching) được đề xuất như sau:

1. **Ước lượng hướng bố cục**: Xác định vị trí tương đối phổ quát của biểu tượng so với nhãn trong toàn bộ biểu đồ (ví dụ: biểu tượng nằm bên trái/phải/trên/dưới nhãn).
2. **Ghép cặp tham lam (Greedy Matching)**: Dựa trên hướng đã xác định, tính toán khoảng cách từ điểm tham chiếu của nhãn đến tâm của biểu tượng. Các cặp có khoảng cách nhỏ nhất sẽ được ưu tiên ghép trước.

3. **Kiểm tra ràng buộc**: Để loại bỏ các liên kết sai (spurious matches) do dương tính giả/âm tính giả, hệ thống chỉ chấp nhận các cặp chưa được gán và dừng lại ngay khi xuất hiện sự bất thường về khoảng cách so với các cặp đã hình thành, dựa trên giả định rằng các chú giải trong cùng một biểu đồ thường có khoảng cách căn chỉnh đồng nhất.

2.5 Data Extraction

Giai đoạn Data Extraction tổng hợp thông tin từ các bước trước để tái tạo bảng dữ liệu gốc. Quy trình này chia thành hai tác vụ con:

2.5.1 Plot Element Detection

Nhiệm vụ là định vị chính xác các thành phần biểu diễn dữ liệu (data marks), cụ thể là các thanh (bars) trong biểu đồ cột. Các bộ phát hiện đối tượng (Object Detectors) thông thường thường gặp khó khăn với các thanh có *tỉ lệ khung hình cực đoan* (quá hẹp hoặc quá dài) do giới hạn của các neo kích thước (anchors). Ngoài ra, sự đa dạng trong phong cách hiển thị (màu sắc đơn sắc so với các mẫu texture phức tạp) cũng là một trở ngại đáng kể cho các thuật toán thị giác máy tính.

2.5.2 Raw Data Extraction

Đây là bước tích hợp cuối cùng để suy luận giá trị. Đối với biểu đồ cột, quy trình bao gồm:

- **Ánh xạ Categories (Trục X)**: Gán mỗi thanh vào một nhãn phân loại dựa trên tọa độ. Với biểu đồ nhóm (Grouped Bar Chart), cần thêm bước phân tích khoảng cách để gom nhóm các thanh thuộc cùng một cụm.
- **Ánh xạ Series (Chuỗi dữ liệu)**: Xác định thanh thuộc chuỗi dữ liệu nào dựa trên kết quả của Legend Analysis và đặc trưng thị giác (màu/texture).
- **Suy luận giá trị (Trục Y)**: Chuyển đổi chiều cao thanh thành giá trị số dựa trên thang đo từ Axis Analysis. Hệ thống cần xử lý các trường hợp phức tạp như biểu đồ chồng (Stacked Bar Chart - cần xác định hai điểm đầu mút để tính giá trị) hoặc các trường hợp giá trị bằng 0 (dẫn đến các thanh "vô hình" trên ảnh nhưng vẫn tồn tại trong logic dữ liệu).

Cần lưu ý rằng trong các hệ thống dạng module (pipelined systems), lỗi từ các giai đoạn sớm (như OCR hoặc Legend Analysis) sẽ lan truyền (propagate) và khuếch đại ở bước trích xuất dữ liệu này, làm giảm độ chính xác tổng thể.

3 Phương pháp đề xuất

3.1 Phát hiện và Nhận dạng Văn bản

Giai đoạn đầu tiên có nhiệm vụ trích xuất tất cả các đoạn văn bản (text snippets) và vị trí chính xác của chúng từ ảnh biểu đồ. Giai đoạn này sử dụng một kiến trúc lai (Hybrid Architecture), kết hợp sức mạnh của `YOLO` (để đảm bảo độ phủ cao trong phát hiện) và `PaddleOCR` (để nhận dạng ký tự hiệu suất cao).

3.1.1 Kiến trúc Lai Phát hiện Văn bản (Hybrid Text Detection)

Mặc dù `PaddleOCR` đã bao gồm một module phát hiện văn bản, nhưng để đối phó với sự đa dạng và thách thức về kích thước, góc độ, và độ nhiễu trong dữ liệu biểu đồ (như đã phân tích ở Mục 4.1), chúng tôi đã quyết định tích hợp thêm một lớp phát hiện độc lập bằng `YOLO`

Bởi vì `YOLO` (You Only Look Once) là một mô hình phát hiện đối tượng nổi tiếng với tốc độ và độ chính xác cao. Việc huấn luyện `YOLO` trên một tập dữ liệu biểu đồ được gán nhãn cho các vùng văn bản sẽ giúp:

- Tăng tính mạnh mẽ: `YOLO` có khả năng phát hiện tốt các vùng văn bản nhỏ, bị nghiêng, hoặc nằm trên nền phức tạp (như lưới biểu đồ), những trường hợp mà các mô hình detection chung của `OCR` có thể bỏ sót.
- Chuẩn hóa đầu ra: Đảm bảo đầu ra là một tập hợp các bounding box tứ giác chuẩn xác, sẵn sàng cho bước nhận dạng.

Quá trình huấn luyện `YOLO` diễn ra như sau:

- Huấn luyện mô hình `YOLO` (khuyến nghị dùng các phiên bản mới như `YOLOv8`) trên tập dữ liệu biểu đồ đã được gán nhãn cho lớp "Text Bounding Box"
- Đầu vào là ảnh biểu đồ, đầu ra là danh sách các bounding box $(x_{min}, y_{min}, x_{max}, y_{max})$ bao quanh các vùng chứa văn bản.

3.1.2 Nhận dạng Ký tự (Text Recognition) bằng PaddleOCR

Sau khi có được tọa độ chính xác của từng vùng văn bản, module nhận dạng của PaddleOCR sẽ được sử dụng để chuyển đổi hình ảnh thành văn bản kỹ thuật số.

Model/Engine: Chúng tôi sử dụng PaddleOCR, cụ thể là mô hình PP-OCRv4 (hoặc phiên bản phù hợp nhất với tài nguyên).

Lý do lựa chọn PaddleOCR:

- Độ chính xác cao: PaddleOCR có hiệu suất SOTA trong nhận dạng ký tự tiếng Anh và các ký tự đặc biệt thường thấy trong biểu đồ (số, đơn vị).
- Tốc độ: Tốc độ suy luận (Inference speed) nhanh, phù hợp cho việc xử lý hàng loạt tài liệu.
- Không cần huấn luyện (Zero-shot): Do PaddleOCR được huấn luyện trên một lượng lớn dữ liệu đa dạng, nhóm không cần phải huấn luyện lại module nhận dạng mà chỉ cần sử dụng mô hình đã được huấn luyện sẵn (Pre-trained Model).

Quá trình:

1. Mỗi bounding box (là kết quả từ YOLO) được cắt ra khỏi ảnh gốc (Image Cropping).
2. Vùng ảnh cắt ra được đưa vào module nhận dạng của PaddleOCR.
3. PaddleOCR trả về chuỗi văn bản (Text String) và độ tin cậy (Confidence Score).

3.2 Phân loại Vai trò Ngữ nghĩa Văn bản

Giai đoạn này đóng vai trò cầu nối quan trọng giữa kết quả của OCR (văn bản thô) và việc tổng hợp dữ liệu cuối cùng. Mục tiêu là xác định vai trò ngữ nghĩa (Senmantic Role) củ mỗi giai đoạn văn bản được trích xuất từ biểu đồ

3.2.1 Giới thiệu Model LayoutLMv3

Trong các bài toán Phân tích Cấu trúc Tài liệu (Document Structure Analysis), các mô hình NLP truyền thống chỉ xử lý chuỗi văn bản, bỏ qua thông tin về vị trí và hình ảnh. LayoutLMv3 là một mô hình Transformer đa phương thức (Multi-Modal Transformer) tiên tiến, được lựa chọn vì những ưu điểm sau:

- Kết hợp ba yếu tố: `LayoutLMv3` học biểu diễn bằng cách kết hợp đồng thời Văn bản (Text), Hình ảnh (Image) và Thông tin Bố cục (Layout Position) (tọa độ bounding box). Điều này cho phép mô hình không chỉ đọc được nội dung mà còn hiểu được ngữ cảnh không gian của văn bản trong biểu đồ.
- Hiểu Biểu đồ (Chart Understanding): Do cấu trúc của biểu đồ rất phụ thuộc vào vị trí (ví dụ: tiêu đề luôn ở trên cùng), việc sử dụng thông tin bố cục là bắt buộc để phân biệt một số (Label) với một giá trị (Value).
- Hiệu suất: `LayoutLMv3` đã chứng minh hiệu suất vượt trội so với các mô hình chỉ dựa trên văn bản hoặc chỉ dựa trên hình ảnh trong các tác vụ hiểu tài liệu phức tạp.

Kiến trúc `LayoutLMv3` được xây dựng trên cơ sở `Transformer Encoder`, sử dụng cơ chế tự chú ý (Self-Attention) để tích hợp ba loại đầu vào:

1. Text Embedding: Biểu diễn vector của token văn bản.
2. Layout Embedding: Biểu diễn không gian 2D (tọa độ $x_{min}, y_{min}, x_{max}, y_{max}$ đã được chuẩn hóa) của bounding box chứa token đó.
3. Image Embedding: Vector đặc trưng hình ảnh trích xuất từ vùng xung quanh token (Patch Embedding).

$$\text{Embedding} = \text{Text\_Emb} + \text{Layout\_Emb} + \text{Image\_Emb}$$

Tất cả các embedding này được đưa vào các lớp Transformer để học mối quan hệ tương tác phức tạp giữa nội dung và vị trí.

3.2.2 Thiết lập Tác vụ Phân loại Vai trò (Role Classification Setup)

LayoutLMv3 được tinh chỉnh (Fine-tuned) trên bộ dữ liệu biểu đồ đã được gán nhãn để thực hiện tác vụ phân loại nhãn.

Chúng tôi sử dụng 9 lớp phân loại chi tiết như sau. Việc phân biệt rõ các lớp này, đặc biệt là các nhãn liên quan đến số liệu (`Tick_Label`, `Mark_Label` , `Value_Label`), là chìa khóa để thuật toán ghép nối ở Giai đoạn 3 hoạt động hiệu quả.

- `Chart_Title` là tiêu đề chính, tóm tắt nội dung biểu đồ

- `Axis_Title` là nhãn mô tả ý nghĩa của trục hoành (X) hoặc trục tung (Y)
- `Tick_Label` là các giá trị hoặc danh mục được ghi trên trục (ví dụ: các năm "2021", "2022" trên trục X)
- `Tick_Grouping` là văn bản nhóm các nhãn trục (ví dụ: nếu `Tick_Label` là tháng, `Tick_Grouping` có thể là năm "Quý 1, 2023")
- `Legend_Title` là tiêu đề của phần chú giải (ví dụ: "Loại sản phẩm")
- `Legend_Label` là chú thích mô tả ý nghĩa của các màu sắc/ký hiệu trong biểu đồ
- `Value_Label` là các giá trị số được ghi trực tiếp trên hoặc ngay cạnh thanh/điểm dữ liệu, dùng để làm nổi bật giá trị đó.
- `Mark_Label` là nhãn danh mục (thường là chữ) gắn với một điểm hoặc thanh dữ liệu cụ thể
- `Other` là văn bản không liên quan đến dữ liệu cốt lõi (ví dụ: Logo, nguồn trích dẫn, tên tác giả)

Đầu vào (Input) của LayoutLMv3:

- Văn bản OCR: Chuỗi token đã được nhận diện.
- Bố cục (Layout): Tọa độ bounding box chuẩn hóa cho mỗi token.
- Hình ảnh (Image): Các Patch Embedding (biểu diễn hình ảnh) tương ứng với từng vùng văn bản.

Đầu ra (Output) của LayoutLMv3:

- Một vector vai trò ngữ nghĩa (Semantic Role) cho mỗi token, được sử dụng làm cơ sở để tái tạo cấu trúc dữ liệu ở Giai đoạn 3.

3.2.3 Tinh chỉnh model LayoutLMv3)

LayoutLMv3 được tinh chỉnh (Fine-tuned) để thực hiện tác vụ Sequence Labeling (Gán nhãn cho chuỗi văn bản). Mô hình này nhận đầu vào là các token văn bản cùng với tọa độ bounding box, và có nhiệm vụ phân loại mỗi token vào một trong chín (9) vai trò ngữ nghĩa

(Semantic Role) đã định nghĩa trong tập nhãn của nhóm.

Do đặc thù của dữ liệu biểu đồ (Mục 4.1.5), lớp `Tick_Label` (các nhãn trên trục) chiếm số lượng mẫu lớn nhất, trong khi các lớp quan trọng khác như `Chart_Title` hoặc `Value_Label` lại có số lượng ít hơn nhiều. Hiện tượng **`mất cân bằng lớp`** này có thể khiến mô hình LayoutLMv3 thiên vị dự đoán lớp đa số, dẫn đến hiệu suất kém đối với các lớp thiểu số.

Để giải quyết vấn đề mất cân bằng lớp và nâng cao khả năng phân loại các lớp thiểu số, chúng tôi sử dụng **`Weighted Cross-Entropy Loss`** làm hàm mất mát trong quá trình tinh chỉnh LayoutLMv3.

(i) **Công thức Loss Function:**

$$\mathcal{L} = -\sum_{i=1}^{N} w_i \cdot y_i \log(\hat{y}_i)$$

Trong đó, w_i là trọng số được gán cho lớp thứ i, được tính toán dựa trên tần suất nghịch đảo của lớp (Inverse Frequency Weighting) trên tập huấn luyện:

$$w_i = \frac{\text{Tổng số mẫu}}{\text{Số mẫu của lớp } i \times \text{Tổng số lớp}}$$

(ii) Cơ chế Trọng số:

- Lớp `Tick_Label` được gán trọng số $w_{\text{tick}} < 1$ (giảm nhẹ ảnh hưởng của lớp đa số).
- Các lớp thiểu số quan trọng (như `VALUE_LABEL`, `CHART_TITLE`) được gán trọng số $w_{\text{minority}} \geq 1$.

Mục đích là để mô hình tập trung vào việc học đặc trưng của các lớp có số lượng mẫu thấp hơn, từ đó cải thiện **Recall** và **F1-Score** trên các lớp quan trọng này.

Đầu vào/Đầu ra và Tích hợp

(i) Đầu vào (Input): Các Embedding Đa phương thức (Text, Layout, Image).

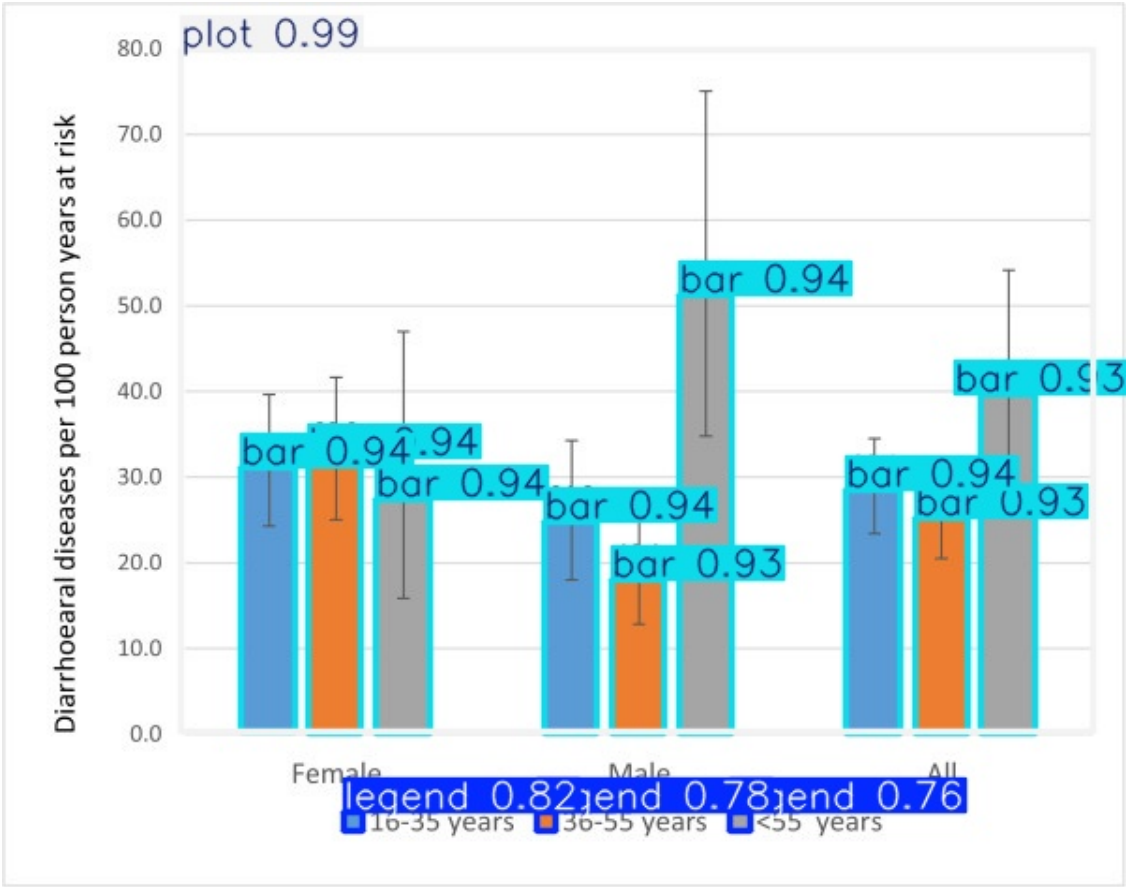
(ii) Quá trình Huấn luyện: LayoutLMv3 được tinh chỉnh (Fine-tuned) với **Weighted Cross-Entropy Loss**.

(iii) Đầu ra (Output): Vai trò ngữ nghĩa (Semantic Role) đã được phân loại cho từng token, làm cơ sở tin cậy cho **Giai đoạn 3 (Thuật toán ghép nối).**

3.3 Vai trò của YOLOv8s

Hệ thống được thiết kế theo hướng mô-đun, trong đó kết quả từ các bước xử lý văn bản (Text Detection/Recognition và Text Role Classification) được kết hợp với mô-đun phát hiện đối tượng (`YOLOv8s`) để phục vụ hai nhiệm vụ chính: *(i) phân tích trục (Axis Analysis)*, *(ii) phân tích chú giải (Legend Analysis)* và *(iii) Phát hiện các cột (Element Detection).*

Cụ thể, `YOLOv8s` được huấn luyện để phát hiện ba nhóm phần tử chính trên biểu đồ cột:



Hình 3.1: Vai trò của YOLOv8s

- **Plot region** (vùng biểu đồ): dùng để suy ra vị trí **trục X/Y** theo giả định bố cục của biểu đồ cột 2D.
- **Legend patch** (ô màu/ký hiệu chú giải): đại diện trực quan cho một chuỗi dữ liệu (data series).
- **Bar** (cột dữ liệu): phần tử đồ hoạ cần được suy ra giá trị số ở bước trích xuất dữ liệu.

Đầu vào của pipeline tại giai đoạn này gồm:

- Ảnh biểu đồ (RGB).
- JSON trung gian từ khối xử lý văn bản, bao gồm danh sách text blocks và nhãn vai trò (tick label, axis title, legend label, . . .).
- Kết quả `YOLOv8s`: tập bbox cho `plot`, `legend` và `bar`.

Đầu ra của hai task trong mục này gồm:

- Trục chính `xaxis`, `yaxis` và danh sách `tick labels` tương ứng.
- Danh sách cặp `legend entry` dưới dạng `(legend label, legend patch)`.

3.4 Phương pháp để phân tích Chart Axis

3.4.1 Mục tiêu

Trục của biểu đồ xác định không gian dữ liệu và cung cấp cơ sở để ánh xạ từ toạ độ pixel sang giá trị số. Mục tiêu của bài toán là:

1. Xác định hai trục chính (ngang và dọc) của vùng biểu đồ.
2. Liên kết các `tick labels` với trục tương ứng (X hoặc Y), không phụ thuộc vào việc có vạch tick trực quan hay không.
3. Phân tách tiêu đề trục (axis title) theo trục X/Y.

3.4.2 Bước 1: Suy ra trục từ vùng `plot`

Trong biểu đồ cột tiêu chuẩn (2D, không 3D), vùng `plot` thường bao phủ phần đồ thị chính và hai trục nằm ở biên của vùng này. Từ bbox lớn nhất của lớp `plot`:

$$\texttt{plot} = (x_1, y_1, x_2, y_2),$$

ta suy ra:

- Trục X là cạnh dưới: $\texttt{xaxis} = (x_1, y_2, x_2, y_2)$.
- Trục Y là cạnh trái: $\texttt{yaxis} = (x_1, y_1, x_1, y_2)$.

Cách làm này giảm phụ thuộc vào việc ảnh có vạch trục đậm/nhạt hay có bị nhiễu bởi phần tử khác.

3.4.3 Bước 2: Tách tick labels theo vùng không gian (cross product)

Sau khi có vai trò văn bản từ bước Text Role Classification, tập các text region có role `tick_label` được chia thành hai nhóm: tick của trục X và tick của trục Y.

Với một tick label có bbox (t_x, t_y, w, h), ta dùng tâm bbox:

$$c_x = t_x + \frac{w}{2}, \quad c_y = t_y + \frac{h}{2}.$$

Ký hiệu:

$$\texttt{xaxis} = (x_1, y_1, x_2, y_2), \quad \texttt{yaxis} = (x'_1, y'_1, x'_2, y'_2).$$

Dấu của tích có hướng (2D cross product) giúp xác định điểm nằm về phía nào so với đường thẳng:

$$s_X = \operatorname{sign}\big((x_2 - x_1)(c_y - y_1) - (y_2 - y_1)(c_x - x_1)\big),$$

$$s_Y = \operatorname{sign}\big((x'_2 - x'_1)(c_y - y'_1) - (y'_2 - y'_1)(c_x - x'_1)\big).$$

Quy tắc phân vùng:

- Nếu $s_Y = 1$: xem là **Y-tick** (thường nằm bên trái trục Y).
- Nếu $s_X = 1$ và $s_Y = -1$: xem là **X-tick** (thường nằm bên dưới trục X và bên phải trục Y).

3.4.4 Bước 3: Phân tách tiêu đề trục theo khoảng cách tới trục

Các text region có role `axis_title` được gán về trục gần nhất bằng khoảng cách từ tâm bbox đến đường thẳng của trục.

Khoảng cách từ điểm (p_x, p_y) tới đường thẳng qua (x_1, y_1) và (x_2, y_2):

$$d = \frac{|(y_2 - y_1)p_x - (x_2 - x_1)p_y + x_2y_1 - y_2x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}.$$

Nếu $d(\text{title}, \texttt{yaxis}) < d(\text{title}, \texttt{xaxis})$ thì gán về **Y-axis title**, ngược lại gán về **X-axis title**.

3.5 Phương pháp để phân tích Chart Legend

3.5.1 Mục tiêu

Legend là tập các *legend entries*, mỗi entry là một cặp:

$$(\text{legend label}, \text{legend symbol/patch}),$$

trong đó legend label mô tả tên chuỗi dữ liệu và legend patch biểu diễn kiểu hiển thị (màu/texture) tương ứng. Mục tiêu là phát hiện và ghép đúng các cặp này trong điều kiện có thể có nhiễu (false positives/false negatives).

3.5.2 Đầu vào

- **Legend labels**: danh sách bbox văn bản có role `legend_label` từ khối phân loại vai trò.
- **Legend patches**: danh sách bbox của lớp `legend` từ YOLOv8s (ô màu/ký hiệu).

3.5.3 Xử lý trường hợp không có legend

Nếu không có `legend_label`, hệ thống chuyển sang chế độ *single-series* và gán mặc định tên chuỗi là `series_0` để đảm bảo pipeline trích xuất dữ liệu vẫn chạy được.

3.5.4 Ghép legend label và legend patch bằng bài toán gán tối ưu (Hungarian)

Với:

$$L = \{l_i\}_{i=1}^{n_L}, \quad P = \{p_j\}_{j=1}^{n_P}$$

là tập legend labels và legend patches, ta xây dựng ma trận chi phí $C \in \mathbb{R}^{n_L \times n_P}$.

Mỗi label l_i có bbox (t_x, t_y, t_w, t_h) và patch p_j có bbox (x, y, w, h), lấy tâm:

$$(c_x^L, c_y^L), \quad (c_x^P, c_y^P).$$

Ràng buộc quan trọng:

- **Cùng hàng**: legend patch và legend text phải có y gần bằng nhau.
- **Ưu tiên patch nằm bên trái label**: $c_x^P < c_x^L$.

Sau đó, áp dụng thuật toán Hungarian để tối thiểu tổng chi phí ghép. Kết quả là ánh xạ:

$$\pi(i) = j \quad \text{hoặc} \quad \pi(i) = \varnothing$$

cho biết label l_i được ghép với patch p_j nào (hoặc không ghép được).

3.5.5 Đầu ra

Sau khi ghép, ta thu được danh sách *legend entries*:

$$E = \{(\text{text}_i,\ \text{patch\_bbox}_{\pi(i)})\},$$

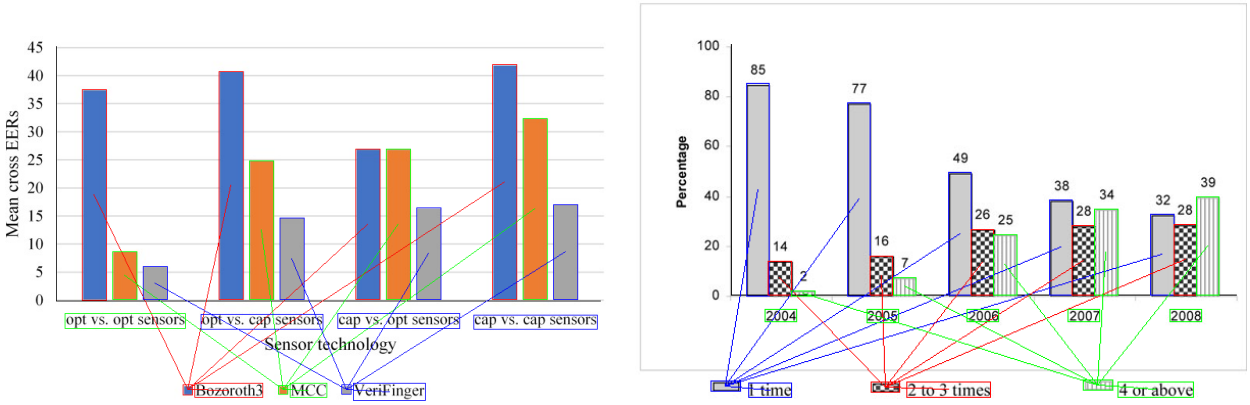
được dùng ở các bước sau để (i) định danh chuỗi dữ liệu và (ii) hỗ trợ gán bar vào đúng series trong bài toán trích xuất dữ liệu.

3.6 Data Extraction

Giai đoạn cuối cùng của pipeline nhằm mục đích tổng hợp các kết quả phân tích sơ cấp để tái cấu trúc bảng dữ liệu hoàn chỉnh. Quá trình này được chia thành hai tác vụ con: phát hiện phần tử đồ họa (Task 5a) và trích xuất dữ liệu thô (Task 5b).

3.6.1 Plot Element Detection (Task 5a)

Để định vị chính xác các thanh dữ liệu (bars) trong biểu đồ, nhóm sử dụng mô hình phát hiện đối tượng **YOLOv8** (phiên bản s) đã được tinh chỉnh (fine-tuned) trên tập dữ liệu huấn luyện. Mô hình nhận đầu vào là toàn bộ ảnh biểu đồ và trả về tập hợp các bounding box $B = \{b_1, b_2, \ldots, b_n\}$ cho lớp đối tượng `bar`



Bar-to-Category Mapping Bar-to-Category Mapping

Hình 3.2: Mapping

So với các phương pháp xử lý ảnh truyền thống (như Connected Components hay Color Thresholding), phương pháp học sâu giúp hệ thống bền vững hơn trước các yếu tố nhiễu thường gặp trong thực tế như: đường lưới nền (grid lines), sự chồng lấn của các thành phần đồ họa, và sự đa dạng về mẫu kết cấu (texture patterns) của các cột.

3.6.2 Raw Data Extraction (Task 5b)

Sau khi đã thu được tập hợp các bounding box của cột, quy trình trích xuất dữ liệu thực hiện ba bước liên kết và tính toán như sau:

1. Bar-to-Series Mapping (Liên kết chuỗi dữ liệu) Đối với biểu đồ nhiều chuỗi (multi-series), việc xác định mỗi cột thuộc về chuỗi nào (được định nghĩa trong Legend) là thách thức lớn nhất. Nhóm sử dụng mô hình Deep learning để embedding ảnh của cột và của ô chú giải, sau đó so sánh tìm legend có score tốt nhất cho mỗi cột bằng độ đo cosine:

- **Preprocessing:** Mỗi vùng ảnh cột (bar patch) được cắt (crop) và tiền xử lý bằng kỹ thuật *Vertical Shrinking*. Cụ thể, bounding box được co hẹp biên theo chiều ngang nhưng giữ nguyên chiều dọc để loại bỏ viền và nhiễu từ các cột lân cận trong khi vẫn bảo toàn thông tin chiều cao.
- **Feature Matching:** Vector đặc trưng của cột (v_{bar}) và của mẫu chú giải (v_{legend}) được trích xuất thông qua mạng backbone ResNet50. Nhãn chuỗi của cột b_i được xác định

bởi mẫu chú giải có độ tương đồng cosine cao nhất:

$$\text{Series}(b_i) = \arg\max_{L_j \in \mathcal{L}} \left(\frac{v_{b_i} \cdot v_{L_j}}{\|v_{b_i}\| \|v_{L_j}\|} \right) \tag{3.1}$$

Trong trường hợp biểu đồ đơn chuỗi (single-series) hoặc không phát hiện chú giải, hệ thống mặc định gán tất cả các cột vào một chuỗi duy nhất.

2. Bar-to-Category Mapping (Liên kết nhãn trục hoành) Để xác định nhãn phân loại (X-axis label) cho từng cột, nhóm áp dụng thuật toán *Nearest Neighbor* dựa trên khoảng cách hình học. Với mỗi cột b_i có tọa độ tâm theo trục hoành là $C_{b_i}.x$, hệ thống duyệt qua tập các nhãn trục hoành T_X (đã được xác định từ module Axis Analysis) để tìm nhãn t_k có khoảng cách không gian nhỏ nhất:

$$\text{Category}(b_i) = \arg\min_{t_k \in T_X} \|C_{b_i}.x - C_{t_k}.x\| \tag{3.2}$$

Phương pháp này giả định rằng cột và nhãn tương ứng luôn được căn chỉnh thẳng hàng theo trục dọc (vertical alignment).

3. Value Estimation (Ước lượng giá trị) Giá trị thực $V(b_i)$ của mỗi cột được tính toán dựa trên chiều cao pixel h_{pixel} và tham số tỉ lệ (scaling factor) α (tỉ lệ pixel-to-value) đã ước lượng được từ module Axis Analysis.

$$V(b_i) = V_{base} + h_{pixel}(b_i) \times \alpha \tag{3.3}$$

Trong đó V_{base} là giá trị tại đường cơ sở (thường là trục hoành, tương ứng giá trị 0). Kết quả cuối cùng được làm tròn (rounding) dựa trên độ chính xác số học (số chữ số thập phân) được suy luận từ định dạng của các tick labels trên trục tung, nhằm đảm bảo tính nhất quán của dữ liệu đầu ra.

4 Kết quả và Đánh giá

4.1 Dữ liệu để huấn luyện và đánh giá

Nhóm nghiên cứu sử dụng bộ dữ liệu ICPR 2022 Chart Information Extraction (Chart-Info), được cung cấp thông qua thử thách quốc tế về Trích xuất Thông tin từ Biểu đồ tại

Hội nghị Quốc tế về Nhận dạng Mẫu (ICPR). Bộ dữ liệu này được chọn vì tính đầy đủ, đa dạng và tính chuẩn mực cao, tạo điều kiện thuận lợi cho việc so sánh kết quả với các phương pháp tiếp cận tiên tiến (State-of-the-Art - SOTA)

4.1.1 Nguồn gốc và Phương pháp thu thập

Bộ dữ liệu ChartInfo được tạo ra từ việc hợp nhất và mở rộng các bộ dữ liệu trước đó, với quá trình thu thập và xây dựng dataset mới cho phiên bản 2024 (mà nhóm bạn sử dụng) tuân theo quy trình nghiêm ngặt:

- Nguồn dữ liệu thực tế: Các hình ảnh biểu đồ được trích xuất từ PubMed Central (PMC) trong khoảng thời gian từ tháng 12/2017 đến tháng 10/2021.
- Tiêu chí lọc: Chỉ các bài báo có sẵn dưới giấy phép CC BY hoặc CC-0 và chứa các từ khóa "chart" hoặc "plot" trong văn bản chính mới được xem xét.
- Quy trình lọc tự động: Một thuật toán phân loại hình ảnh nhị phân đã được sử dụng để xác định các ứng viên biểu đồ từ tất cả các hình vẽ trong 20K bài báo được chọn ngẫu nhiên. Sau đó, các ứng viên này được phân loại thủ công theo loại biểu đồ.
- Định dạng: Toàn bộ hình ảnh trong bộ dữ liệu là định dạng JPEG, do được lấy trực tiếp từ PMC. Điều này đặt ra hạn chế không bao gồm định dạng vector, nhưng chúng tôi lưu ý rằng hình ảnh vector có thể dễ dàng được chuyển đổi sang raster để mô hình có thể nhận diện.

4.1.2 Thách thức về Chất lượng và Độ đa dạng

Bộ dữ liệu ChartInfo đặt ra nhiều thách thức quan trọng đối với các mô hình học sâu, đặc biệt là các mô hình thị giác (Vision Models) như `YOLO` và `LayoutLMv3`:

- Kích thước và Độ phân giải đa dạng: Hình ảnh có sự biến thiên lớn về kích thước.
  - Training Set: Kích thước từ 76×75 đến 10.800×6.000 pixels.
  - Testing Set: Kích thước từ 120×66 đến 6.299×6.299 pixels.
  - Ý nghĩa: Sự biến thiên lớn này gây khó khăn cho các mô hình yêu cầu độ phân giải đầu vào cố định. Việc thay đổi kích thước ảnh có thể làm mất đi các chi tiết quan trọng (ví dụ: phân biệt giữa Line Chart và Scatter-Line Chart) hoặc làm giảm độ rõ nét của văn bản, ảnh hưởng đến hiệu suất `OCR`

- Độ rõ nét (Clarity): Các nhà sáng tạo biểu đồ đã sử dụng nhiều công cụ khác nhau, dẫn đến sự đa dạng về chất lượng hình ảnh.
- Tỷ lệ tương đối: Sự khác biệt lớn trong tỷ lệ tương đối giữa kích thước văn bản và kích thước của các đối tượng đồ họa khác làm tăng độ khó của nhiệm vụ phát hiện (Detection) trong giai đoạn YOLO.

4.1.3 Phân tích chi tiết phân bố biểu đồ

Để cung cấp cái nhìn sâu sắc về tính đa dạng và quy mô của bộ dữ liệu được sử dụng, bảng dưới đây trình bày phân bố chi tiết số lượng biểu đồ theo từng loại và theo từng phiên bản ICPR (ICPR 2020, 2022, 2024), trong đó, phiên bản ICPR 2022 là phiên bản được nhóm sử dụng làm cơ sở huấn luyện và đánh giá.

Chart Type	ICPR 2020		ICPR 2022		ICPR 2024 (Sử dụng)	
	Train	Test	Train	Test	Train	Test
Area	120	52	172	136	308	229
Line	7.401	3.155	10.556	3.400	13.955	5.142
Manhattan	123	53	176	80	256	68
Scatter	875	475	1.350	1.247	2.597	1.311
Scatter-Line	1.260	558	1.818	1.628	3.446	1.684
Pie	170	72	242	191	433	213
Vertical Box	316	447	763	775	1.538	802
Horizontal Bar	429	358	787	634	1.421	636
Vertical Bar	3.818	1.636	5.454	3.745	9.199	3.692
Horizontal Interval	109	47	156	430	586	326
Vertical Interval	342	147	489	182	671	202
Map	373	160	533	373	906	363
Heatmap	138	59	197	180	377	177
Surface	110	45	155	128	283	127
Venn	52	23	75	131	206	121
Total	**15.636**	**7.287**	**22.923**	**13.260**	**36.182**	**15.093**

Bảng 4.1: Thống kê số lượng biểu đồ theo loại và bộ dữ liệu ICPR

4.1.4 Tổng quan về Bộ dữ liệu ChartInfo

Bộ dữ liệu ICPR 2022 Chart Information Extraction (ChartInfo), được xây dựng và mở rộng qua các phiên bản cuộc thi quốc tế, là cơ sở dữ liệu chính cho dự án này. Bộ dữ liệu được chọn vì tính đầy đủ, đa dạng và việc gán nhãn chi tiết cho phép huấn luyện và đánh giá từng module độc lập trong kiến trúc hệ thống của nhóm.

Bộ dữ liệu ChartInfo là bộ dữ liệu đa tác vụ, cung cấp các nhãn Ground Truth (GT) chi tiết cho nhiều nhiệm vụ khác nhau trong quy trình trích xuất thông tin biểu đồ. Sự phong phú này là lý do chính khiến bộ dữ liệu này phù hợp với kiến trúc nhiều giai đoạn

(Multi-stage Pipeline) của nhóm:

Tác vụ được Hỗ trợ	**Ứng dụng trong Pipeline của Nhóm**	**Mục đích của Nhãn GT**
Phát hiện & Nhận dạng Từ (Text Detection & Recognition)	Giai đoạn 1 (YOLO + PaddleOCR)	Cung cấp bounding box chính xác và chuỗi văn bản gốc.
Phân loại Vai trò (Role Classification)	Giai đoạn 2 (LayoutLMv3)	Cung cấp vai trò ngữ nghĩa của mỗi đoạn văn bản (`TICK_LABEL`, `VALUE_LABEL`, v.v.).
Phân tích Trục (Axis Analysis)	Giai đoạn 3 (Thuật toán Ghép nối)	Cung cấp mối quan hệ giữa các nhãn trục (Tick Labels) với trục X và Y.
Phân tích Chú giải (Legend Analysis)	Giai đoạn 4 (Thuật toán Ghép nối)	Cung cấp mối quan hệ giữa các nhãn chú giải (Legend Labels) và các màu sắc/hình dạng dữ liệu tương ứng.
Phát hiện & Phân tích Dữ liệu (Data Analysis)	Giai đoạn 5 (Thuật toán Ghép nối)	Cung cấp vị trí và giá trị thực của các phần tử đồ họa (Bar, Line, Point).

Bảng 4.2: Các tác vụ được hỗ trợ và vai trò của nhãn Ground Truth trong pipeline

Dữ liệu được gán nhãn dưới định dạng JSON, giúp dễ dàng xử lý và chuyển đổi thành định dạng đầu vào cho các mô hình học sâu. Cụ thể, dữ liệu cung cấp:

- Văn bản: Gồm nội dung (Text String), Tọa độ (Bounding Box) và Vai trò ngữ nghĩa (Semantic Role). Dữ liệu này được sử dụng trực tiếp để fine-tune LayoutLMv3.
- Hình ảnh: Tọa độ của các phần tử đồ họa (cột, đường) được dùng để huấn luyện mô hình YOLO trong Giai đoạn 3 (phát hiện đối tượng đồ họa).
- Liên kết (Association): Các liên kết giữa nhãn trục, nhãn chú giải và dữ liệu số đã được gán nhãn thủ công, cung cấp Ground Truth để đánh giá độ chính xác của **Thuật toán Ghép nối (Data Association Algorithm)** của nhóm.

Sự chi tiết trong việc gán nhãn GT này cho phép nhóm đánh giá và tối ưu hóa từng module một cách độc lập trước khi đánh giá hệ thống end-to-end.

4.2 Đánh giá Task 2: Phát hiện và Nhận diện văn bản

Giai đoạn này đóng vai trò trích xuất thông tin thô từ hình ảnh, bao gồm hai bước chính: Phát hiện vị trí văn bản (Text Detection) và Nhận diện nội dung văn bản (Text Recognition).

4.2.1 Cấu hình thực nghiệm

- **Text Detection:** Sử dụng mô hình **YOLOv8s-obb** (Oriented Bounding Box) được tinh chỉnh (fine-tune) trên tập dữ liệu ICPR-2022. Việc sử dụng OBB giúp mô hình xử lý hiệu quả các văn bản xoay nghiêng hoặc nằm sát nhau thường gặp trong biểu đồ.
- **Text Recognition:** Sử dụng thư viện **PaddleOCR** để chuyển đổi vùng ảnh chứa văn bản thành chuỗi ký tự.

4.2.2 Kết quả Phát hiện văn bản (Text Detection)

Chúng tôi đánh giá hiệu năng phát hiện văn bản dựa trên các độ đo Precision, Recall, F1-score và Mean IoU (Intersection over Union). Kết quả tổng hợp trên tập kiểm thử được trình bày trong Bảng 4.3.

Bảng 4.3: Kết quả đánh giá Text Detection (Micro-average)

Phương pháp	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Mean IoU (%)**
YOLOv8s-obb	81.19	82.73	81.95	70.54

Thống kê chi tiết: TP: 29,964; FP: 6,944; FN: 6,254.

Nhận xét: Mô hình đạt F1-Score xấp xỉ 82% với sự cân bằng tốt giữa Precision và Recall. Chỉ số Mean IoU đạt trên 70% cho thấy khung bao dự đoán bám khá sát với thực tế, đảm bảo việc cắt ảnh cho bước nhận diện sau này có chất lượng tốt.

4.2.3 Kết quả Nhận diện văn bản (Text Recognition)

Hiệu năng nhận diện được đánh giá trên các cặp ảnh/nhãn đã khớp (matched pairs) từ bước detection. Các chỉ số quan trọng bao gồm Accuracy (tỷ lệ từ đúng), Character Accuracy (tỷ lệ ký tự đúng) và Mean Normalized Edit (sai số chỉnh sửa chuẩn hóa).

Bảng 4.4: Kết quả đánh giá Text Recognition

Tiêu chí đánh giá	**Kết quả**
Word Accuracy (Raw)	83.93%
Word Accuracy (Normalized)	84.90%
Character Accuracy	92.11%
Mean Normalized Edit Distance	0.0497
Tổng số cặp đánh giá	**29,964**

Nhận xét:

- **Độ chính xác từ (Word Accuracy):** Đạt khoảng 84-85%, nghĩa là phần lớn các nhãn văn bản được đọc đúng hoàn toàn.
- **Độ chính xác ký tự (Character Accuracy):** Đạt tới 92.11%, cho thấy ngay cả khi từ bị sai, sai số thường chỉ nằm ở 1-2 ký tự nhỏ (như dấu chấm, dấu phẩy), điều này được thể hiện qua chỉ số *Mean Normalized Edit* rất thấp (0.0497 $\approx$ 5% sai lệch trên chiều dài chuỗi).

4.3 Đánh giá Task 3: Phân loại vai trò văn bản (Text Role Classification)

Sau khi văn bản được nhận diện, hệ thống thực hiện gán nhãn ngữ nghĩa (Semantic Labeling) để xác định vai trò của từng thành phần (ví dụ: Tiêu đề biểu đồ, Tiêu đề trục, Chú thích...). Đây là bước then chốt để hiểu cấu trúc thông tin của biểu đồ.

4.3.1 Phương pháp và Mô hình

Chúng em sử dụng mô hình **LayoutLMv3** (Multimodal Transformer). Mô hình này kết hợp đồng thời ba nguồn thông tin đầu vào:

- **Text:** Nội dung văn bản (từ kết quả OCR).
- **Layout:** Tọa độ không gian (Bounding Box).
- **Image:** Đặc trưng thị giác của vùng ảnh chứa văn bản.

4.3.2 Kết quả Thực nghiệm

Kết quả đánh giá trên tập kiểm thử được trình bày trong Bảng 4.5.

Bảng 4.5: Kết quả đánh giá Task 3 sử dụng LayoutLMv3 (Micro-average)

Tiêu chí (Metric)	**Giá trị**
Accuracy	70.28%
Precision	**98.90%**
Recall	70.28%
F1-Score	**82.17%**
Số lượng GT items đánh giá	1748

4.3.3 Phân tích và Đánh giá (Analysis)

Dựa trên số liệu từ Bảng 4.5, chúng em có các nhận xét sau:

1. **Độ chính xác (Precision) ấn tượng (98.90%):** Chỉ số Precision gần như tuyệt đối cho thấy khi mô hình đưa ra dự đoán, nó hầu như luôn chính xác. Sự kết hợp giữa nội dung văn bản và vị trí (Layout) giúp LayoutLMv3 phân biệt rạch ròi giữa các thành phần dễ nhầm lẫn (như *Axis Title* nằm giữa cạnh và *Tick Label* nằm dọc trục), chứng minh tính hiệu quả của kiến trúc đa phương thức.

2. **Vấn đề về dữ liệu và chỉ số Recall (70.28%):** Trong quá trình kiểm thử, hệ thống ghi nhận cảnh báo về các tệp dự đoán bị thiếu ("Missing prediction files"). Qua phân tích, chúng em xác định nguyên nhân chính **không phải do lỗi mô hình**, mà xuất phát từ sự **không đồng nhất của bộ dữ liệu ICPR-2022**.

   Cụ thể, một lượng lớn ảnh trong tập dữ liệu này không được gán nhãn đầy đủ cho Task 3 (Text Role Classification). Do đó, khi đánh giá trên toàn bộ tập test (dựa trên danh sách ảnh của Task 2), script đánh giá ghi nhận sự thiếu hụt dữ liệu đối sánh (Ground Truth) hoặc thiếu file kết quả tương ứng cho các ảnh này, dẫn đến chỉ số Recall bị kéo giảm xuống 70.28%.

   Kết luận: Nếu chỉ xét trên tập dữ liệu có nhãn đầy đủ, hiệu năng thực tế của mô hình (thể hiện qua F1-Score và Precision) là rất cao, đủ độ tin cậy để phục vụ cho các tác vụ trích xuất dữ liệu phía sau.

4.4 Kết quả và đánh giá Axis Analysis, Legend Analysis, Data Extraction

4.4.1 Phạm vi đánh giá và Dữ liệu đầu ra

Trong giai đoạn hiện tại, nhóm nghiên cứu tập trung hoàn thiện và đánh giá mô hình phát hiện đối tượng (Object Detection) sử dụng kiến trúc YOLOv8s. Mô hình được huấn luyện để nhận diện ba lớp đối tượng cốt lõi trong biểu đồ: `plot` (vùng vẽ), `legend_patch` (ô chú giải), và `bar` (cột dữ liệu).

Dữ liệu đầu ra hiện tại bao gồm:

(i) Tập hợp các khung bao (bounding boxes) dự đoán với toạ độ và độ tin cậy (confidence score).

(ii) Hình ảnh trực quan hoá (visualization) kết quả phát hiện trên tập dữ liệu kiểm thử.

(iii) Các biểu đồ theo dõi quá trình huấn luyện (learning curves, mAP, PR curves).

Do các mô-đun xử lý phía sau (như Axis Analysis, Legend Analysis, Data Extraction) phụ thuộc chặt chẽ vào độ chính xác của bước phát hiện đối tượng, phần này sẽ tập trung vào (1) báo cáo hiệu năng của YOLOv8s và (2) phân tích định tính các lỗi phổ biến làm cơ sở cho việc tối ưu hoá hệ thống trong giai đoạn tiếp theo.

4.4.2 Hiệu năng mô hình YOLOv8s (Task 5a: Plot Element Detection)

4.4.3 Kết quả định lượng trên tập huấn luyện

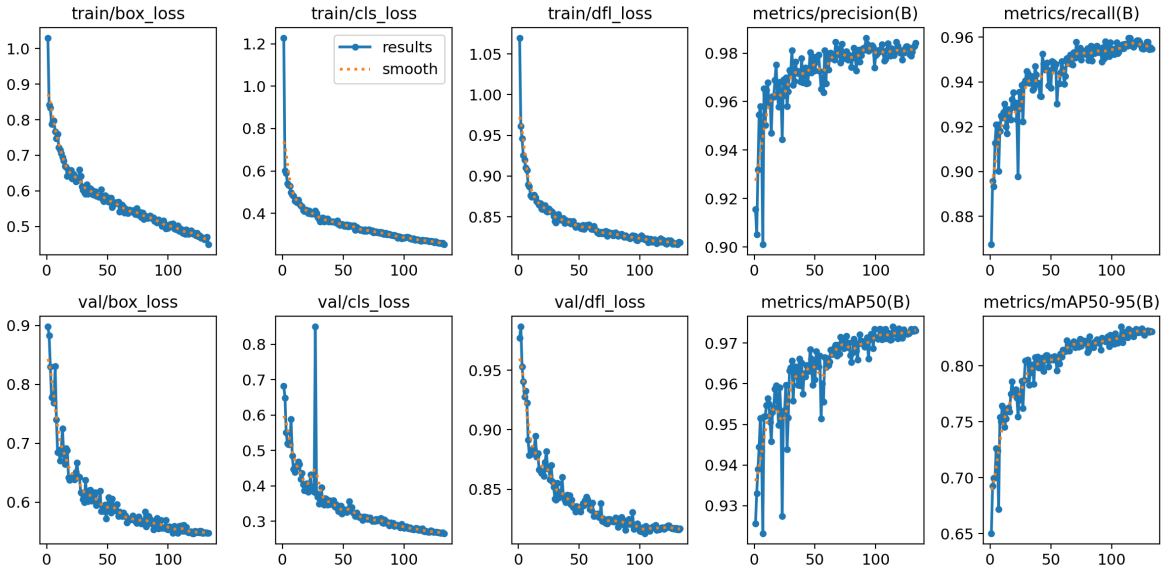
Tiêu chí (Metric)	Giá trị
Precision (P)	0.977
Recall (R)	0.955
mAP@0.5	0.974
mAP@0.5:0.95	0.835

Hình 4.1: Kết quả huấn luyện mô hình Yolov8s.

Hiệu năng của mô hình được đánh giá thông qua các chỉ số tiêu chuẩn trong quá trình huấn luyện:

- **Hàm mất mát (Loss):** Sự hội tụ của Box Loss, Class Loss và DFL Loss qua các epoch.
- **Độ chính xác trung bình (mAP):** Báo cáo mAP@0.5 và mAP@0.5:0.95 cho từng lớp `plot`, `legend_patch`, và `bar`.
- **Precision-Recall (PR) Curve:** Đánh giá sự cân bằng giữa độ chính xác và độ phủ của mô hình.

Các chỉ số này đóng vai trò tiên quyết, xác nhận khả năng trích xuất đặc trưng của mô hình trước khi đi vào các bước xử lý logic phức tạp hơn.



Hình 4.2: Biểu đồ huấn luyện YOLOv8s thể hiện sự hội tụ của hàm loss và sự cải thiện chỉ số mAP trên ba lớp đối tượng.

Nhận xét biểu đồ huấn luyện YOLOv8s. Quan sát Hình 4.2, có thể rút ra các nhận xét chính như sau:

- **Hội tụ của hàm mất mát (loss):**

  - Các đường `train/box_loss`, `train/cls_loss`, `train/dfl_loss` giảm đều theo epoch và dần ổn định về cuối, cho thấy quá trình tối ưu diễn ra hiệu quả và mô hình có xu hướng hội tụ.
  - Các đường `val/box_loss`, `val/cls_loss`, `val/dfl_loss` cũng giảm theo xu hướng tương tự và bám sát train, không xuất hiện khoảng cách tăng dần rõ rệt giữa train và val. Điều này gợi ý mô hình học ổn định và chưa có dấu hiệu overfitting mạnh.

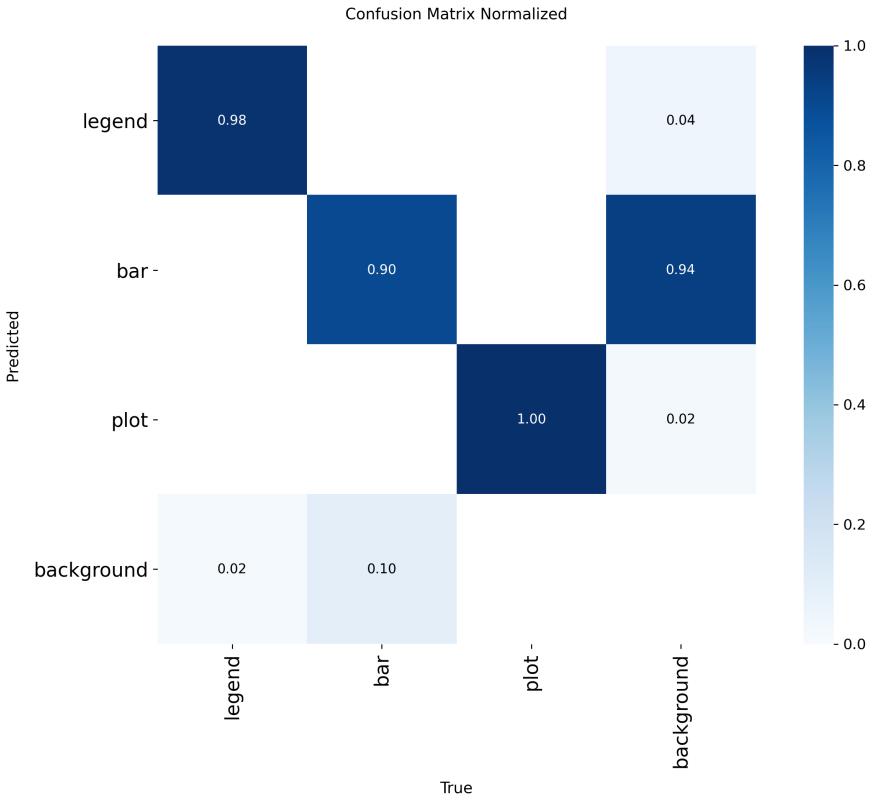
- **Chất lượng dự đoán (precision/recall):**
  - `precision` tăng nhanh và duy trì ở mức cao (xấp xỉ ∼0.98), cho thấy tỷ lệ dự đoán đúng trong các bbox được mô hình phát hiện là tốt.
  - `recall` ổn định quanh ∼0.95, phản ánh mô hình phát hiện tương đối đầy đủ các đối tượng `plot`/`legend_patch`/`bar` (ít bỏ sót).

- **mAP và mức độ "khít" của bounding box:**
  - `mAP@0.5` đạt mức cao (khoảng ∼0.97), cho thấy mô hình hoạt động tốt ở ngưỡng IoU vừa phải.
  - `mAP@0.5:0.95` thấp hơn đáng kể (khoảng ∼0.83), hàm ý vẫn còn dư địa cải thiện độ *khít* của bbox, đặc biệt với các đối tượng nhỏ/hẹp như `bar` mảnh hoặc `legend_patch`.

Nhận xét. Hình 4.3 cho thấy mô hình phân biệt tốt giữa ba lớp đối tượng chính.

- **Độ đúng theo lớp (đường chéo):**
  - `plot` đạt gần như tuyệt đối (khoảng 1.00), cho thấy mô hình nhận diện vùng vẽ rất ổn định.
  - `legend` đạt khoảng 0.98, phản ánh legend patch thường có hình dạng/biên rõ ràng và ít nhầm lẫn.
  - `bar` thấp hơn (khoảng 0.90), cho thấy đây là lớp khó do cột có thể mảnh/hẹp, sát nhau, hoặc bị ảnh hưởng bởi đường kẻ/lưới và hoạ tiết nền.

- **Bỏ sót (false negative) nhìn từ hàng `background`:**



Hình 4.3: Ma trận nhầm lẫn chuẩn hoá của YOLOv8s cho ba lớp `plot`, `legend`, `bar`. Trục hoành: nhãn thật (True), trục tung: nhãn dự đoán (Predicted).

  - `bar` bị bỏ sót vào khoảng 0.10 (cao nhất), trong khi `legend` khoảng 0.02 và `plot` gần như không đáng kể.
  - Điều này phù hợp với lỗi thực tế: cột nhỏ hoặc cột có màu gần nền dễ không được detector bắt đúng.

- **Dự đoán nhầm nền thành đối tượng (false positive) nhìn từ cột `background`:**
  - Phần lớn false positive đến từ lớp `bar` (khoảng 0.94), tức mô hình có xu hướng nhầm các cấu trúc dạng thanh (gridlines, trục, hoạ tiết) thành cột dữ liệu.
  - Các lớp `plot` và `legend` ít gây nhầm nền hơn.

Tóm lại, mô hình đã đạt mức ổn định cao cho `plot` và `legend`, trong khi điểm nghẽn chính nằm ở lớp `bar` (vừa dễ bỏ sót, vừa dễ tạo false positive). Do đó, các cải tiến tiếp theo nên ưu tiên cho lớp `bar`, ví dụ: tăng mẫu bar mảnh/nhỏ, điều chỉnh augmentation theo hướng tăng tương phản, và tinh chỉnh ngưỡng `conf`/`iou` để giảm nhầm lẫn với nền.

4.4.4 Phân tích định tính (Qualitative Analysis)

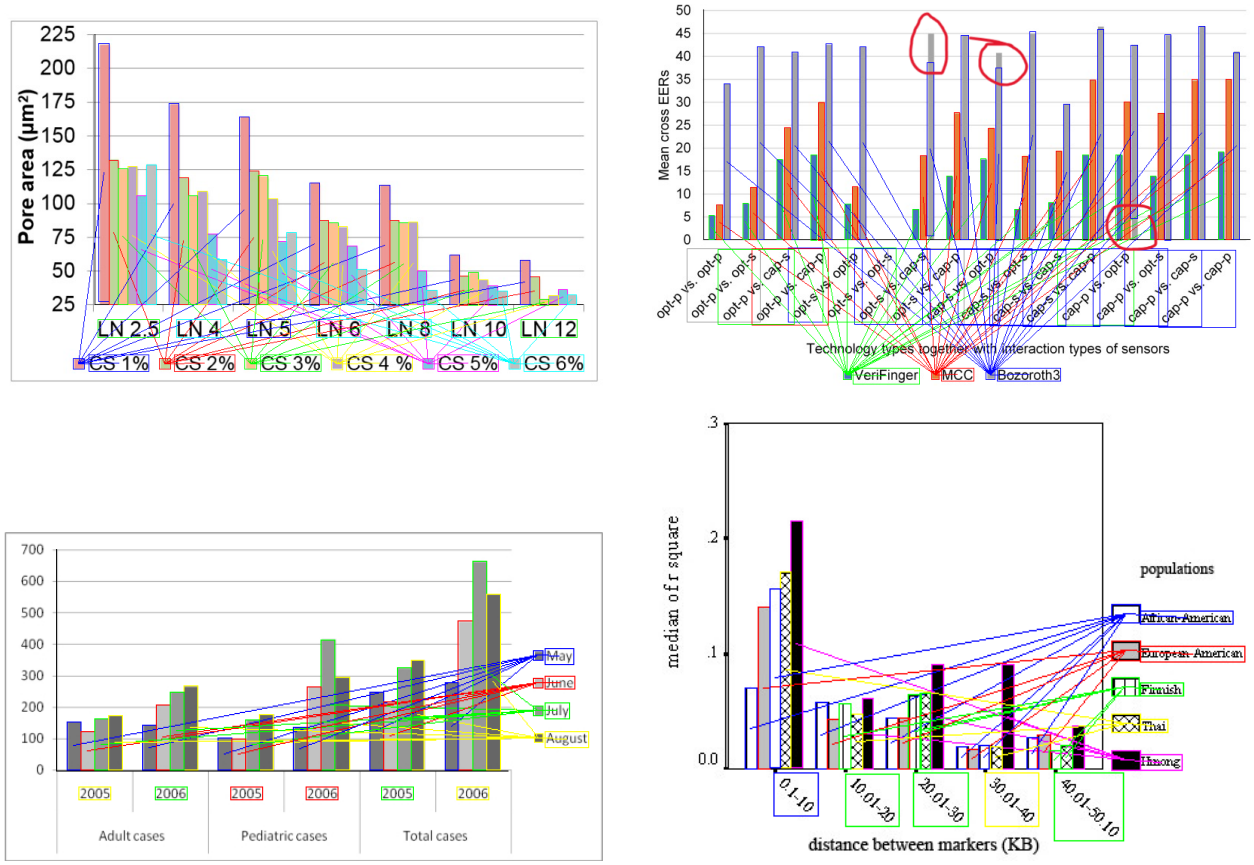
Dựa trên kết quả trực quan hoá, nhóm thực hiện đánh giá chi tiết theo các tiêu chí đặc thù của bài toán:

- **Lớp Plot:** Bounding box cần bao phủ chính xác vùng không gian Euclide của biểu đồ, tránh nhiễu từ tiêu đề (title) hoặc chú giải (legend).
- **Lớp Legend Patch:** Bounding box cần khu trú chính xác ô màu hoặc ký hiệu, không lấn sang phần văn bản mô tả.
- **Lớp Bar:** Yêu cầu khả năng phân tách (segmentation) cao, đảm bảo mỗi cột dữ liệu tương ứng với một bounding box riêng biệt.

Nhận xét: Mô hình thể hiện độ ổn định cao trên các biểu đồ cột có cấu trúc chuẩn (nền đơn sắc, khoảng cách cột rộng). Tuy nhiên, một số thách thức kỹ thuật vẫn tồn tại:

- **Vấn đề kích thước nhỏ (Small Objects):** Các cột mảnh hoặc có chiều cao thấp dễ bị bỏ sót (False Negative).
- **Vấn đề dính líu (Occlusion/Merging):** Khi các cột nằm quá sát nhau, mô hình có xu hướng gộp nhiều cột thành một bounding box lớn.

- **Nhiễu nền:** Trong trường hợp chú giải (legend) nằm đè lên vùng vẽ (plot area), sự chồng lấn giữa các đối tượng gây ra sai lệch toạ độ dự đoán.



Hình 4.4: Trực quan hoá kết quả dự đoán: (Trái) Các trường hợp mô hình hoạt động tốt; (Phải) Các lỗi điển hình như gộp cột (merge), bỏ sót (miss), hoặc nhiễu vùng chú giải.

4.4.5 Tác động đến Phân tích Trục (Task 3)

Trong kiến trúc hệ thống đề xuất, toạ độ của trục X và Y được suy diễn trực tiếp từ các cạnh của bounding box `plot`. Do đó, sai số của bước này sẽ lan truyền (error propagation) sang các bước sau:

- **Trường hợp tích cực:** Khi `plot` được phát hiện chính xác, hệ trục toạ độ được thiết lập ổn định, tạo tiền đề tốt cho việc gán nhãn tick và ước lượng thang đo.
- **Trường hợp tiêu cực:** Nếu bounding box bị lệch (do nhiễu title hoặc cắt xén vùng vẽ), vị trí trục tham chiếu sẽ bị tịnh tiến sai lệch, dẫn đến sai số hệ thống trong việc

ánh xạ toạ độ pixel sang giá trị thực.

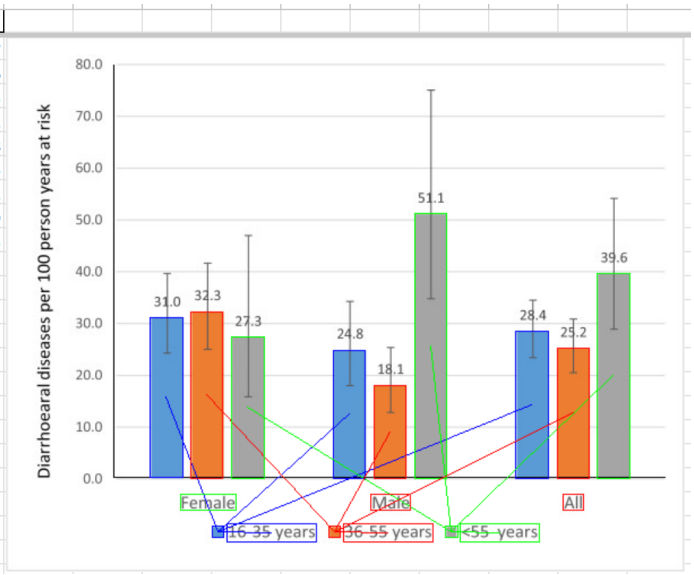
4.4.6 Tác động đến Phân tích Chú giải (Task 4)

Phân hệ Legend Analysis yêu cầu đầu vào gồm toạ độ `legend_patch` (từ YOLO) và văn bản (từ OCR). Đánh giá sơ bộ cho thấy:

- Việc tách biệt `legend_patch` thành công nhất khi chú giải nằm ngoài vùng vẽ và có bố cục lưới (grid) hoặc danh sách (list) rõ ràng.
- Các lỗi phổ biến bao gồm việc bounding box bao trùm cả phần văn bản, gây khó khăn cho việc phân loại vai trò (role classification) và ghép cặp (pairing) sau này.

4.4.7 Tiềm năng Trích xuất Dữ liệu Thô (Task 5b)

legend	x_label	value
16-35 years	Female	30,7
16-35 years	Male	24,6
16-35 years	All	28,3
36-55 years	Female	32
36-55 years	Male	17,8
36-55 years	All	25
<55 years	Female	27,2
<55 years	Male	50,9
<55 years	All	39,3



Hình 4.5: Trích xuất Dữ liệu Thô.

So sánh kết quả trích xuất và giá trị tham chiếu trên biểu đồ. Bảng kết quả trích xuất ở bên trái cho thấy cấu trúc dữ liệu phù hợp với biểu đồ cột ở bên phải: cột `legend` tương ứng với ba nhóm tuổi (*16–35 years*, *36–55 years*, *<55 years*), cột `x_label` tương ứng với ba nhóm (*Female*, *Male*, *All*), và cột `value` là giá trị của từng cột. Đối chiếu với các giá trị được gắn trực tiếp trên đỉnh cột trong biểu đồ, các số trích xuất nhìn chung khớp tốt và chỉ lệch nhỏ (xấp xỉ 0.1 – 0.3 đơn vị) ở hầu hết trường hợp. Ví dụ, nhóm *16–35 years* lần

lượt là 30.7 so với 31.0, 24.6 so với 24.8, và 28.3 so với 28.4; nhóm *36–55 years* là 32.0 so với 32.3, 17.8 so với 18.1, và 25.0 so với 25.2; nhóm *<55 years* là 27.2 so với 27.3, 50.9 so với 51.1, và 39.3 so với 39.6. Như vậy, hệ thống đã phục hồi đúng xu hướng và thứ tự các cột theo từng nhóm, trong khi sai lệch còn lại nhiều khả năng đến từ bước ước lượng thang đo pixel-to-value và/hoặc cơ chế làm tròn khi chuyển đổi sang giá trị số.

Mục tiêu cuối cùng là tái cấu trúc bảng dữ liệu từ ảnh. Mặc dù chưa tính toán chỉ số MAE/MAPE cụ thể tại thời điểm này, phân tích định tính cho thấy mối tương quan rõ rệt:

- Độ chính xác của việc đếm số lượng cột (Bar Counting) từ YOLO quyết định trực tiếp cấu trúc của bảng dữ liệu đầu ra.
- Các lỗi "merge bar" sẽ dẫn đến việc mất mát điểm dữ liệu hoặc sai lệch giá trị nghiêm trọng do gộp tổng.

4.4.8 Kế hoạch Phát triển và Đánh giá Tiếp theo

Báo cáo hiện tại tập trung vào đánh giá nền tảng thị giác máy tính. Trong giai đoạn tiếp theo, nhóm sẽ mở rộng phạm vi đánh giá sang các độ đo mức độ hệ thống:

- **Axis Analysis:** Đánh giá F1-score cho phân loại tick và sai số tương đối của thang đo (Scale Estimation Error).
- **Legend Analysis:** Đánh giá độ chính xác ghép cặp (Pairing Accuracy) giữa patch và text.
- **End-to-End:** Tính toán chỉ số MAE/MAPE trên từng điểm dữ liệu và tỷ lệ trích xuất thành công theo biểu đồ (Chart Success Rate).

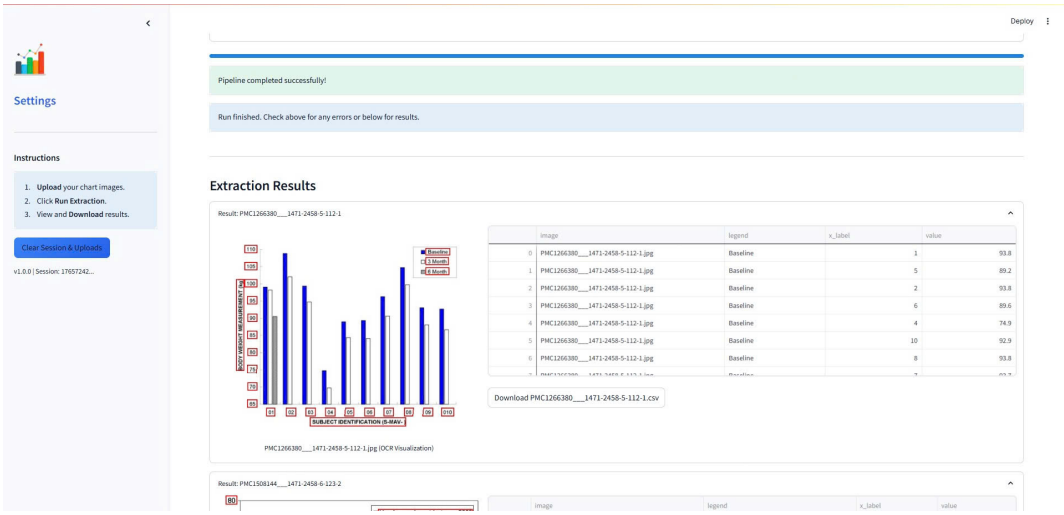
5 Demo Ứng dụng

Để chứng minh tính hiệu quả và khả năng ứng dụng thực tế của hệ thống trích xuất thông tin biểu đồ, nhóm đã phát triển một giao diện người dùng web đơn giản. Ứng dụng này cho phép người dùng cuối tương tác trực tiếp, tải lên biểu đồ và nhận kết quả trích xuất ở định dạng có cấu trúc.

5.1 Thiết kế Giao diện Người dùng (User Interface)

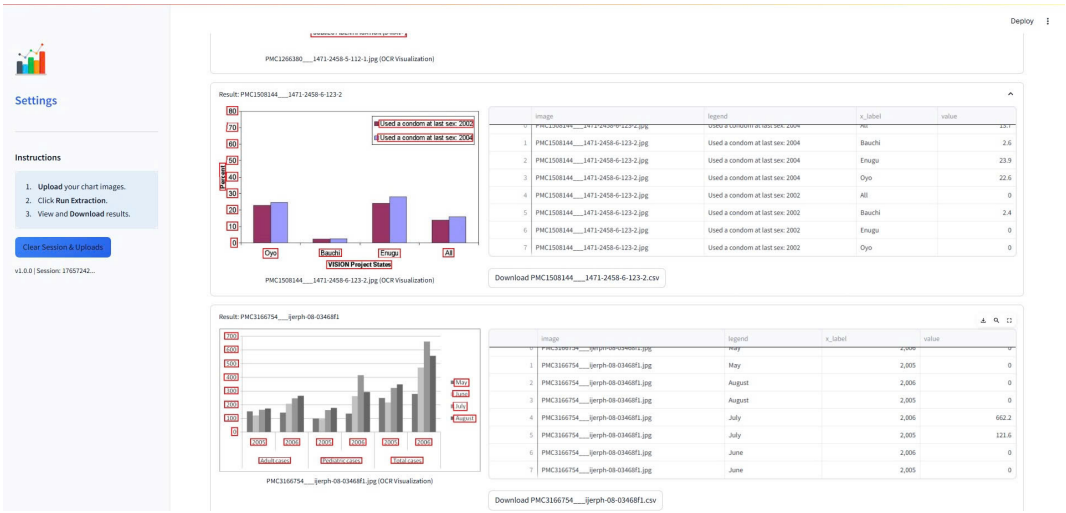
Giao diện được thiết kế tối giản, tập trung vào trải nghiệm người dùng theo quy trình ba bước đơn giản (Hình 5.1):

- Tải lên Hình ảnh Biểu đồ: Người dùng có thể kéo/thả hoặc duyệt để tải lên các tệp hình ảnh biểu đồ (hỗ trợ các định dạng như PNG, JPEG).
- Chạy Trích xuất: Kích hoạt quá trình xử lý pipeline end-to-end.
- Xem và Tải kết quả: Xem trực quan kết quả và tải về dữ liệu đã trích xuất dưới định dạng CSV.



Hình 5.1: Giao diện Web thứ 1

Hình 5.2: Giao diện Web thứ 2



Hình 5.3: Giao diện Web thứ 3

5.2 Quá trình Trích xuất và Kết quả

Quá trình demo minh họa cách hệ thống xử lý các biểu đồ cột phức tạp được trích xuất từ các tài liệu khoa học (ví dụ: PMC1508144, PMC1266300)

5.2.1 Đầu vào và Hoạt động Hệ thống

Sau khi người dùng tải lên các tệp, hệ thống sẽ xử lý chúng theo pipeline đã trình bày ở Chương 3. Màn hình xác nhận quá trình xử lý hoàn tất thành công (Hình 5.2).

Đầu vào: Các hình ảnh biểu đồ cột (Bar Chart) đa dạng về bố cục, màu sắc và mật độ dữ liệu.
Hoạt động: Hệ thống chạy lần lượt qua các giai đoạn: `YOLO + PaddleOCR` (Detection Recognition), `LayoutLMv3` (Role Classification), và Thuật toán Ghép nối (Data Association).

5.2.2 Trực quan hóa và Kết quả Đầu ra

Kết quả được trình bày dưới dạng trực quan và bảng dữ liệu có cấu trúc:

1. Trực quan hóa Dữ liệu Trích xuất (OCR Visualization):
    - Hệ thống hiển thị lại biểu đồ gốc
    - Các thành phần văn bản được phát hiện và nhận dạng (OCR) được vẽ trực tiếp lên biểu đồ để người dùng kiểm tra
    - Đây là kết quả trung gian sau Giai đoạn 1, chứng tỏ các nhãn trục (Y-axis: Percent, X-axis: Oyo, Bauchi, Enugu) và chú giải đã được phát hiện thành công
2. Bảng Dữ liệu có Cấu trúc (Structured Data Table):
    - Đây là đầu ra cuối cùng của hệ thống sau **Giai đoạn 3 (Thuật toán Ghép nối)
    - Dữ liệu được tổ chức thành các cột rõ ràng: ‘Image‘ (ID), ‘Legend‘ (Chú giải/Series), ‘x_label‘ (Nhãn trục X) và ‘value‘ (Giá trị số)

5.2.3 Tính năng Tải xuống Dữ liệu

Người dùng có thể tải xuống bảng dữ liệu có cấu trúc đã trích xuất dưới định dạng `CSV` (Comma-Separated Values) bằng nút `Download PMC...`

Việc cung cấp kết quả ở định dạng CSV giúp người dùng dễ dàng tích hợp dữ liệu biểu đồ vào các công cụ phân tích khác (như Excel, Python Pandas, R) để tái sử dụng và phân tích chuyên sâu.

6 Hạn chế và Định hướng tương lai

6.1 Hạn chế của Hệ thống Hiện tại

Mặc dù hệ thống đã đạt được mục tiêu chính là trích xuất thông tin từ các biểu đồ phổ biến (Cột, Đường) bằng cách sử dụng kiến trúc lai `YOLO` + PaddleOCR và `LayoutLMv3`, nhưng vẫn tồn tại một số hạn chế kỹ thuật cần được giải quyết:

6.1.1 Sự Lan truyền Lỗi từ Giai đoạn Đầu tiên (Error Propagation from OCR)

Vấn đề: Mặc dù đã sử dụng `PaddleOCR` và bổ sung `YOLO` Detection, hiệu suất của module Nhận dạng Ký tự Quang học (OCR) vẫn là một điểm nghẽn. Các ký tự hiếm, phông chữ lạ, hoặc văn bản có độ phân giải thấp, bị mờ có thể dẫn đến sai số từ (Word Error Rate - WER) cao

Hệ quả: Vì LayoutLMv3 và thuật toán ghép nối ở Giai đoạn 3 phụ thuộc hoàn toàn vào chuỗi văn bản và tọa độ do Giai đoạn 1 cung cấp, sai số OCR sẽ lan truyền (error propagation), gây ra lỗi phân loại vai trò (`LayoutLMv3`) hoặc lỗi liên kết dữ liệu (Data Association) nghiêm trọng.

6.1.2 Xử lý Dữ liệu Ngữ cảnh (Contextual Data Handling)

Mối quan hệ Phức tạp: Thuật toán ghép nối (Giai đoạn 3) dựa chủ yếu vào mối quan hệ hình học (tọa độ). Trong các biểu đồ có mật độ dữ liệu cao hoặc chồng lấn, chỉ dựa vào tọa độ có thể không đủ để xác định mối quan hệ chính xác giữa 'Tick Label' và 'Visual Mark'.

6.1.3 Phân tích Trục (Axis Analysis)

- **Độ nhạy của ước lượng thang đo:** Quá trình hồi quy tuyến tính để tìm thang đo (scale) rất nhạy cảm với nhiễu từ OCR (nhận diện sai số) và số lượng tick numeric thưa thớt, dễ gây ra sai số lớn khi ánh xạ ngược từ pixel sang giá trị thực.

6.1.4 Phân tích Chú giải (Legend Analysis)

- **Heuristic về không gian chưa tối ưu:** Việc ghép cặp *legend label* và *legend patch* dựa trên các quy tắc cố định (patch nằm bên trái label) dễ gặp lỗi khi xử lý các chú giải dạng lưới (grid) hoặc đa cột.

- **Vấn đề nhận diện đối tượng nhỏ:** Mô hình dễ phát sinh lỗi (False Positive/Negative) đối với các `legend_patch` có kích thước quá nhỏ, bị dính liền với văn bản hoặc nằm chồng lấn lên vùng vẽ (`plot`).
- **Giới hạn của Embedding:** Việc gán chuỗi dữ liệu (series matching) dựa trên độ tương đồng embedding có thể không chính xác khi các series có màu sắc/kết cấu (texture) gần giống nhau hoặc khi cột dữ liệu có hoạ tiết phức tạp mà mô hình chưa học được.

6.1.5 Task 5a: Phát hiện phần tử biểu đồ (YOLOv8s)

- **Thách thức phân đoạn đối tượng mật độ cao:** Lớp `bar` vẫn là thách thức lớn nhất, đặc biệt khi các cột mảnh, nằm sát nhau hoặc bị nhiễu bởi đường lưới (gridlines), dẫn đến lỗi bỏ sót (miss) hoặc gộp cột (merge).
- **Độ chính xác định vị:** Mặc dù `mAP@0.5` đạt mức khá, nhưng sự chênh lệch lớn so với `mAP@0.5:0.95` cho thấy độ khớp (IoU) của các bbox dự đoán vẫn chưa đạt độ tinh chỉnh cao.
- **Biên độ dữ liệu:** Hiệu năng mô hình suy giảm đối với các trường hợp dữ liệu "khó" (hard cases) chưa xuất hiện nhiều trong tập huấn luyện, như biểu đồ có texture lạ hoặc độ phân giải thấp.

6.1.6 Task 5b: Trích xuất dữ liệu thô (Raw Data Extraction)

- **Hiện tượng lan truyền lỗi (Error Propagation):** Là bài toán end-to-end dạng đường ống (pipeline), sai số từ bất kỳ khâu nào (YOLO, Axis, Legend) đều tích luỹ và làm sai lệch bảng dữ liệu cuối cùng.
- **Phạm vi hỗ trợ:** Hệ thống chưa xử lý triệt để các biến thể phức tạp như biểu đồ cột chồng (stacked), cột nhóm (grouped), các giá trị bằng 0 (không hiển thị bar), hoặc trường hợp trục bị ẩn.
- **Đánh giá định lượng:** Do giới hạn thời gian, các chỉ số đánh giá mức hệ thống (Chart-level success rate, MAE/MAPE toàn cục) chưa được thống kê đầy đủ.

6.2 Định hướng Phát triển Tương lai (Future Work)

Để nâng cao tính ổn định, độ chính xác và khả năng ứng dụng thực tế của hệ thống, nhóm đề xuất các hướng nghiên cứu và phát triển sau:

6.2.1 Cải thiện và Tăng cường Khả năng OCR

Chiến lược Tăng cường Dữ liệu: Xây dựng thêm bộ dữ liệu phụ (Supplementary Dataset) bao gồm các ký tự đặc biệt, đơn vị đo lường khoa học, và các phông chữ lạ thường xuất hiện trong biểu đồ để tinh chỉnh (Fine-tune) module `PaddleOCR` (hoặc thay thế bằng một mô hình OCR chuyên dụng hơn)
Đánh giá Đa mô hình: Tích hợp cơ chế đánh giá chéo (cross-validation) giữa kết quả OCR từ PaddleOCR và các gợi ý từ mô hình ngôn ngữ (Language Model) để tăng độ tin cậy của chuỗi văn bản đầu ra.

6.2.2 Mở rộng Phạm vi Biểu đồ được Hỗ trợ

Mở rộng Tập Nhãn: Mở rộng tập dữ liệu huấn luyện (Training Set) và tập nhãn để bao gồm nhiều loại biểu đồ khác hơn như Box Plots, Scatter Plots chi tiết hơn, hoặc các dạng biểu đồ chuyên ngành (ví dụ: Network Graphs).
Kiến trúc Dạng Modularity: Thiết kế lại Giai đoạn 3 theo hướng modular, nơi mỗi loại biểu đồ có một thuật toán ghép nối tối ưu riêng, thay vì một thuật toán chung cho tất cả, nhằm tăng cường khả năng xử lý các cấu trúc phức tạp.

6.2.3 Tăng cường Mối quan hệ Ngữ cảnh và Hình học

Sử dụng Vision-Language Models: Thử nghiệm tích hợp các mô hình Vision-Language (như BLIP, LLaVA) để mô hình có thể diễn giải mối quan hệ không gian và ngữ cảnh giữa các thành phần thay vì chỉ dựa vào tọa độ cố định.
Xây dựng Mô hình Đồ thị (Graph Models): Áp dụng Graph Neural Networks (GNN) để biểu diễn biểu đồ dưới dạng đồ thị, trong đó các node là các thành phần văn bản/đồ họa và các edge là mối quan hệ giữa chúng. Điều này có thể giúp giải quyết hiệu quả vấn đề chồng lấn và liên kết dữ liệu phức tạp.

6.2.4 Axis Analysis

- Dùng hồi quy robust (RANSAC/Huber) cho thang đo và kiểm tra tính nhất quán (monotonicity, spacing).

6.2.5 Legend Analysis

- Kết hợp hình học + confidence để lọc ghép cặp không chắc chắn; bổ sung cơ chế fallback khi thiếu patch/text.
- Cải thiện embedding (fine-tune/metric learning) và kết hợp thêm đặc trưng màu (HSV/histogram) để phân biệt series gần nhau.

6.2.6 Task 5a: Plot Element Detection/Classification (YOLOv8s)

- Tăng dữ liệu cho trường hợp khó (bar mảnh, sát nhau, texture) và tối ưu cho đối tượng nhỏ (imgsz cao hơn, multi-scale).
- Tinh chỉnh ngưỡng `conf`/`iou` và hậu xử lý theo cấu trúc (giảm merge bars, loại false positive trên nền).

6.2.7 Task 5b: Raw Data Extraction

- Tận dụng *value labels* (nếu có) để hiệu chỉnh thang đo và giảm sai số giá trị.
- Hoàn thiện bộ metric end-to-end theo nhiều mức (bar-level/cell-level/chart-level) và phân tích lỗi theo nguyên nhân.

Tài liệu

[1] Yue Deng, Jian Chen, Yulan Zhang, and Huajun Chen. Chart decoder: Generating textual and numeric information from chart images automatically. *ScienceDirect*, 2023. Accessed via ScienceDirect.

[2] Wenbo Li, Peng Li, and Wei Lu. ChartEye: A Deep Learning Framework for Chart Information Extraction. *arXiv preprint arXiv:2408.16123*, 2024.

[3] Jun Mao, Xinlong Wang, and Hao Chen. Chart Mining: A Survey of Methods for Automated Chart Analysis. *Computer Vision and Image Understanding*, 2021. General Survey Paper.

[4] Alejandro Palma, Raul Perez, et al. CHART-Info 2024: A dataset for Chart Analysis and Recognition. *arXiv preprint*, 2024.

[5] The ICPR 2022 Chart Analysis and Recognition Team. ICPR 2022 - CHART Competition. In *International Conference on Pattern Recognition (ICPR) 2022*. IEEE, 2022. Official Competition Website.

[6] Zhibo Wang, Yu Liu, and Bo Sun. A Comprehensive Survey on Chart Image Analysis. *arXiv preprint arXiv:1812.10636*, 2018.