**­­­­­­­BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**

**­­**

**Đề tài:**

**TÌM HIỂU VỀ MÔ HÌNH YOLO**

**VÀ ỨNG DỤNG TRONG PHÁT HIỆN KHỐI U NÃO**

**TRÊN ẢNH CHỤP MRI**

**GVHD: TS. Lê Trọng Ngọc**

**LỚP**: DHKHMT18A - 420300233001

**NHÓM:** Não Cá Vàng (nhóm 1)

**NĂM HỌC:** 2024 - 2025

**TP. Hồ Chí Minh, ngày … tháng … năm 2024**

**NHẬN XÉT CỦA GIẢNG VIÊN**

|  |  |
| --- | --- |
| **Điểm** | **Xếp loại** |
|  |  |

**Chữ ký giảng viên**

**DANH SÁCH THÀNH VIÊN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ** | **Tên** | **Ghi chú** |
| **1** | 22683171 | Phan Tôn Lộc | Nguyên | Nhóm trưởng |
| **2** | 22651091 | Nguyễn Thị Tuyết | Lan |  |
| **3** | 22651621 | Đỗ Đức Hoàn | Châu |  |
| **4** | 22676051 | Lâm Phát | Đạt |  |
| **5** | 22658841 | Đỗ Minh | Thư |  |

Mục lục

[1. Giới thiệu 1](#_Toc185368360)

[2. Khái niệm chung về u não 1](#_Toc185368361)

[3. Mô hình YOLO 1](#_Toc185368362)

[3.1. YOLOv9 2](#_Toc185368363)

[3.2. YOLOv11 3](#_Toc185368364)

[4. Các thông số độ đo 4](#_Toc185368365)

[4.1. Precision 4](#_Toc185368366)

[4.2. Recall 4](#_Toc185368367)

[4.3. F1 Score 5](#_Toc185368368)

[4.4. IoU 5](#_Toc185368369)

[4.5. Mean Average Precision (mAP) 5](#_Toc185368370)

[4.6. Loss function 6](#_Toc185368371)

[4.7. Loss curve 6](#_Toc185368372)

[4.8. Confusion Matrix 6](#_Toc185368373)

[5. Mô hình cho bài toán 6](#_Toc185368374)

[5.1. Giới thiệu phân tích bộ dữ liệu 6](#_Toc185368375)

[5.1.1. Nguồn, thông tin bộ dataset 6](#_Toc185368376)

[5.1.2. Phân tích khám phá dữ liệu 7](#_Toc185368377)

[5.2. Tiền xử lý 10](#_Toc185368378)

[5.2.1. Cắt bỏ phần nền đen, giữ lại phần não 10](#_Toc185368379)

[5.2.2. Làm sắt nét hình: 11](#_Toc185368380)

[5.2.3. Làm sáng ảnh 11](#_Toc185368381)

[5.3. Thiết lập đào tạo mô hình 11](#_Toc185368382)

[5.3.1. Train và đánh giá kết quả mô hình lần 1 12](#_Toc185368383)

[5.3.2. Cải tiến data để Train và đánh giá lần 2 13](#_Toc185368384)

[5.3.3. Xử lý tăng thêm dữ liêu bằng xử lý ảnh và Train, đánh giá mô hình lần 3 15](#_Toc185368385)

[5.3.4. Chia và cập nhật lại data tránh hiện tượng học cùng tấm ảnh và Train, đánh giá mô hình lần 4 18](#_Toc185368386)

[5.3.5. Train K-Fold Cross Validation 20](#_Toc185368387)

[6. Tổng kết 21](#_Toc185368388)

[7. Nguồn tham khảo 22](#_Toc185368389)

**Mục lục ảnh**

[Hình 1: Cấu trúc chung cơ bản của Yolo 2](#_Toc185368390)

[Hình 2: PGI và các kiến ​​trúc và phương pháp mạng liên quan. 3](#_Toc185368391)

[Hình 3: So sảnh một vài cấi trúc khác với GELAN 3](#_Toc185368392)

[Hình 4: Công thức Precision 4](#_Toc185368393)

[Hình 5: Công thức Recall 5](#_Toc185368394)

[Hình 6: Công thức F1 Sore 5](#_Toc185368395)

[Hình 7: Mô tả IoU 5](#_Toc185368396)

[Hình 8: Công thức mAP 6](#_Toc185368397)

[Hình 9: Mô tả Confusion Matrix 6](#_Toc185368398)

[Hình 10: Mô hình bài toán 6](#_Toc185368399)

[Hình 11: Mô hình các mặt cắt MRI 7](#_Toc185368400)

[Hình 12: Các mặt cắt của dữ liệu 7](#_Toc185368401)

[Hình 13: Biêủ đồ thể hiện chênh lệch image-label tập train 8](#_Toc185368402)

[Hình 14: Biểu đồ thể hiện chênh lệch image-label tập test 8](#_Toc185368403)

[Hình 15: Biểu đồ image-label tập train đã cắt image dư thừa 8](#_Toc185368404)

[Hình 16: Tỷ lệ tập train-test 9](#_Toc185368405)

[Hình 17: Biểu đồ thể hiện tỷ lệ các class 9](#_Toc185368406)

[Hình 18: Biểu đồ thể hiện mức xám trên nhiều ảnh 10](#_Toc185368407)

[Hình 19: Biểu đồ thống kê kích thước ảnh 10](#_Toc185368408)

[Hình 20: Các ảnh não đã được cắt nền 11](#_Toc185368409)

[Hình 21: Ảnh não sau tiền xử lý 11](#_Toc185368410)

[Hình 22: Results lần1 12](#_Toc185368411)

[Hình 23: F1 confidence curve tập Val lần 1 12](#_Toc185368412)

[Hình 24: Confusion matrix lần 1 13](#_Toc185368413)

[Hình 25: F1 confidence curve tập Train lần 1 13](#_Toc185368414)

[Hình 26: Chia data Train lần 2 14](#_Toc185368415)

[Hình 27: Results lần 2 14](#_Toc185368416)

[Hình 28: F1 confidence curve tập Val lần 2 14](#_Toc185368417)

[Hình 29: Confusion matrix lần 2 15](#_Toc185368418)

[Hình 30: F1 confidence curve tập Train lần 2 15](#_Toc185368419)

[Hình 31: Chia data Train lần 3 16](#_Toc185368420)

[Hình 32: Results lần 3 16](#_Toc185368421)

[Hình 33: F1 confidence curve tập Val lần 3 16](#_Toc185368422)

[Hình 34: Confusion matrix lần 3 17](#_Toc185368423)

[Hình 35: F1 confidence curve tập Train lần 3 17](#_Toc185368424)

[Hình 36: Chia data Train lần 4 18](#_Toc185368425)

[Hình 37: Results lần 4 18](#_Toc185368426)

[Hình 38: F1 confidence curve tập Val lần 4 19](#_Toc185368427)

[Hình 39: Confusion matrix lần 4 19](#_Toc185368428)

[Hình 40: F1 confidence curve tập Test lần 4 19](#_Toc185368429)

[Hình 41: F1 confidence curve tập Train lần 4 20](#_Toc185368430)

[Hình 42 Kết quả sau Train 21](#_Toc185368431)

[Hình 43 Kết quả sau test trên cùng tập dữ liệu 21](#_Toc185368432)

# Giới thiệu

U não là một loại bệnh lý nghiêm trọng khi xuất hiện các khối u bất thường phát triển trong não. Các khôi u nguy hiểm đe dọa đến sức khỏe, tính mạng con người. Hiện nay, tỷ lệ mắc u não đang có xu hướng tăng, một phần do sự gia tăng tuổi thọ và các yếu tố nguy cơ từ môi trường. Việc chuẩn đoán chủ yếu phụ thuộc vào các bác sĩ chuyên môn cao, đòi hỏi thời gian và công sức lớn để phân tích hình ảnh y khoa.

Hiện nay có nhiều nghiên cứu ứng dụng trí tuệ nhân tạo (AI) và học sâu (Deep Learning) trong y khoa, đặc biệt trong chẩn đoán và phân loại khối u não, đang nhận được sự chú ý lớn. Các mô hình học sâu không chỉ có khả năng tự động hóa việc phân tích dữ liệu mà còn cải thiện độ chính xác trong việc phát hiện các dấu hiệu bất thường nhỏ nhất. Một số tổ chức và công ty công nghệ lớn đang đầu tư mạnh vào các giải pháp AI hỗ trợ y tế, nhằm giảm tải cho đội ngũ y bác sĩ và nâng cao hiệu quả điều trị

Nhóm thực hiện đề tài với mong muốn hỗ trợ, tiết kiệm thời gian cho bác sĩ bằng cách sử dụng trí tuệ nhân tạo hỗ trợ bác sĩ phân loại khối u (Glioma, Meningioma, Pituitary), đồng thời xác định vị ví khối u đó.

# Khái niệm chung về u não

Khối u não là một khối u phát triển trong hệ thần kinh trung ương (CNS), có thể là ác tính (ung thư) hoặc lành tính (không ung thư). Khối u não có thể phát triển nhanh hoặc chậm, và khoảng một phần ba trong số đó là ung thư. Dù là ung thư hay không, khối u não có thể ảnh hưởng nghiêm trọng đến chức năng não và sức khỏe nếu nó đủ lớn để chèn ép lên các dây thần kinh, mạch máu hoặc mô xung quanh. Các khối u phát triển trực tiếp trong não được gọi là khối u nguyên phát, trong khi những khối u di căn từ các cơ quan khác đến não gọi là khối u thứ cấp hay khối u não di căn.

**Trong ảnh chụp T1**

U thường xuất hiện giảm tín hiệu (tối hơn) so với mô não bình thường.

Một số u (như có xuất huyết, melanin, hoặc chất béo) có thể xuất hiện **tăng tín hiệu** (sáng hơn).

Khối u thường sẽ có một lớp biên bao xung quanh ( đối với các khói u đơn giản).

# Mô hình YOLO

YOLO (You Only Look Once), một mô hình phát hiện đối tượng và phân đoạn hình ảnh phổ biến, được phát triển bởi Joseph Redmon và Ali Farhadi tại Đại học Washington. Ra mắt vào năm 2015, YOLO nhanh chóng trở nên phổ biến vì tốc độ và độ chính xác cao.

Thuật toán YOLO chia hình ảnh đầu vào thành một lưới các ô và đối với mỗi ô, nó dự đoán xác suất xuất hiện của một đối tượng và tọa độ hộp giới hạn của đối tượng. Nó cũng dự đoán lớp của đối tượng. Không giống như các bộ phát hiện đối tượng hai giai đoạn như R-CNN và các biến thể của nó, YOLO xử lý toàn bộ hình ảnh trong một lần, giúp nó nhanh hơn và hiệu quả hơn.

YOLO đã được phát triển thành nhiều phiên bản , chẳng hạn như YOLOv1, YOLOv2, YOLOv3, YOLOv4, …YOLOv9, YOLOv10 và YOLOv11. Mỗi phiên bản đều được xây dựng dựa trên phiên bản trước với các tính năng nâng cao như độ chính xác được cải thiện, xử lý nhanh hơn và xử lý tốt hơn các đối tượng nhỏ.

**Thuật toán YOLO: Nó hoạt động như thế nào?**

Ý tưởng cơ bản đằng sau YOLO là chia hình ảnh đầu vào thành một lưới các ô và, đối với mỗi ô, dự đoán xác suất xuất hiện của một đối tượng và tọa độ hộp giới hạn của đối tượng. Quá trình YOLO có thể được chia thành một số bước:

1. Hình ảnh đầu vào được truyền qua CNN để trích xuất các đặc điểm từ hình ảnh.

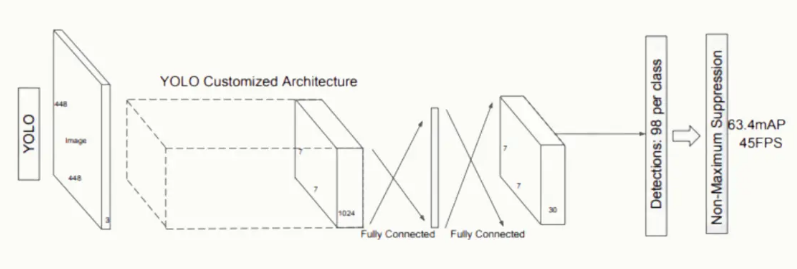
2. Các tính năng sau đó được truyền qua một loạt các lớp được kết nối đầy đủ, dự đoán xác suất lớp và tọa độ hộp giới hạn.

3. Hình ảnh được chia thành một lưới các ô và mỗi ô có nhiệm vụ dự đoán một tập hợp các hộp giới hạn và xác suất lớp.

4. Đầu ra của mạng là một tập hợp các hộp giới hạn và xác suất lớp cho mỗi ô.

5. Các hộp giới hạn sau đó được lọc bằng thuật toán hậu xử lý có tên là loại bỏ không tối đa để loại bỏ các hộp chồng lấn và chọn hộp có xác suất cao nhất.

6. Đầu ra cuối cùng là một tập hợp các hộp giới hạn được dự đoán và nhãn lớp cho từng đối tượng trong hình ảnh.



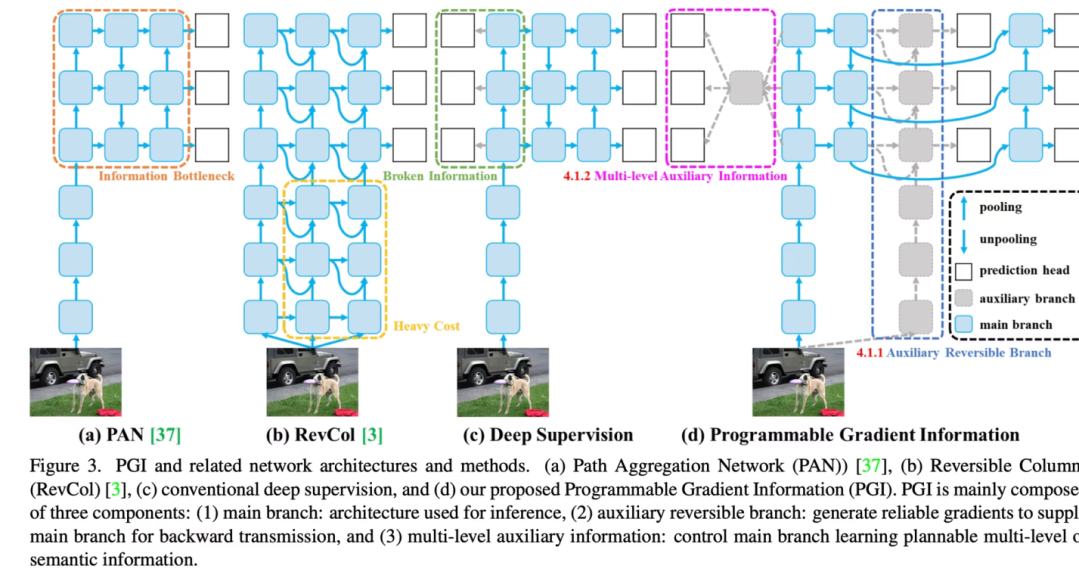
Hình 1: Cấu trúc chung cơ bản của Yolo

## YOLOv9

YOLOv9 giới thiệu các phương pháp cải tiến như Thông tin Gradient có thể lập trình (PGI) và Mạng tổng hợp lớp hiệu quả tổng quát (GELAN).

1. **Programmable Gradient Information (PGI)**

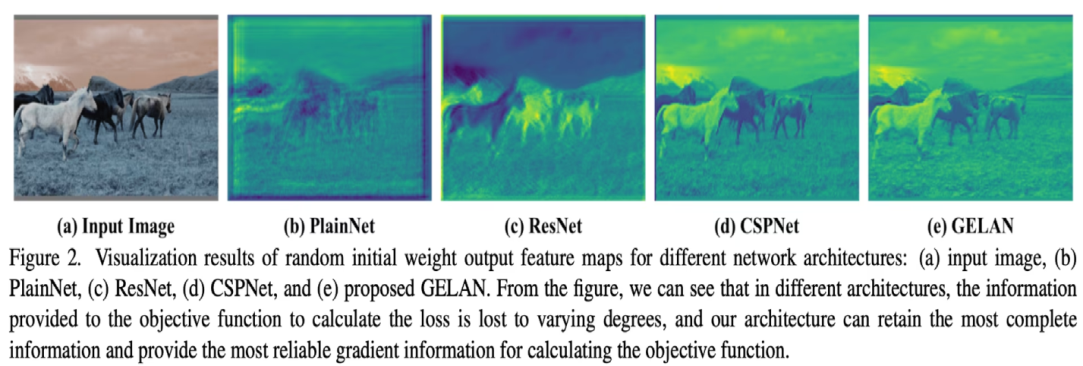
PGI là một khái niệm mới được giới thiệu trong YOLOv9 để giải quyết vấn đề tắc nghẽn thông tin, đảm bảo bảo toàn dữ liệu cần thiết trên các lớp mạng sâu. Điều này cho phép tạo ra các gradient đáng tin cậy, tạo điều kiện cập nhật mô hình chính xác và cải thiện hiệu suất phát hiện tổng thể



Hình 2: PGI và các kiến ​​trúc và phương pháp mạng liên quan.

1. **Generalized Efficient Layer Aggregation Network (GELAN)**

GELAN đại diện cho một tiến bộ kiến trúc chiến lược, cho phép YOLOv9 đạt được hiệu quả sử dụng tham số và tính toán vượt trội. Thiết kế của nó cho phép tích hợp linh hoạt các khối tính toán khác nhau, giúp YOLOv9 có thể thích ứng với nhiều ứng dụng mà không ảnh hưởng đến tốc độ hoặc độ chính xác.



Hình 3: So sảnh một vài cấi trúc khác với GELAN

## YOLOv11

YOLOv11 mô hình cung cấp hiệu suất tiên tiến (SOTA) trên nhiều tác vụ, bao gồm phát hiện , phân đoạn , ước tính tư thế , theo dõi và phân loại , tận dụng khả năng trên nhiều ứng dụng và miền AI khác nhau.

Các tính năng:

Trích xuất tính năng nâng cao: YOLO11 sử dụng kiến trúc xương sống và cổ được cải tiến, giúp tăng cường khả năng trích xuất tính năng để phát hiện đối tượng chính xác hơn và thực hiện tác vụ phức tạp.

Được tối ưu hóa cho hiệu quả và tốc độ: YOLO11 giới thiệu các thiết kế kiến trúc tinh tế và quy trình đào tạo được tối ưu hóa, mang lại tốc độ xử lý nhanh hơn và duy trì sự cân bằng tối ưu giữa độ chính xác và hiệu suất.

Độ chính xác cao hơn với ít tham số hơn: Với những tiến bộ trong thiết kế mô hình, YOLO11m đạt được Độ chính xác trung bình (mAP) cao hơn trên tập dữ liệu COCO trong khi sử dụng ít hơn 22% tham số so với YOLOv8m , giúp tính toán hiệu quả mà không ảnh hưởng đến độ chính xác.

Khả năng thích ứng trong nhiều môi trường: YOLOv11 có thể được triển khai liền mạch trên nhiều môi trường khác nhau, bao gồm các thiết bị biên, nền tảng đám mây và hệ thống hỗ trợ NVIDIA GPU đảm bảo tính linh hoạt tối đa.

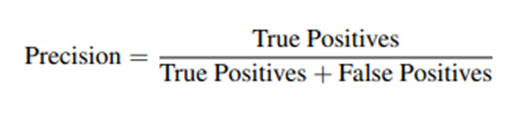
Phạm vi rộng các tác vụ được hỗ trợ: Cho dù đó là phát hiện đối tượng, phân đoạn thể hiện, phân loại hình ảnh, ước tính tư thế hay phát hiện đối tượng theo hướng (OBB), YOLO11 được thiết kế để giải quyết nhiều thách thức khác nhau về thị giác máy tính.

# Các thông số độ đo

## Precision

Độ chính xác tỷ lệ phát hiện dương tính (true positive) trên tổng số lần phát hiện dương tính (positive). Số liệu này phản ánh khả năng của mô hình trong việc phân loại chính xác các loại khối u mà không tạo ra kết quả dương tính giả (false positive). Độ chính xác cao cho thấy tỷ lệ phát hiện sai thấp hơn, điều này rất quan trọng trong các ứng dụng quan trọng về an toàn.

Công thức:

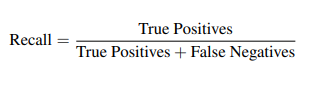
****

Hình 4: Công thức Precision

## Recall

Tỷ lệ số lần phát hiện dương tính thực (true positive) trên tổng số trường hợp dương tính thực tế (true positives and false negatives). Tỷ lệ càng cao cho thấy mô hình có hiệu quả trong hầu hết các trường hợp liên quan, ngay cả trong các tình huống khó khăn với các dữ liệu bị che khuất hoặc nhìn thấy được một phần.

Công thức:

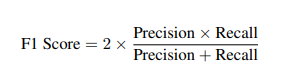
****

Hình 5: Công thức Recall

## F1 Score

Là giá trị trung bình hài hòa của độ chính xác (precision) và độ thu hồi (recall), cung cấp đánh giá cân bằng về hiệu suất của mô hình trong khi xem xét cả kết quả dương tính giả và kết quả âm tính giả. Giúp phản ánh cả độ precision và recall trong một chỉ số duy nhất.

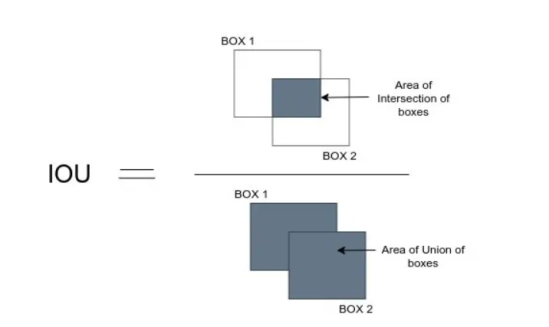
Công thức:



Hình 6: Công thức F1 Sore

## IoU

IoU là một phép đo định lượng sự chồng chéo giữa bounding box dự đoán và bounding box thực tế. Nó đóng vai trò cơ bản trong việc đánh giá độ chính xác của việc định vị đối tượng.



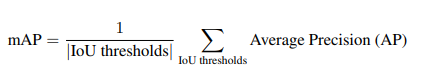
Hình 7: Mô tả IoU

## Mean Average Precision (mAP)

Là một chỉ số đánh giá phổ biến được sử dụng trong bài toán phát hiện đối tượng, tính toán độ chính xác trung bình trên nhiều ngưỡng Intersection over Union (IoU). Hai mức độ của mAP thường được báo cáo là:

**mAP50**: Độ chính xác trung bình được tính tại ngưỡng IoU là 0.5, chủ yếu đánh giá độ chính xác của việc phân loại.

**mAP50-95**: Độ chính xác trung bình được tính trên các ngưỡng IoU từ 0.5 đến 0.95, đại diện cho một tiêu chuẩn đánh giá khắt khe hơn về cả độ chính xác của vị trí và phân loại.



Hình 8: Công thức mAP

## Loss function

Loss function trong là một thành phần quan trọng giúp tối ưu hóa bài toán phát hiện đối tượng, bao gồm các thành phần con để cân bằng giữa việc định vị (localization), phân loại (classification), và dự đoán độ tin cậy của đối tượng (objectness).

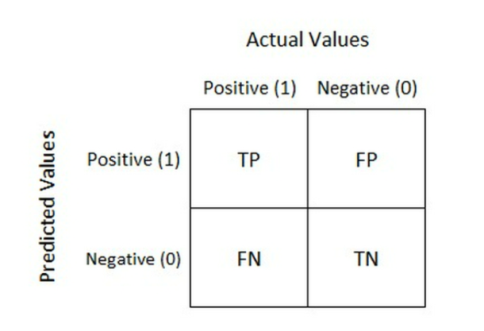
## Loss curve

Loss curve là biểu đồ thể hiện sự thay đổi giá trị mất mát của mô hình trong quá trình huấn luyện và kiểm tra theo thời gian (epochs). Nó giúp đánh giá quá trình học, phát hiện hội tụ, overfitting, hoặc underfitting của mô hình. Đường cong càng giảm chứng tỏ mô hình càng tốt.

## Confusion Matrix

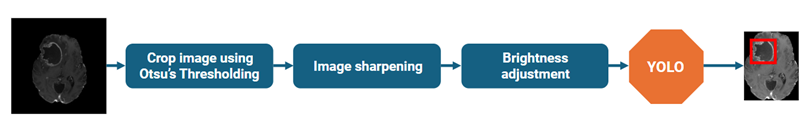
Confusion Matrix cho thấy hiệu suất của mô hình trong việc phân loại, làm rõ mối quan hệ giữa nhãn thực tế và nhãn dự đoán.

Dạng chuẩn hóa giúp xác định rõ các khu vực dự đoán chính xác cao và các lỗi cần cải thiện, hỗ trợ đánh giá toàn diện khả năng phát hiện phương tiện của mô hình.



Hình 9: Mô tả Confusion Matrix

# Mô hình cho bài toán



Hình 10: Mô hình bài toán

## Giới thiệu phân tích bộ dữ liệu

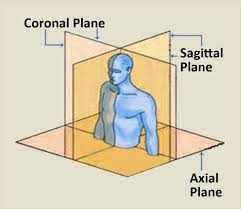
### Nguồn, thông tin bộ dataset

Nguồn bộ dataset được lấy từ kaggle [Brain tumor object detection datasets](https://www.kaggle.com/datasets/davidbroberts/brain-tumor-object-detection-datasets/data).

Thông tin bộ data:

Bao gồm tập ảnh train và tập labels/ tọa độ bouding box khoanh vùng khối u trong ảnh não MRI.

Tập ảnh JPG được xuất với kích thước gốc, phân tách thành 3 mặt phẳng (Axial, Coronal, Sagittal).

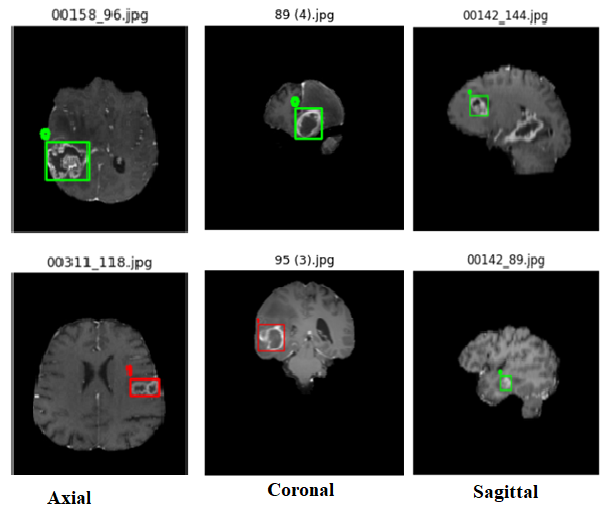


Hình 11: Mô hình các mặt cắt MRI

Các khối u được dán nhãn bằng tay sử dụng công cụ <https://makesense.ai>.

Tọa độ bounding box và labels MGMT positive (u ác tính) đã được đánh dấu trên ~ 400 hình ảnh cho mỗi mặt phẳng trong chuỗi T1wCE từ tập dữ liệu RSNA-MICCAI.

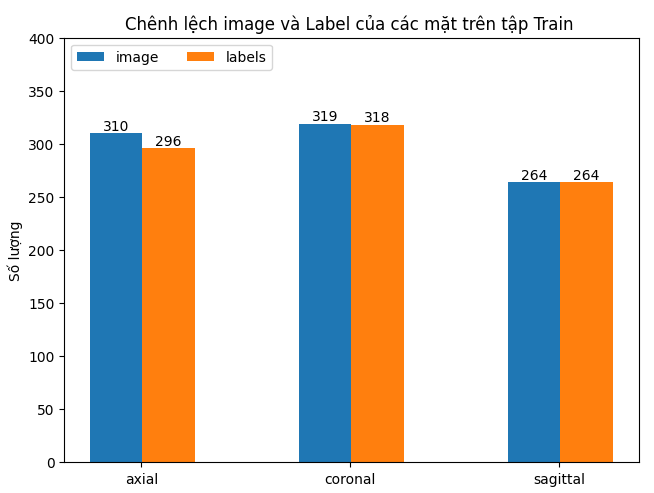
### Phân tích khám phá dữ liệu



Hình 12: Các mặt cắt của dữ liệu

* **Thống kê số lượng ảnh, label**

Trước khi cắt dữ liệu



Hình 13: Biêủ đồ thể hiện chênh lệch image-label tập train

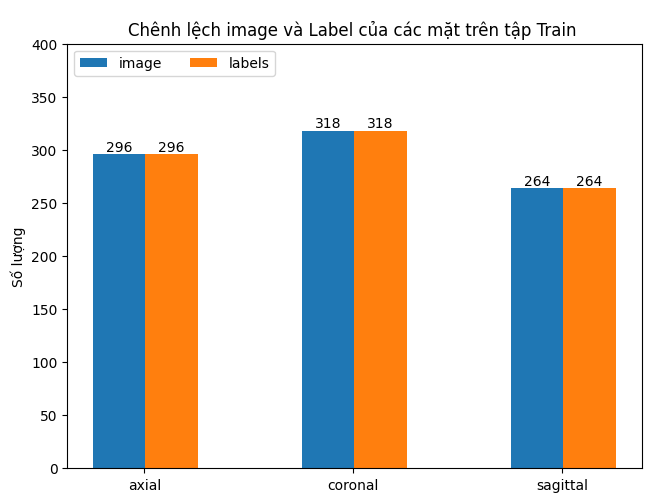
* Số lượng image-label trong tập train có sự chênh lệch cần loại bỏ các ảnh chưa đánh label.



Hình 14: Biểu đồ thể hiện chênh lệch image-label tập test

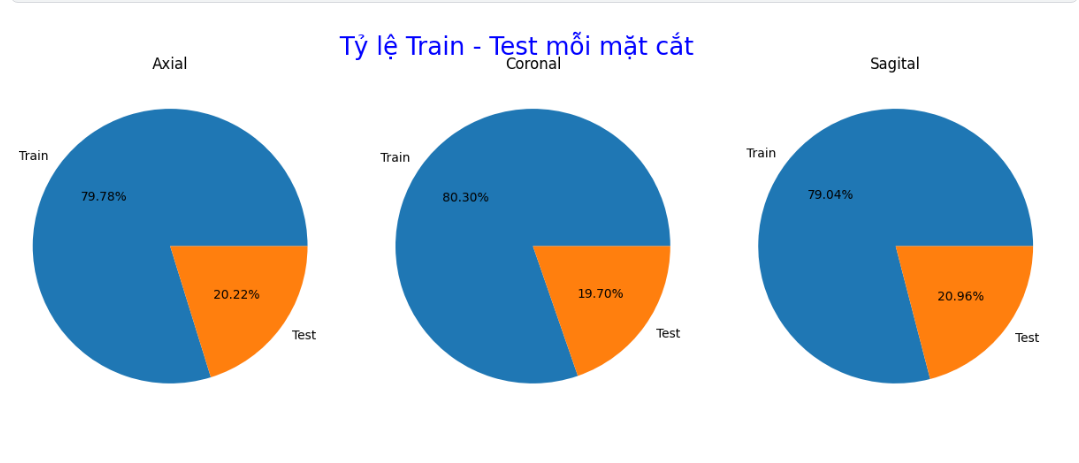
* Số lượng image-label trong tập test đã cân bằng không cần xử lý thêm

Sau khi cắt dữ liệu

****

Hình 15: Biểu đồ image-label tập train đã cắt image dư thừa

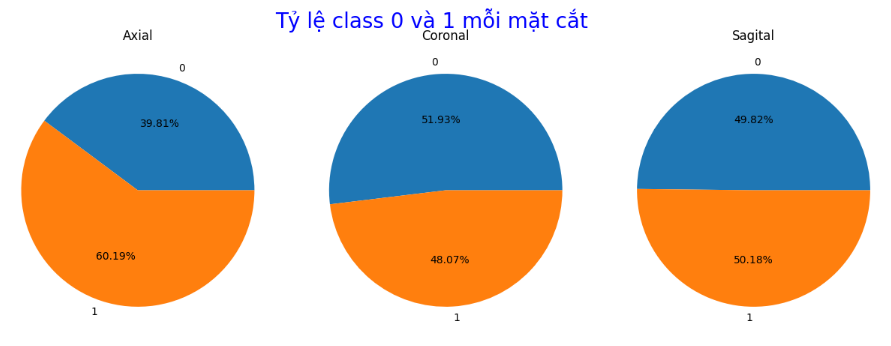
* **Kiểm tra tỷ lệ dữ liệu Train-Test**

****

Hình 16: Tỷ lệ tập train-test

Tỷ lệ giữa tập train-test các mặt cắt đều vào khoảng 80-20 đây là tỷ lệ hợp lý.

* **Kiểm tra tỷ lệ class trong mỗi mặt cắt của tập train**

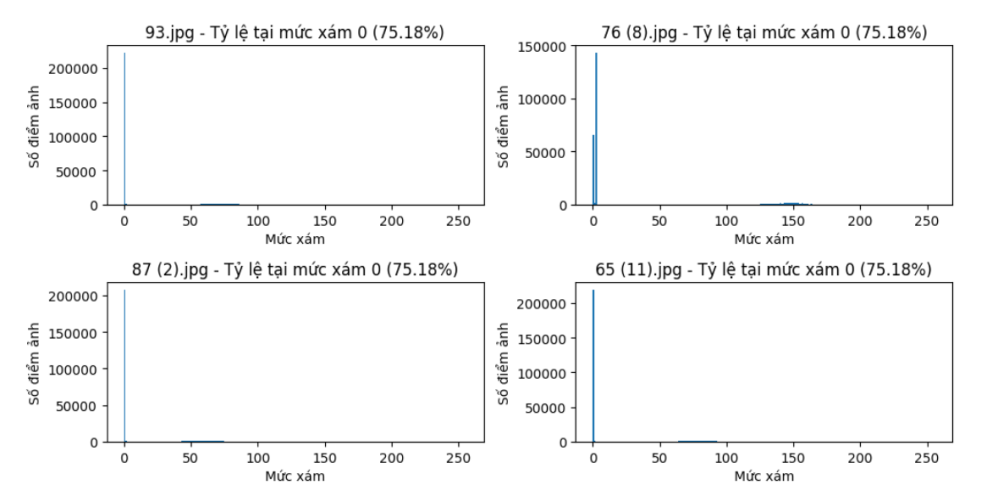
****

Hình 17: Biểu đồ thể hiện tỷ lệ các class

Tỷ lệ 2 class trong mặt cắt Coronal, Sagittal tỷ lệ  cân bằng cao.

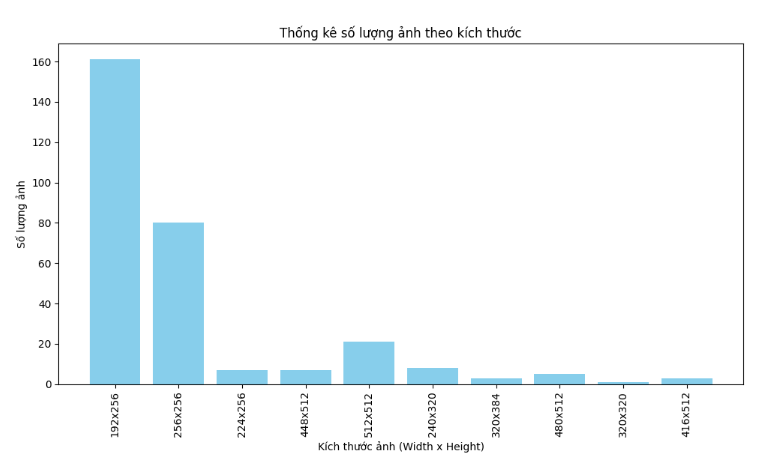
Tỷ lệ 2 class mặt cắt Axial có sự mất cân bằng nhẹ 40-60 nhưng vẫn nằm khoảng có thể chấp nhận được.

* **Kiểm tra mức xám trong ảnh**



Hình 18: Biểu đồ thể hiện mức xám trên nhiều ảnh

* Mức xám tập trung chủ tại mức 0 do nền đen dư thừa lớn, không tập trung vào chủ thể gốc cần phải xử lý nền này.
* **Kiểm tra kích thước ảnh**

**­**

Hình 19: Biểu đồ thống kê kích thước ảnh

Kích thước ảnh trong tập data không đều với nhiều mô hình cần tinh chỉnh lại kích thước đầu vào.Nhưng với mô hình Yolo đã được tích hợp sẵn việc thay đổi kích thước.

## Tiền xử lý

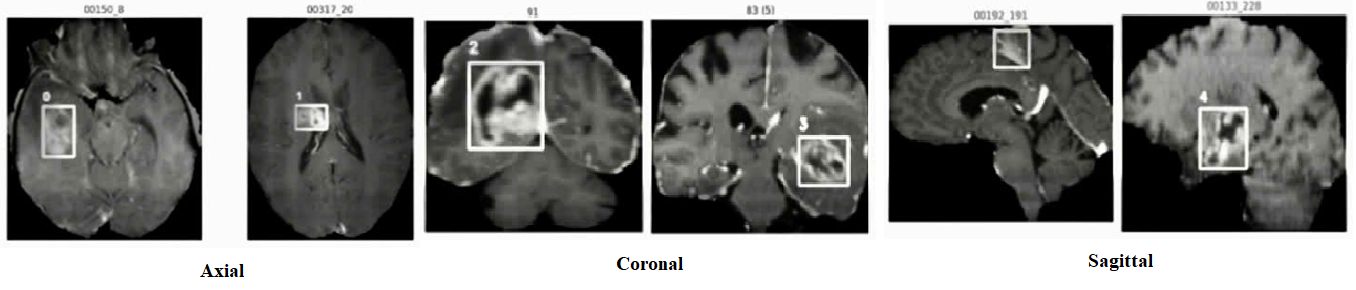
Sau khi kiểm tra các bức xám trong ảnh ta có thể thấy các phần đen của background ảnh đang chiếm đa số nên dữ liệu đang bị nhiễu chưa có thể tập trung chính xác vào chủ thể (não). Nên để dữ liệu được tập trung vào chủ thể hơn, ta thực hiện cắt các phần đen của background.

### Cắt bỏ phần nền đen, giữ lại phần não

Trong quá trình chuẩn bị dữ liệu cho bài toán phát hiện vật thể, việc tối ưu hóa dữ liệu đầu vào là rất quan trọng nhằm cải thiện hiệu quả và độ chính xác của mô hình. Chúng em đã triển khai thực hiện nhiệm vụ cắt bỏ phần nền đen không cần thiết từ các hình ảnh và điều chỉnh lại nhãn (bounding box) để phù hợp với hình ảnh đã cắt.

Thuật toán phân ngưỡng Otsu được sử dụng để phân biệt nội dung chính (phần não) của ảnh với nền đen. Sau khi xác định vùng chứa nội dung quan trọng (phần não), hình ảnh được cắt gọn lại để loại bỏ các phần dư thừa. Đồng thời, nhãn tương ứng của từng hình ảnh được cập nhật lại để phản ánh chính xác tọa độ và kích thước của các đối tượng sau khi cắt.

Phương pháp này giúp giảm thiểu nhiễu, tăng tốc độ xử lý và đảm bảo mô hình học sâu tập trung vào các đặc trưng cần thiết trong ảnh. Đây là một bước chuẩn bị dữ liệu quan trọng trong các bài toán như phát hiện khối u não, phát hiện vật thể hoặc các ứng dụng xử lý hình ảnh tương tự.



Hình 20: Các ảnh não đã được cắt nền

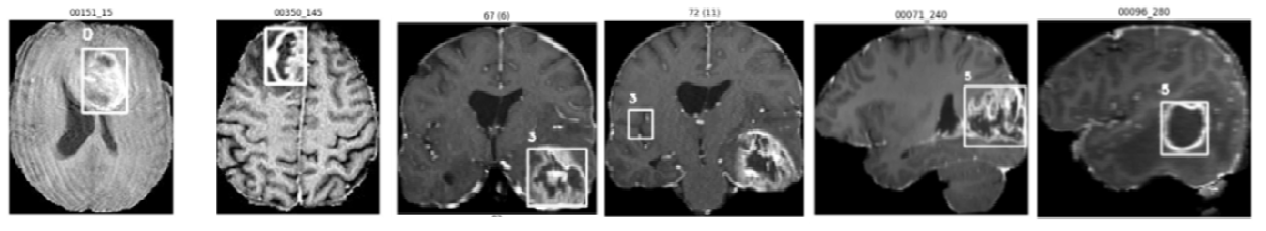
### Làm sắt nét hình:

Tạo một kernel sắc nét 3x3 được tạo ra dưới dạng một ma trận numpy. Kernel này có giá trị âm ở các cạnh và giá trị dương ở giữa, giúp tăng cường độ tương phản và làm nổi bật các cạnh trong ảnh, từ đó tạo hiệu ứng làm sắc nét.

Sau đó, duyệt qua từng ảnh trong danh sách ảnh đầu vào. Với mỗi ảnh, hàm lọc fifter2D của OpenCV được sử dụng để áp dụng kernel sắc nét vào ảnh. Hàm này thực hiện phép toán convolution giữa ảnh và kernel, giúp làm sắc nét các chi tiết trong ảnh.

### Làm sáng ảnh

Để tăng cường độ sáng, hàm sử dụng hàm cv2.convertScaleAbs, trong đó tham số beta được thiết lập là 50, có thể điều chỉnh để tăng thêm độ sáng cho các pixel trong ảnh. Hàm này cộng một giá trị cố định vào tất cả các pixel của ảnh, giúp làm sáng ảnh lên.



Hình 21: Ảnh não sau tiền xử lý

## Thiết lập đào tạo mô hình

Chọn ra 2 phiên bản của YOLO: YOLOv9 và YOLOv11, xem xét và đánh giá trên phiên bản cũ và phiên bản mới nhất.

Sau khi train với các phiên bản YOLO khác nhau , nhận thấy kết quả train không có thay đổi đáng kể.

### Train và đánh giá kết quả mô hình lần 1

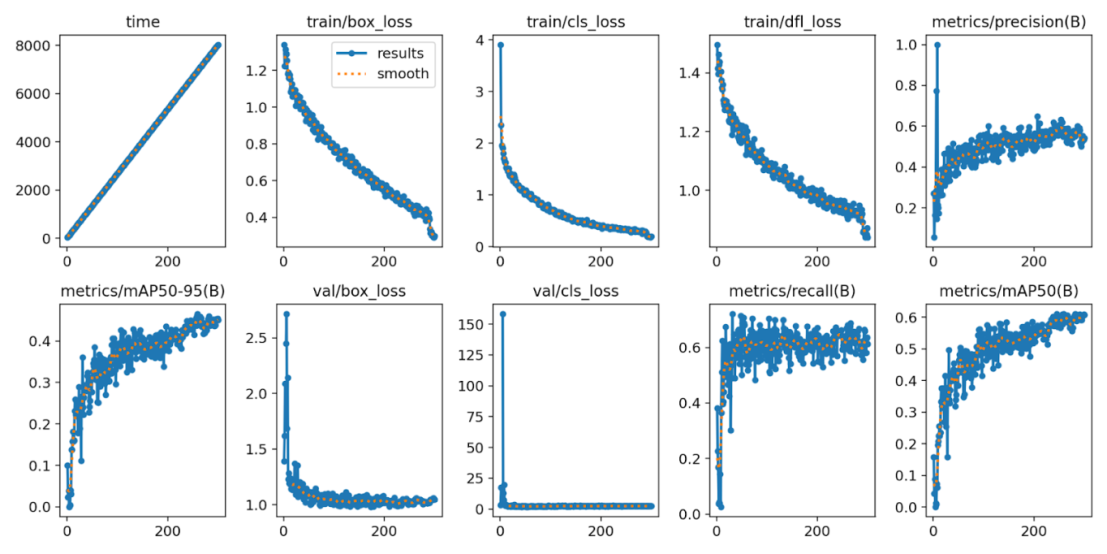
Data: Sử dụng data gốc

Model: yolo11l.pt

Thông số train epochs=300, batch=32, imgsz=640, device=[0, 1]

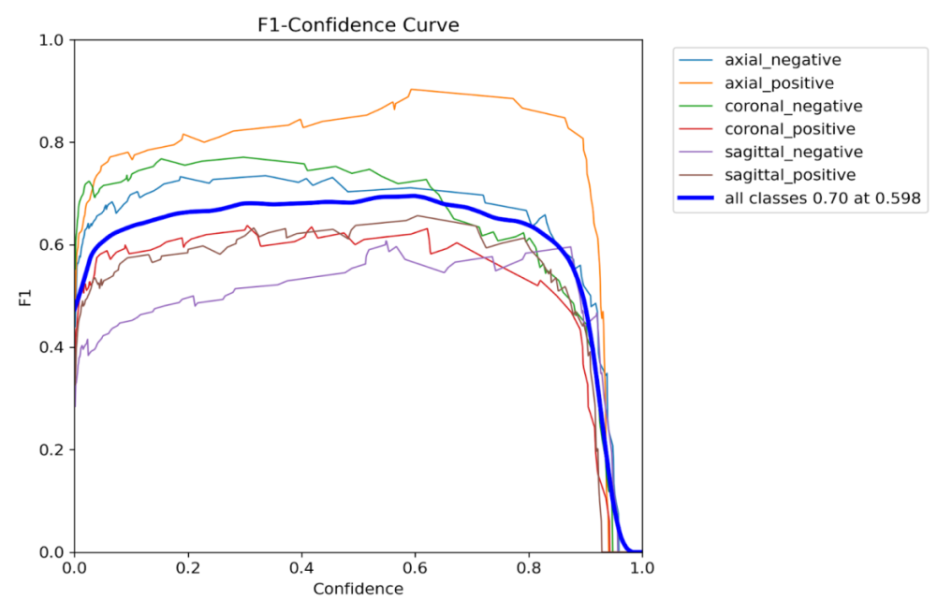
Link code: [Train lần 1](https://www.kaggle.com/code/phantnlcnguyn/brain-tumor-object-detection-by-yolov11/notebook)

Results:

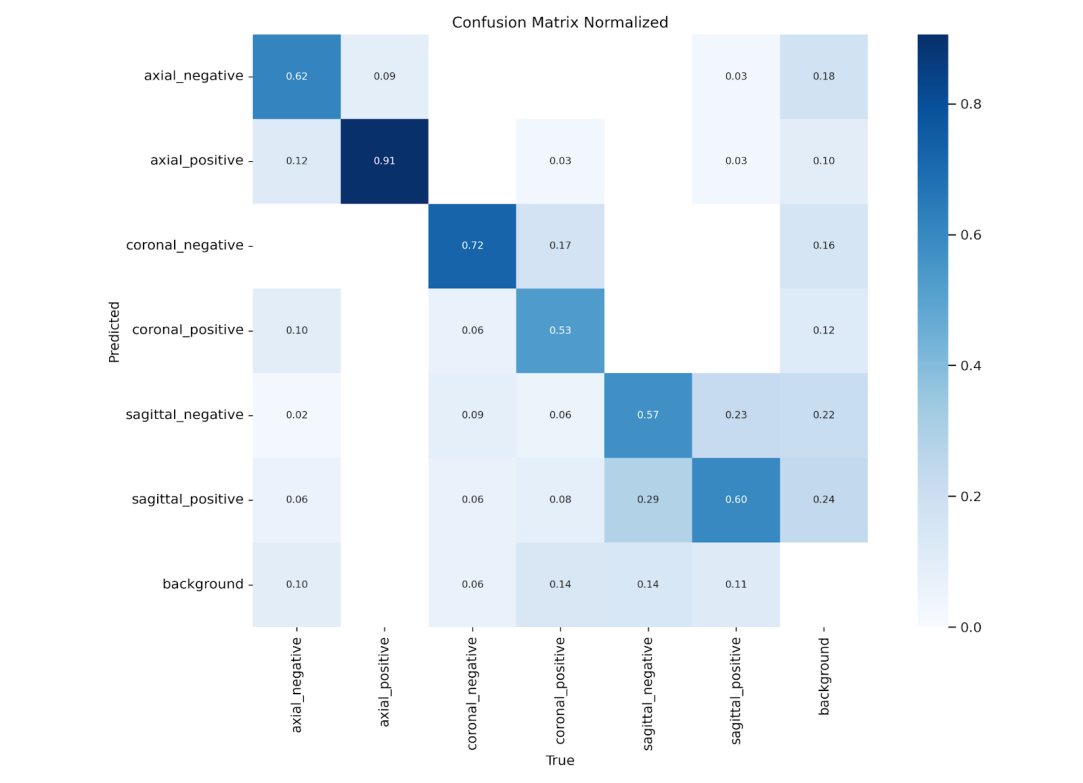


Hình 22: Results lần1

Đánh giá trên tập Validation

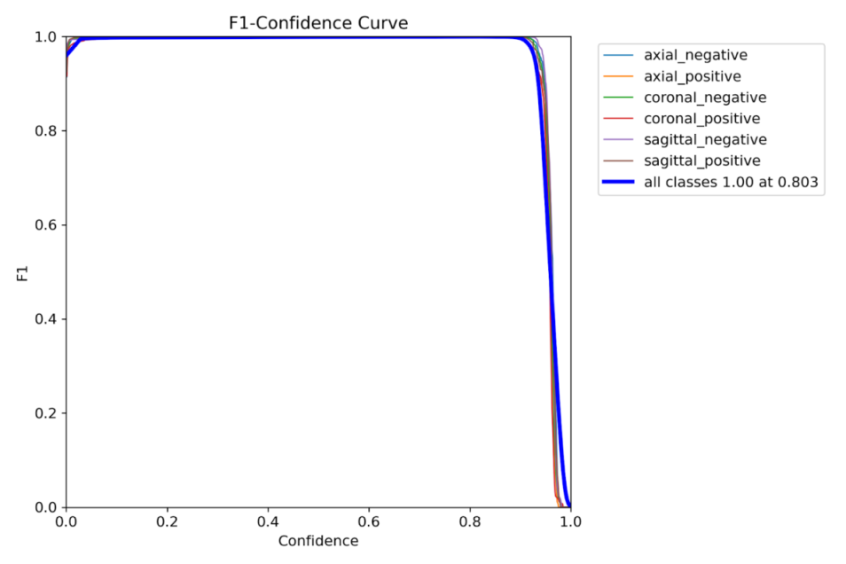


Hình 23: F1 confidence curve tập Val lần 1



Hình 24: Confusion matrix lần 1

Đánh giá tập train



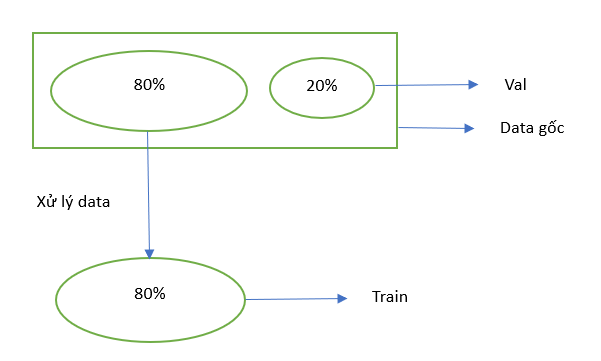
Hình 25: F1 confidence curve tập Train lần 1

Kết quả trên tập Validation thấp, F1 của tập Validation và F1 của tập Train có sự chênh lệch lớn, có thể khẳng định đang bị overfitting.

Vấn đề train thấp là do thiếu data, dữ liệu khan hiếm, data chỉ hơn 1000 ảnh. Có sự khác nhau giữa những ảnh ở tập Train và Test, về độ sáng tối, hình dạng não,... Ảnh ở tập train dễ học và nhận biết hơn sao với ảnh ở tập Test.

### Cải tiến data để Train và đánh giá lần 2

Data: Xử lý chia lại data gốc



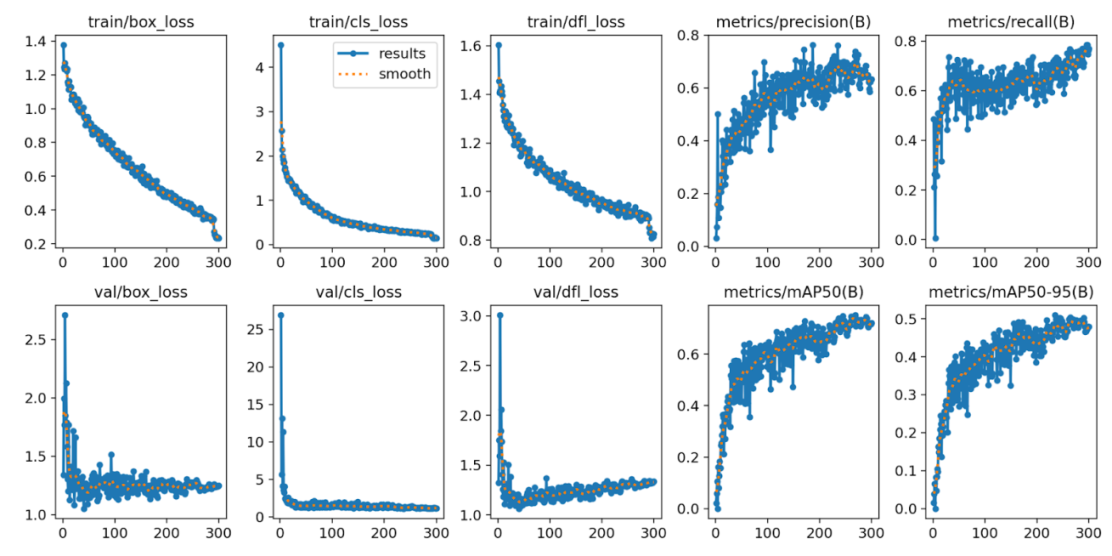
Hình 26: Chia data Train lần 2

Model: yolov9c.pt

Thông số train epochs=300, batch=32, imgsz=640, device=[0, 1]

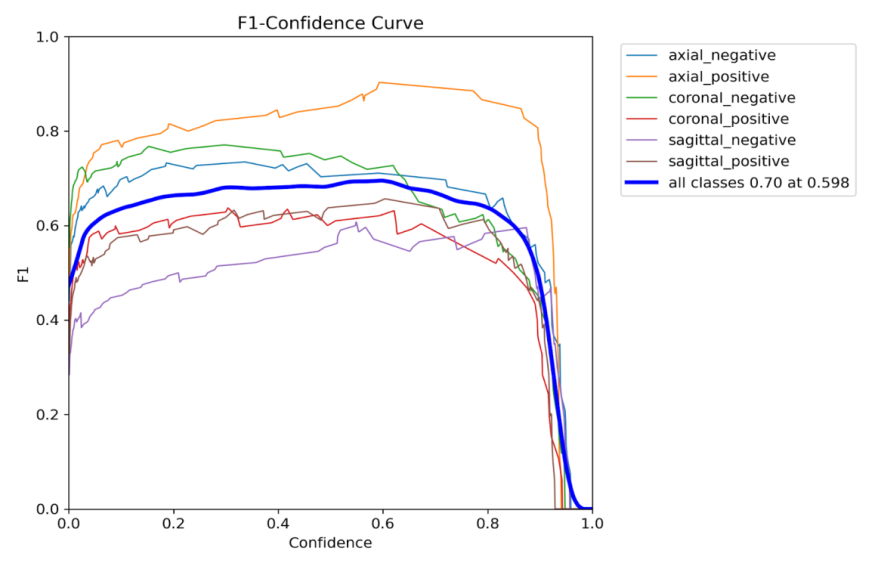
Link code: [Train lần 2](https://www.kaggle.com/code/lannguyenthituyet/brain-tumor-yolo09-train-test-80-20)

Results:

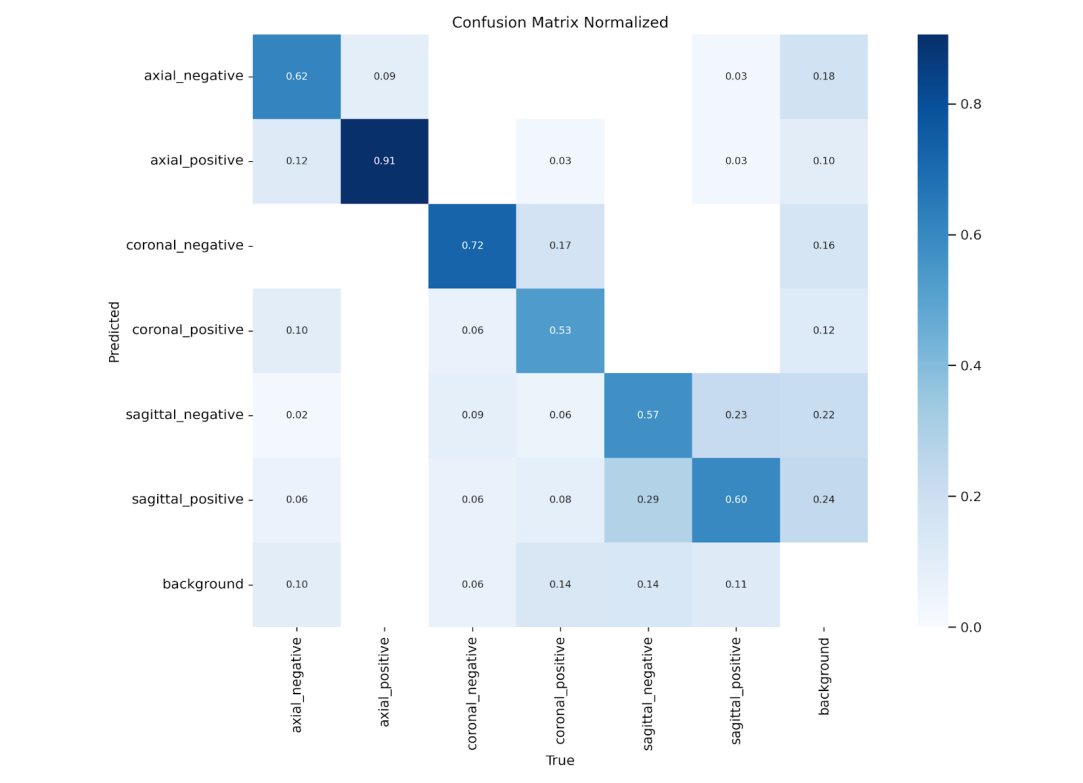


Hình 27: Results lần 2

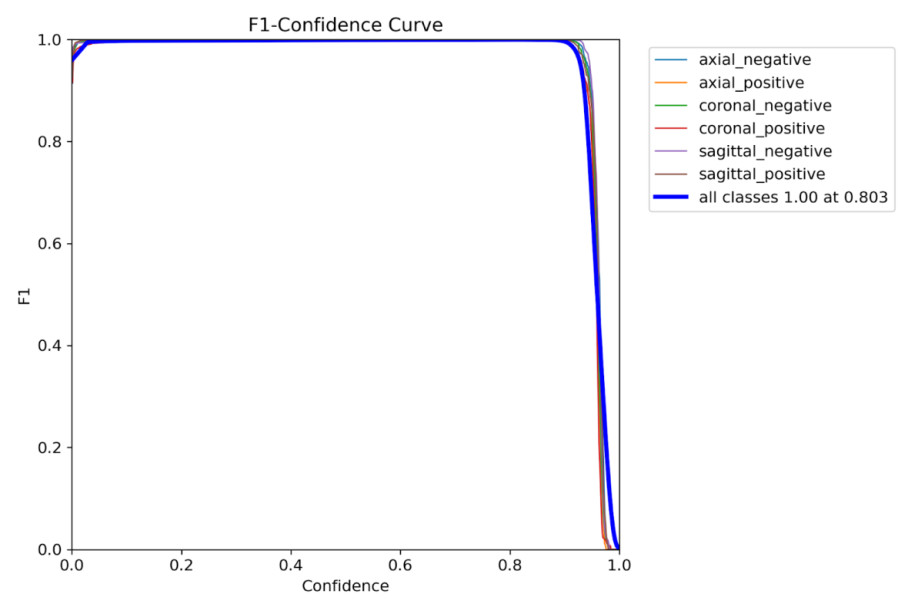
Đánh giá trên tập Validation



Hình 28: F1 confidence curve tập Val lần 2



Hình 29: Confusion matrix lần 2

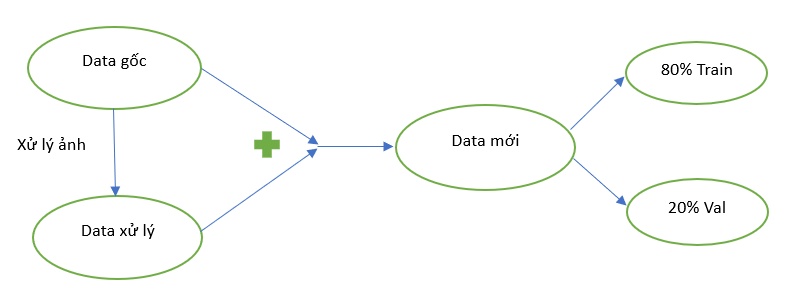


Hình 30: F1 confidence curve tập Train lần 2

Kết quả học và dự đoán trên tập validation được khả quan hơn, khắc phục được phần nào hiện tượng overfitting. Với F1=0.7 tương đối ổn trong khi data còn ít dữ liệu.

### Xử lý tăng thêm dữ liêu bằng xử lý ảnh và Train, đánh giá mô hình lần 3

Data:



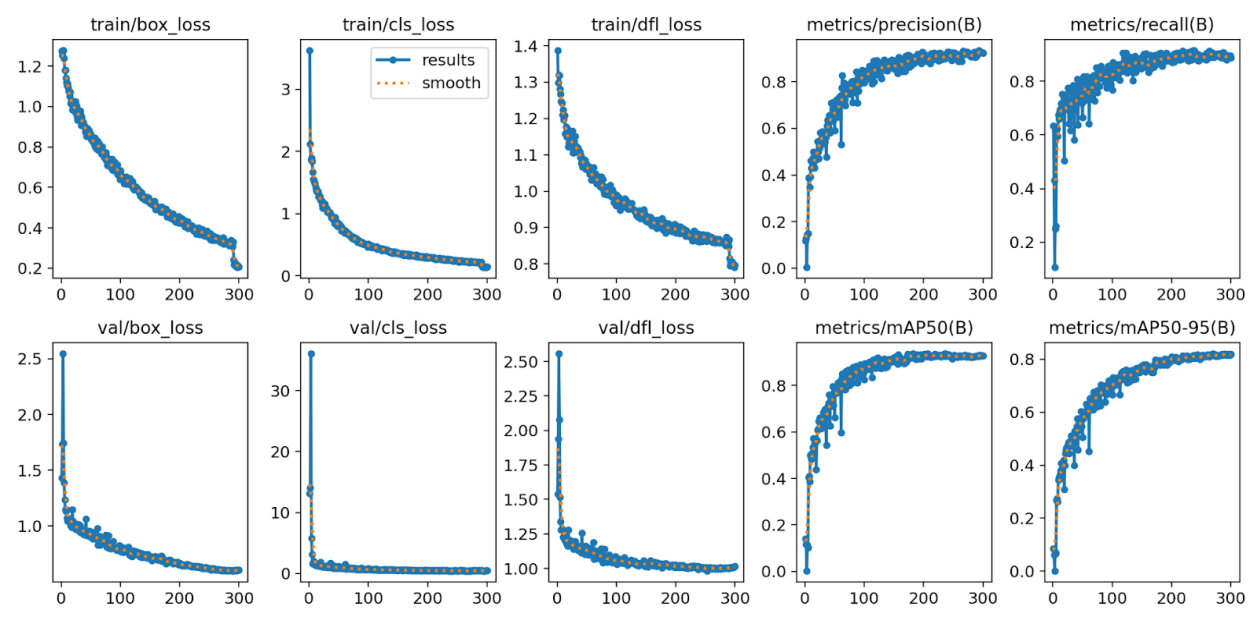
Hình 31: Chia data Train lần 3

Model: yolov9c.pt

Thông số train epochs=300, batch=32, imgsz=640, device=[0, 1]

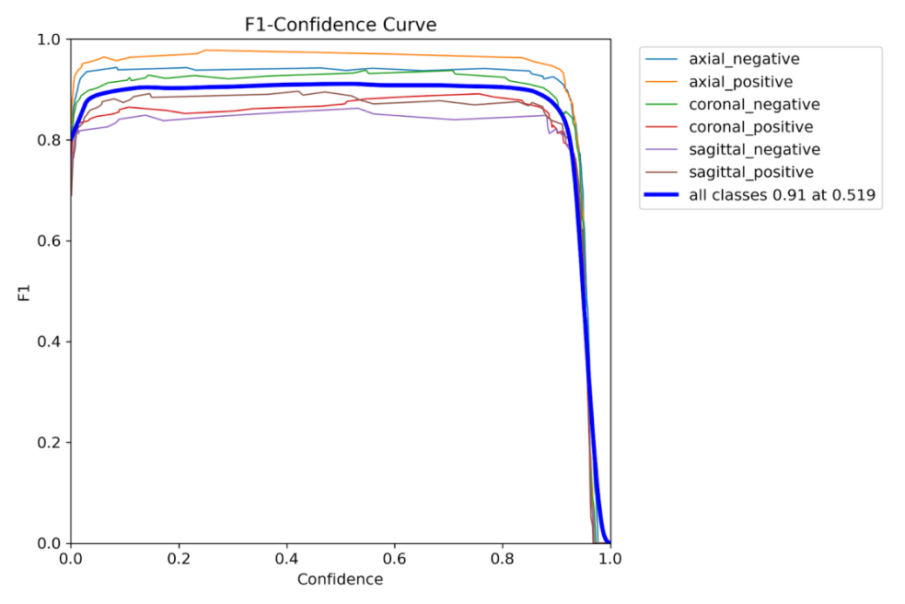
Link code: [Train lần 3](https://www.kaggle.com/code/lannguyenthituyet/train-data-brain-tumor-80-image-processing)

Results:

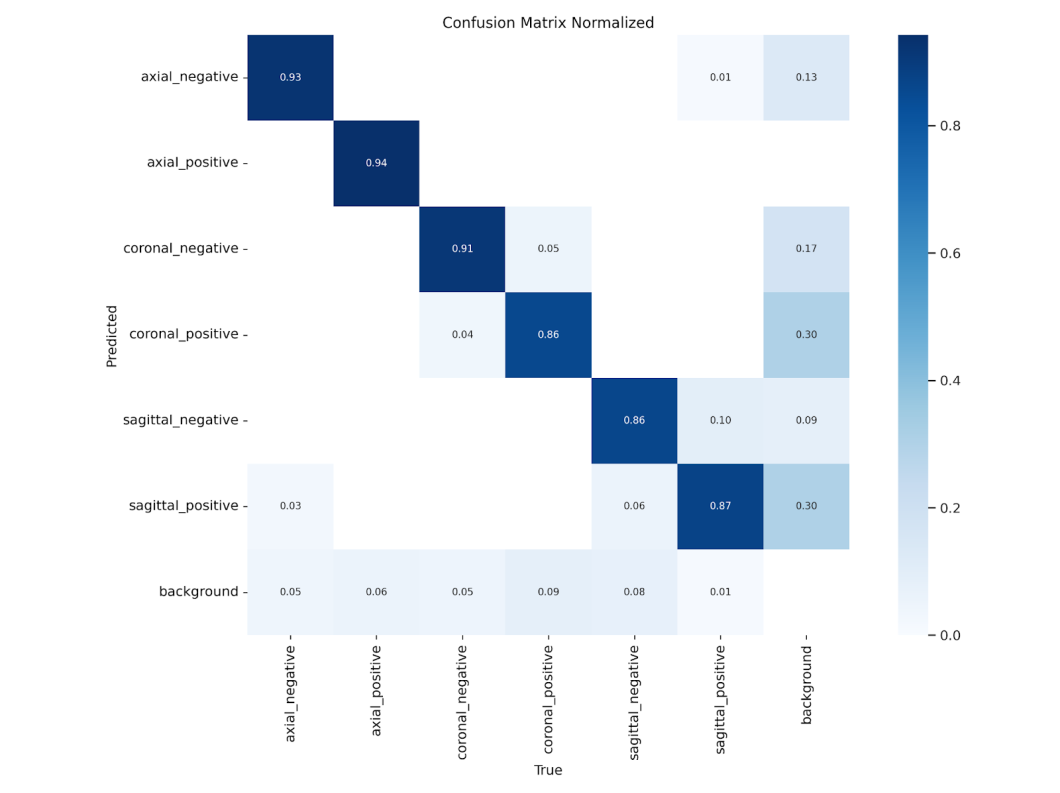


Hình 32: Results lần 3

Đánh giá trên tập Validation

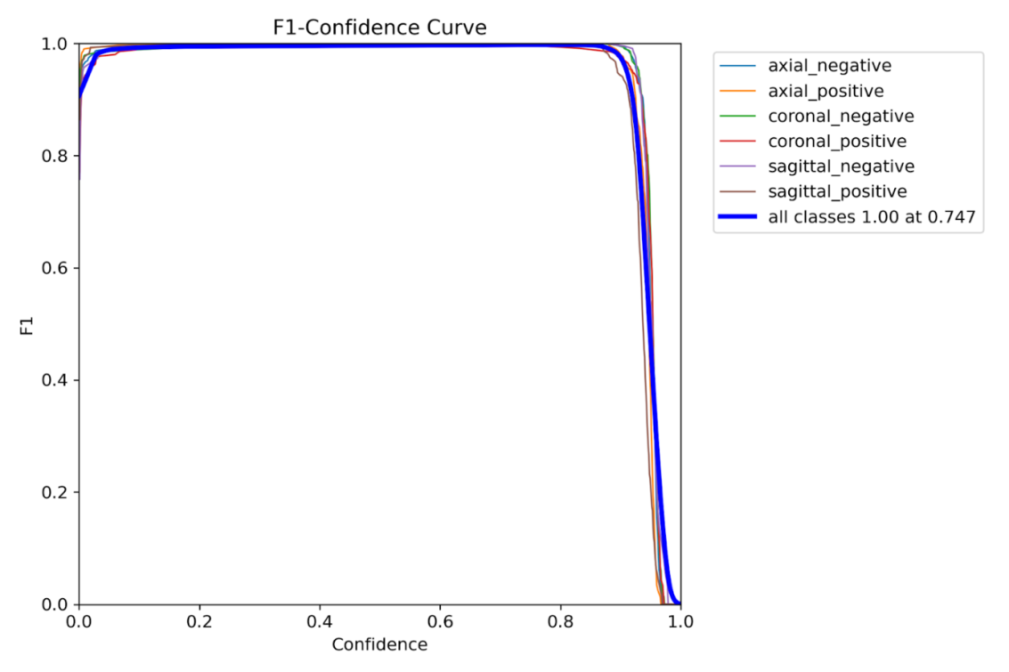


Hình 33: F1 confidence curve tập Val lần 3



Hình 34: Confusion matrix lần 3

Đánh giá trên tập Train

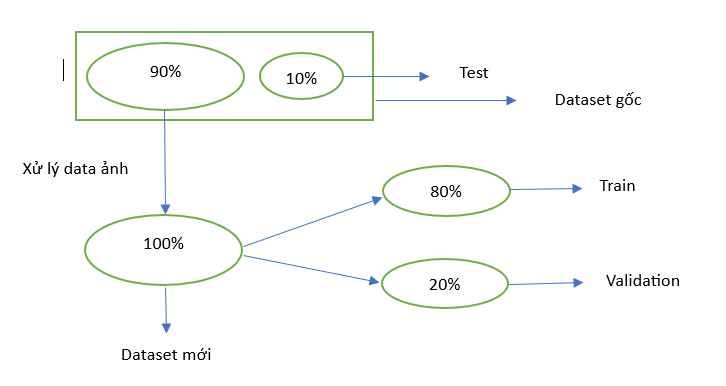


Hình 35: F1 confidence curve tập Train lần 3

Kết quả F1 trên tập Validation cao, giải quyết được phần nào thiếu data, nhưng đang nghi ngờ giữa ảnh gốc và ảnh đã xử lý có sự tương đồng và model đang predict trên chính dữ liệu được học.

### Chia và cập nhật lại data tránh hiện tượng học cùng tấm ảnh và Train, đánh giá mô hình lần 4

Data:



Hình 36: Chia data Train lần 4

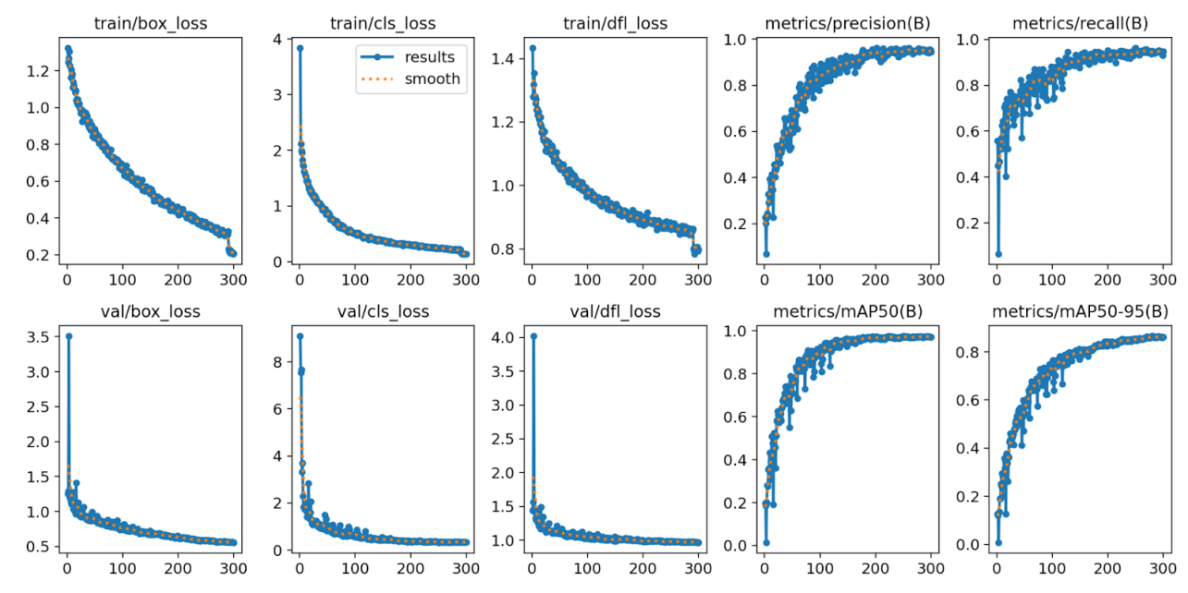
Model: yolov9c.pt

Thông số train epochs=300, batch=32, imgsz=640, device=[0, 1]

Link code train, đánh giá trên tập val: [Link 1](https://www.kaggle.com/code/lannguyenthituyet/train-yolo9-train-90-test-10)

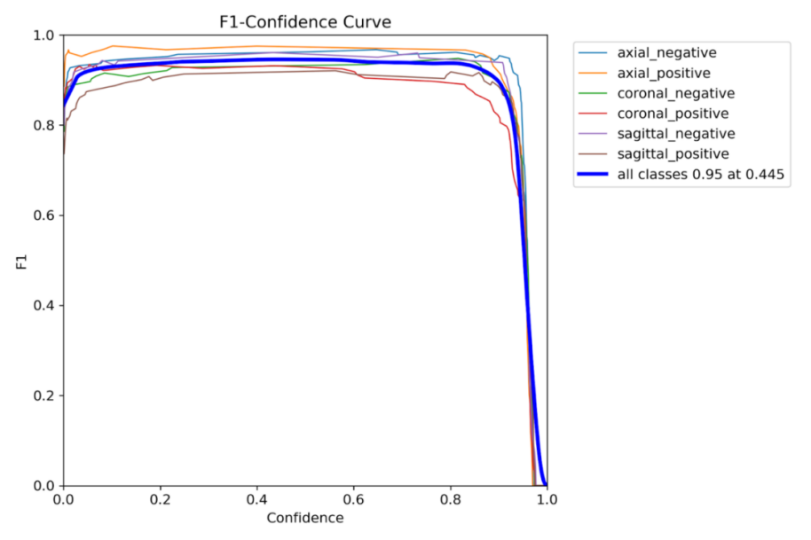
Link code đánh giá trên tập test, train: [Link 2](https://www.kaggle.com/code/lannguyenthituyet/train-val-on-test-and-train-data-of-90-10)

Results:

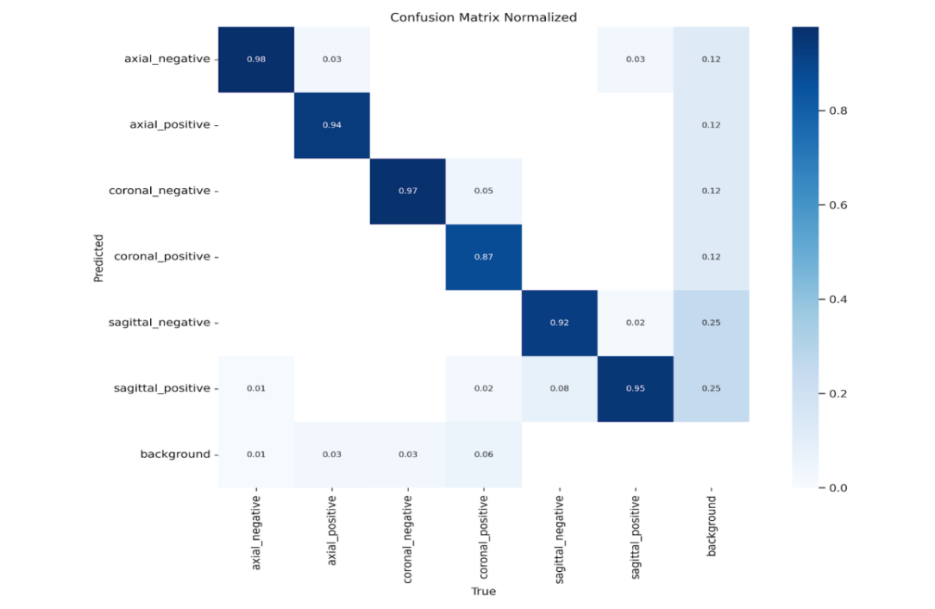


Hình 37: Results lần 4

Đánh giá trên tập Validation

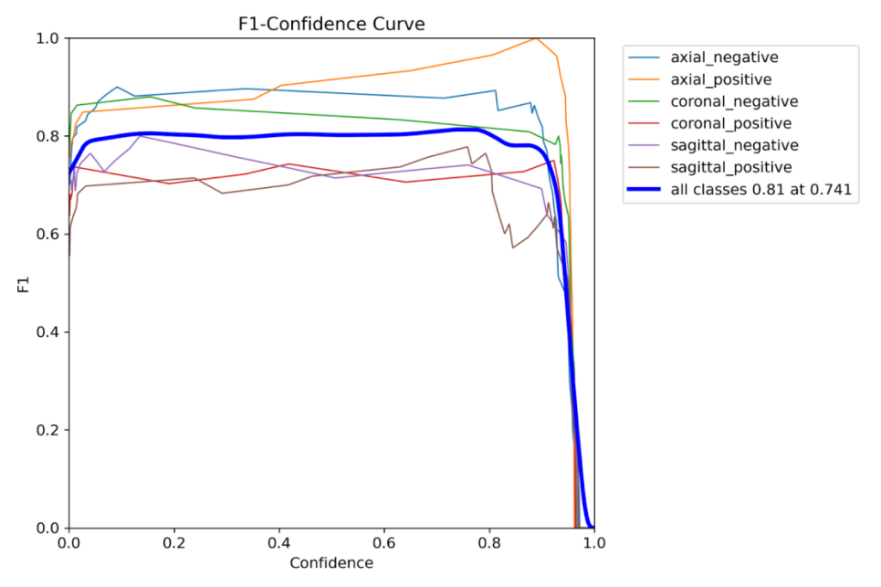


Hình 38: F1 confidence curve tập Val lần 4



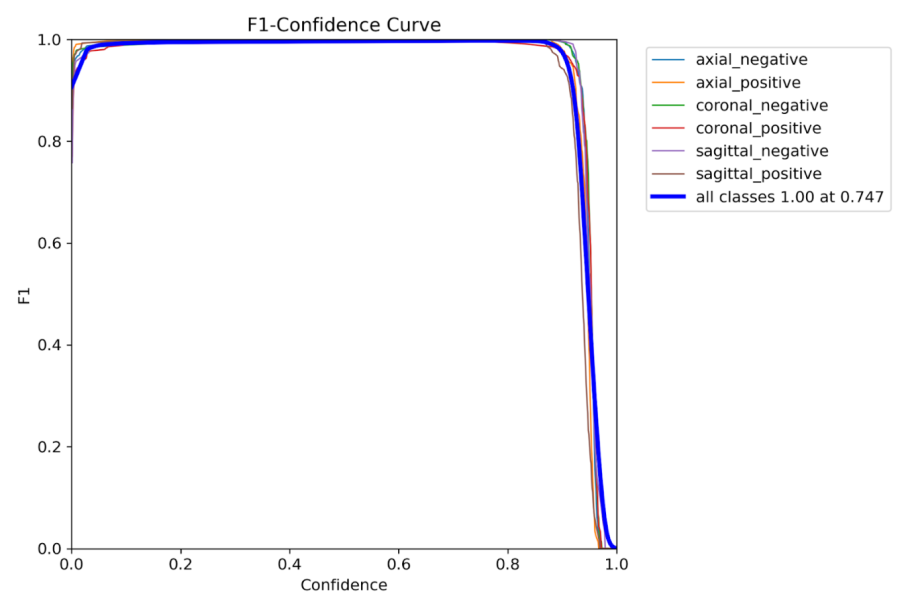
Hình 39: Confusion matrix lần 4

Đánh giá trên tập Test



Hình 40: F1 confidence curve tập Test lần 4

Đánh giá trên tập Train



Hình 41: F1 confidence curve tập Train lần 4

Giải quyết vấn đề nghi ngờ trên: khẳng định cách chia data như trên đang có hiện tượng đánh giá trên chính ảnh được học.

Tỷ lệ F1 trên tập train, val, test không chênh lệch nhiều. Không xảy ra overfit. Mô hình học tốt và dự đoán tương đối cao với ảnh chưa được học.

### Train K-Fold Cross Validation

Nhóm tiến hành sử dụng K-Fold Cross Validation vì đây là kỹ thuật đặc biệt phù hợp khi dữ liệu ít và cần đánh giá mô hình một cách tổng quát hơn. Phương pháp này chia dữ liệu thành K phần và huấn luyện mô hình K lần trên các tập con khác nhau, giúp tận dụng tối đa toàn bộ dữ liệu để vừa huấn luyện vừa kiểm tra mô hình. Nhờ đó, K-Fold giúp giảm thiểu nguy cơ overfitting và hạn chế underfitting, đồng thời cung cấp kết quả đánh giá ổn định và đáng tin cậy hơn so với các phương pháp chia dữ liệu thông thường.

1. **Train**

Data:<https://www.kaggle.com/datasets/ptlocnguyen/brain-tumor-od-datasets-k-fold-cross-validation/data>

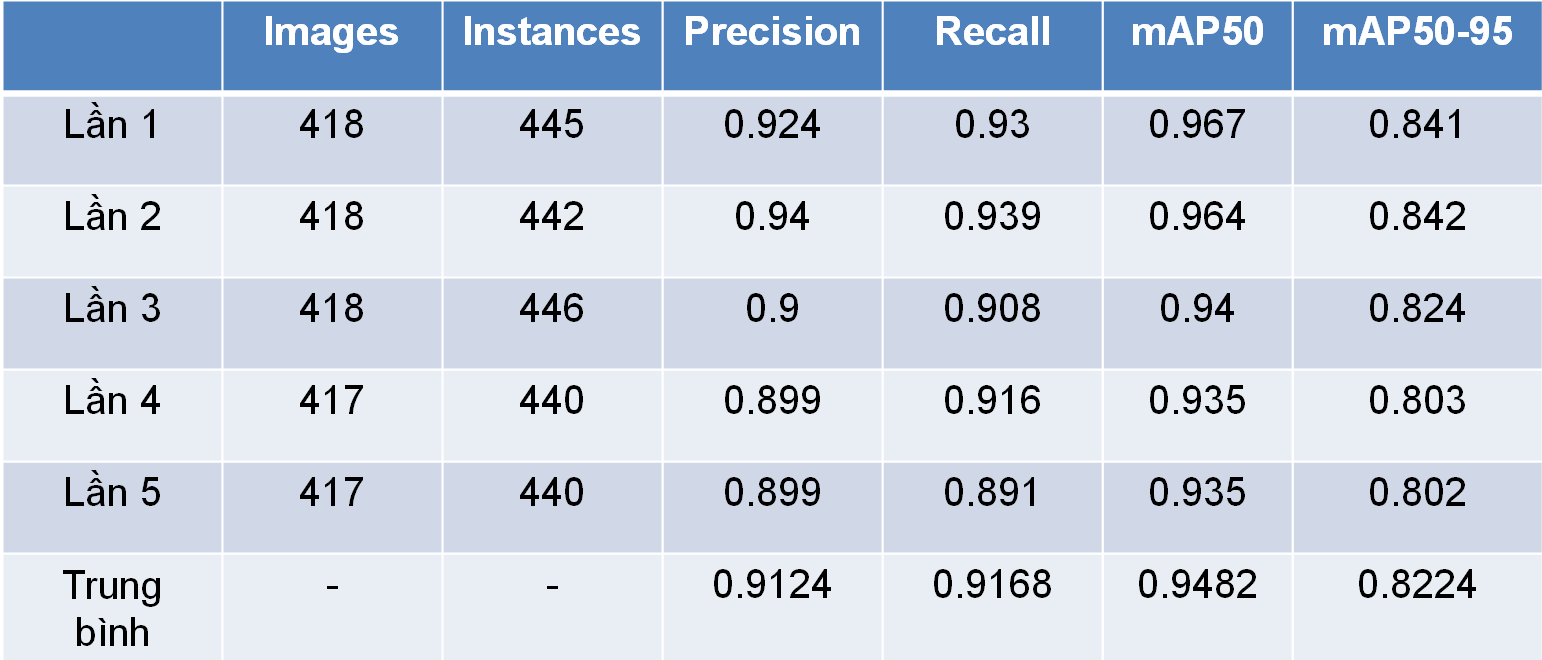
Đây là bộ dữ liệu được nhóm chia thành 5 bộ train-val (80-20) để tiến hành ứng dụng vào train K-Fold Cross Validation

Model: yolo11l.pt

Thông số train epochs=300, batch=32, imgsz=640, device=[0, 1]

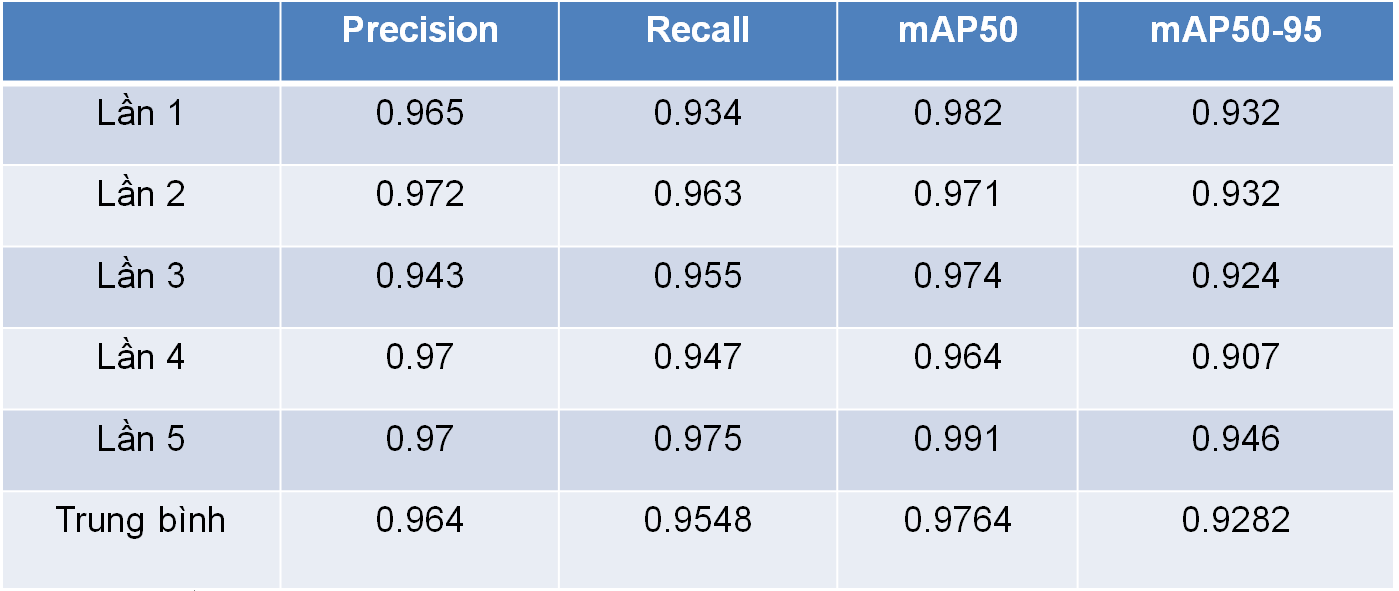
1. **Kết quả**

Sau khi thực hiện K-Fold Cross Validation với k=5 nhận được kết quả khá khả quan với các thông số.



Hình 42 Kết quả sau Train

Kết quả dự đoán trên cùng 1 tập test với images: 112, Instances: 117



Hình 43 Kết quả sau test trên cùng tập dữ liệu

Các link code:

Lần 1: <https://www.kaggle.com/code/ptlocnguyen/k-fold-brain-tumor-od-train-split1>

Lần 2: <https://www.kaggle.com/code/phantnlcnguyn/k-fold-brain-tumor-od-train-split2>

Lần 3: <https://www.kaggle.com/code/thudo25/k-fold-brain-tumor-od-train-split3>

Lần 4: <https://www.kaggle.com/code/ptlocnguyen/k-fold-brain-tumor-od-train-split4>

Lần 5: <https://www.kaggle.com/code/phantnlcnguyn/k-fold-brain-tumor-od-train-split5>

Test: <https://www.kaggle.com/code/thudo25/da2-predict-best-pt>

# Tổng kết

Việc áp dụng kỹ thuật xử lý ảnh và các mô hình học sâu vào phát hiện u não đã mang lại kết quả khả quan với độ chính xác cao, tối ưu hóa và cải thiện hiệu suất mô hình một cách đáng kể. Các chỉ số như Precision, Recall, và mAP đều đạt ngưỡng khá cao, dao động từ 0.8 đến 0.98, cho thấy mô hình có khả năng nhận diện và phân loại u não chính xác và hiệu quả trong bối cảnh dữ liệu hiện tại. Điều này chứng tỏ khả năng vượt trội của các phương pháp học sâu trong việc giải quyết các bài toán phức tạp trong y tế, đặc biệt là trong việc hỗ trợ chẩn đoán u não tự động.

Kết quả này không chỉ khẳng định tính hiệu quả của công nghệ hiện đại trong lĩnh vực y tế mà còn mở ra nhiều cơ hội nghiên cứu và ứng dụng rộng rãi trong các phương pháp chẩn đoán tự động, giúp giảm thiểu sai sót và tăng tính khách quan trong việc phát hiện bệnh. Tuy nhiên, để nâng cao hiệu suất của mô hình và tăng tính đa dạng, việc mở rộng nghiên cứu và thử nghiệm trên nhiều tập dữ liệu khác nhau là rất cần thiết. Điều này sẽ giúp mô hình có thể áp dụng được trong các trường hợp phức tạp và đa dạng hơn, đồng thời nâng cao khả năng nhận diện u não từ các phương diện khác nhau.

Ngoài ra, việc tăng cường số lượng dữ liệu đào tạo và sử dụng các phương pháp tăng cường dữ liệu có thể giúp mô hình học được các đặc trưng phong phú hơn, cải thiện khả năng phát hiện u não chính xác hơn trong các tình huống thực tế. Việc tiếp tục tối ưu hóa các mô hình học sâu, kết hợp với các kỹ thuật xử lý ảnh tiên tiến, sẽ góp phần mở ra những cơ hội lớn trong việc phát triển các công cụ hỗ trợ chẩn đoán y tế tự động, giúp các bác sĩ và các chuyên gia y tế có thêm một công cụ mạnh mẽ để nâng cao chất lượng chăm sóc sức khỏe.

# Nguồn tham khảo

<https://www.youtube.com/playlist?list=PLYSUjrPFpR_nI3OOqkDlz7p4aew0bQ1dr>

<https://www.youtube.com/watch?v=zzmecn9bTtw>

<https://radiopaedia.org/articles/mri-sequences-overview>

[How to Read MRI Results: Interpreting Your Report & Terminology](https://www.pockethealth.com/2023/02/08/whats-in-my-mri-report-decoding-common-terminology/)

<https://www.datacamp.com/blog/yolo-object-detection-explained>

<https://docs.ultralytics.com>