Computer Vision Final Group Project

AI CUP 2025 參賽：電腦斷層心臟肌肉影像分割競賽--主動脈瓣物件偵測

Group Members

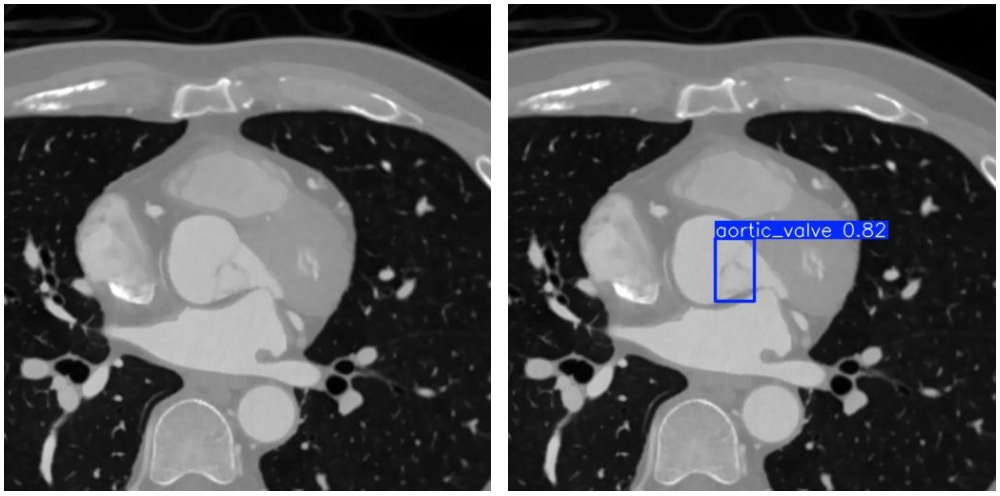
B113040047 王宸澤

B113040056 洪理川

Summarize:

本競賽目標是判別電腦斷層影像中，心臟主動脈瓣的 bounding box，減少人工標註的費時。我們在競賽中實驗兩大方向的模型修改，第一種是「超參數的設定、資料前處理以及訓練方式的調整」，包含以不同 YOLO 模型、data augmentation、調整訓練超參數、訓練及驗證資料的分配與前處理、K-fold ensemble，來盡可能提高物件辨識的準確率；第二種是「模型架構的調整」，我們將 YOLOv12 的 backbone 修改成 ConvNeXt，並將 neck 替換成 BiFPN+CBAM。

最終我們的參賽成績於 536 個報名的隊伍中，以 mAP:0.943878 的成績排名於第 122 名，並且具有最佳成績的模型是使用 YOLOv12m 配合 k-fold ensemble 的模型，在預測結果的 mAP50 達到 0.958524137820230。



1. Introduction

1.1. 競賽介紹與問題描述

1.1.1. 競賽介紹

參加 AI CUP 2025 秋季賽-電腦斷層心臟肌肉影像分割競賽 II-主動脈瓣物件偵測，此競賽目標是藉由人工標記完成的資料，訓練出物件偵測模型，來找出電腦斷層圖像中的心臟主動脈瓣所在位置的 bounding box。主動脈瓣可以幫助心臟正常的將血液送往全身，但若病患發生主動脈瓣狹窄的問題，就會需要進行主動脈瓣置換手術來解決，而此主動脈瓣偵測模型能有效減少過去須以人工標記瓣膜位置的人力成本，改以模型進行定位。

1.1.2. 重要競賽規則

(1) 一天僅能繳交三次 testing data 的預測結果。

(2) 主辦方提供的 datasets 中，具有 training image 和 testing image，而 testing image 則還被分成 public data 與 private data(參賽者不知道哪些 data 是 public 或 private)。

(3) 排名方式：

a. 以繳交的 test data 結果之 mAP50 成績來排名。

b. Public leaderboard: 以繳交的 test data 的 public data 來排名，並且在競賽的全程都能查詢。

c. Private leaderboard: 以繳交的 test data 的 private data 來排名，並且僅能在競賽結束後看到結果。而這個排名會是競賽的最終排名。

1.1.3. 競賽成績計算方式

本競賽的目標是預測主動脈瓣的類別、信心分數、以及其 bounding box 左上角與右下角的座標 (x,y)。模型表現主要以 AP@0.5（Average Precision at IoU ≥ 0.5）作為評估指標，其計算方式如下：

$$AP = \sum_{i=1}^{n} (R_i - R_{i-1}) \times P_i$$

其中：

R_i: 第 i 個 recall 值

P_i: 經過 smoothing 的 precision 值

Precision 與 recall 的定義如下：

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}$$

最後，AP 可視為 precision-recall 曲線下的面積（Area under the Precision-Recall curve），也就是將不同 recall 對應的 precision 點連成曲線後所形成的面積大小。

1.2. Related work

ConvNeXt[1]源自於 Swin transformer 在影像辨識領域上的成功。由於 Swin transformer 將許多 CNN 的概念引入 ViT 中，讓 ConvNeXt 的作者猜想若將 transformer 的技術也一步步地引入當前時最具代表性的 CNN 模型: ResNet-50/200，能否僅使用 ConvNet 而不使用 Transformer 就達到相同的準確程度。因此 Z Liu at el.將 Swin transformer 中，在傳統 ConvNet 不常使用的以下架構套入至 ResNet-50/200 中，最後讓模型的成效在 ImageNet Top1 Acc(%)從 78.8%提升至 82.0%，超過 Swin transformer 的 81.3%。

訓練方式	改善 optimizer、使用更多 epoch、使用各種 data augmentation 技術。
模型設計	ResNet 中，第二至第五 stage 的 blocks 的重複次數，從 3:4:6:3 改為 3:3:9:3。並導入 depthwise convolution。
Kernel size	從自 VGG 後常用的 3x3 convolution 改為 7x7 convolution
Block 設計	導入 inverted bottleneck 於 ResNet Block 中。
Activation function	ReLU 換成 transformer 使用的 GeLU，並減少 activation functions 數量
Normalization 方式	Batch norm 換成 transformer 常用的 Layer norm

BiFPN（Bidirectional Feature Pyramid Network）最早由 EfficientDet[3] 論文提出，作為其特徵融合網路的一部分。BiFPN 可視為 FPN 與 PANet 的延伸：FPN 透過 top-down pathway 融合多尺度特徵，而 PANet 則在 FPN 基礎上加入 bottom-up pathway，使特徵能雙向傳遞。由於不同尺度特徵解析度與資訊量不同，若直接相加未必能反映其重要性，因此 BiFPN 在融合時加入可學習權重，讓網路自動學習各輸入特徵的貢獻程度，以在準確率與效率之間取得更好的取捨。此外，BiFPN 也透過移除單一輸入節點、加入同層級連線，以及重複堆疊多層等方式，進一步強化跨尺度特徵融合能力。。

CBAM[4]則可視為 Squeeze-and-Excitation Network(SE)[5]的延伸，主要由兩個注意力模組組成：Channel Attention 與 Spatial Attention。相較於 SE 僅透過 global average pooling 進行通道權重建模，CBAM 在計算注意力時同時引入 average pooling 與 max pooling，使模型能從「平均特徵」與「最顯著特徵」兩個角度補強資訊表達，進而提升注意力機制對關鍵區域與關鍵通道的辨識能力。

為驗證 CBAM 的有效性，原論文在多個基準資料集上進行實驗（ImageNet-1K、MS COCO、VOC 2007），結果顯示 CBAM 能在多種 backbone 上穩定優於 baseline。以物件偵測為例，當使用 ResNet-50 作為 backbone 時，相較於原本 ResNet-50，加入 CBAM 後 mAP@0.5 可由 46.2 提升至 48.2。

1.3. 研究方法簡介

我們將使用 YOLOv12 模型進行 bounding box 的預測，並且主要分成以下兩個部

分實作：

1.3.1. 超參數的設定、資料前處理以及訓練方式的調整

Ultralytics 釋出的 YOLOv12 能透過調整提供的參數，來改變模型訓練的事項，對於同個模型也提供多種不同參數量的模型，在這個階段，我們將實驗不同參數量的模型，以及調整各種參數後，對於結果的影響（含 patience 與部份能進行 data augmentation 的參數）。

除此之外，在將資料送入模型前，也嘗試對其進行前處理，包含先額外訓練模型，對 test data 進行 classify，找出含有主動脈瓣的圖片，才拿這些圖片進行主動脈瓣位置的預測；以及對 training 與 testing 圖片進行 histogram equalization，讓電腦斷層圖像中的主動脈瓣能更加清楚，才將其送入模型進行訓練及預測。

而在訓練上，我們也採用 K-fold validation，讓所有資料都有機會被訓練到，再將各個訓練所得的模型進行"essembling"，結合各模型的優勢。

1.3.2. 模型架構的調整

由於過去研究指出，部分較新式的卷積骨幹在醫療影像任務（例如 CT 影像的分割與辨識）上常能帶來較好的效果，像 DiagNeXt 與 MedNeXt 等研究也顯示 ConvNeXt 類架構在醫療影像分析中具備不錯的特徵表達能力[6]-[7]。因此在本研究中，我們嘗試將 YOLOv12 原本的 backbone 改為 ConvNeXt，期望模型能更有效學到主動脈瓣相關的特徵。

在 neck 的設計上，ConvNeXt 本身會輸出多尺度的特徵圖，因此能較自然地銜接特徵金字塔類的方法來做多尺度融合。相較之下，若使用標準 ViT 作為 backbone，通常只會產生單一尺度特徵，還需要額外透過上採樣與卷積去建構多尺度特徵，整體架構會更複雜。

基於上述原因，我們將 YOLOv12 原本由 ELAN block 組成的 neck 移除，並在 neck 部分改採用 BiFPN 來做多尺度特徵融合，利用其加權融合的特性，讓模型能自動學習不同尺度特徵的重要性。此外，由於醫療影像常見目標區域偏小、對比不明顯且背景干擾較多，我們也在 neck 後端加入 CBAM，引導模型更聚焦在可能的主動脈瓣位置，以期進一步提升偵測準確率與 mAP@0.5 表現。

2. Details of the approach

本次競賽提供一個 baseline 程式，其使用 YOLO12n 模型來對資料集的圖片進行訓練，再以訓練流程中的最佳模型進行預測主動脈瓣的 bounding box 位置，而在執行後，他在 public leaderboard 的 mAP50 能達到 0.91782976 的表現，如下圖所示。

merged.txt baseline code YOLOv12 1.01 version 上傳成員 洪 理川	2025-11-12 05:24:06	0.91782976	0.8787836795480679	Scoring success.

而我們的實驗將會圍繞著此 baseline 程式，修改其功能，以及加入於 introduction 講述的方式，實驗加入前後的 mAP50 表現能否提升。

2.1. 超參數的設定、資料前處理以及訓練方式的調整

2.1.1. Augmentation

我們嘗試對每張圖片都進行以下 augmentation 操作：

rotation	每張圖片用於訓練前，旋轉-5~5 度或-10~10 度
fliplr	每張圖片用於訓練前，有 10% 的機率會使其進行水平方向的 flip
scale	每張圖片用於訓練前，尺寸縮放 0.95~1.05 或 0.9~1.1 倍
translate	每張圖片用於訓練前，進行平移，平移幅度最高為圖片尺寸的 5%

然而將 testing data 的預測結果顯示，進行 augmentation 反而使 mAP50 降低至 0.85067698 到 0.90168443 之間，並沒有如同預期的，能夠有效提升模型預測的準確率。造成此結果推測是因為 testing data 並非如同一般影像辨識模型，是要預測變化非常大的圖片（例如給了非常多完全不同的照片，要模型預測狗的 bounding box），本次競賽提供的資料是某位患者在一次治療中，連續一段時間內的電腦斷層影像，所以事實上大部分的資料都長得很像，所以用 augmentation 讓模型學習到各種多樣的狀況，反而不益於它於此競賽有更好的表現。因此**在後續的實驗，將不再採用**這種 **augmentation**。

2.1.2. 使用不同版本的 YOLOv12 進行訓練

論文「YOLOv12: Attention-Centric Real-Time Object Detectors」中，Y. Tian et al. [2] 提出一種不同於 Ultralytics 釋出的 YOLOv12 版本，作者表示此版本改善了原先效率不佳與訓練不穩定的問題。然而在實驗後發現，當我們使用兩種版本且規模相同的 YOLOv12n（nano）模型進行訓練時，論文版本的預測準確率反而不如 Ultralytics 版本。

進一步分析兩者架構後，我們發現兩個版本在設計理念上相近，但論文版本使用了較多「重複堆疊」的結構，使整體網路深度增加約 80%～90%。以 nano 模型為例，Ultralytics 版本約為 272 layers，而論文版本約為 497 layers。推測在本競賽資料較少的條件下，過深的模型反而較不利於穩定訓練與提升表現。

因此，**我們在後續實驗皆採用 Ultralytics 原版 YOLOv12 作為主要的影像辨識模型。**

2.1.3. 使用不同參數量的 YOLOv12 進行訓練

Ultralytics 的 YOLOv12 提供了五種參數量的模型，由小到大分別是 yolov12n、yolov12s、yolov12m、yolov12l、yolov12x，然而礙於硬體的限制，我們只測試 yolov12n、yolov12s、yolov12m 三種模型。發現參數量位於中間的 yolov12m 有最佳的辨識準確率，其 mAP50 數值在 public leaderboard 達到 0.94263301。

推測是因為競賽提供的主動脈瓣資料集僅有 16863 張圖片，若使用參數量最大的模型，會因為樣本數不夠，使模型產生 underfitting 的狀況，但使用參數量較小的模型，又因為彈性不夠，而造成 overfitting，使準確率不高。因此**規模中等的 yolov12m 較適合本競賽提供的資料集，後續實驗使用 YOLOv12 模型，也會以 yolov12m 為主**。

2.1.4. Train/Validation 資料比例調整

競賽方提供的 dataset 包含了 50 位病人的電腦斷層掃描圖，每位病人有 250~450 張影像，總共有 16863 張圖片。而我們選用了三種比例 train 與 validation 資料的比例，分別是 30:20、40:10、49:1，目標是觀察提供不同量的 training data，是否能讓模型有不同的表現，這個比例是以病人的數量來區分，例如 30:20 將會拿 30 位病人的資料作為 training data，剩下 20 位病人的資料則作為 validation data。

最後**在 public leaderboard 的成績是 40:10 的比例有最好的表現，但 49:1 則在 private leaderboard 有較好的表現**，兩者的 mAP50 都能達到 0.94，而 30:20 的比例在 mAP50 上則相較這兩種比例有 0.01~0.02 的減少。由於我們的資料集較小，所以推測造成此結果的原因是由於能使訓練的資料更多，讓模型有較好的成效。

2.1.5. 讓模型學習到「沒有主動脈瓣」的圖片的特徵

由於主辦方所提供的 baseline 程式設定為，訓練時只使用有 label 的影像（即是僅使用有主動脈瓣的圖片做訓練），因此在此實驗中，我們嘗試不包含主動脈瓣的影像也納入訓練中，目標減少誤判的可能。而在資料集中，約有八成的圖片是不含主動脈瓣的。

我們進行三組實驗，分別是：

(1) 將所有影像都拿來訓練（含具有主動脈瓣和不含主動脈瓣的影像）
(2) 依照具有主動脈瓣的影像數量，取此數量的 20%的數量的不含主動脈瓣影像和具有主動脈瓣的影像一起訓練
(3) 最後一種是按照 baseline 程式，只取具有主動脈瓣的影像進行訓練。

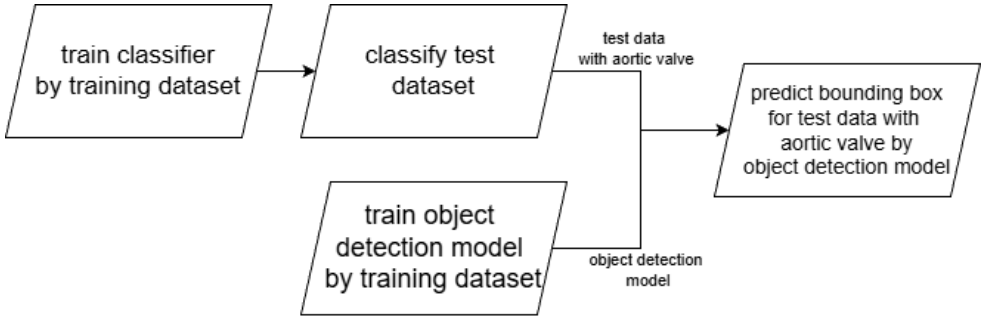
從結果發現，不使用任何不含主動脈瓣影像的訓練資料仍然有最高的準確率，並且將所有圖片一起訓練的版本有最差的表現。推測原因在於含有主動脈瓣的影像資料量不足，若加入不含主動脈瓣的影像來一起訓練，反而讓模型沒辦法準確學習到具有

主動脈瓣的影像長甚麼樣子。

2.1.6. 另行訓練 classifier 來先行分類 test data

這項實驗目標想要先以 classifier 找出 test data 中，具有主動脈瓣的影像，再讓物件偵測模型找出確切的主動脈瓣位置，減少物件偵測模型誤判的機率。訓練時，我們先將原本競賽提供的 training data 分類成 positive（存在主動脈瓣的影像），以及 negative（不存在主動脈瓣的影像），做法是透過判別某張影像是否有競賽方提供的 label 來決定，若影像有對應的 label，就代表影像內有有主動脈瓣，則會將該影像分類為 positive。

接著會將分類完的 training data 讓 YOLOv11n-cls 進行訓練，再把 test data 送入此模型，來判別影像內是否具有主動脈瓣。最後使用先前擁有最好表現的物件偵測模型（採用 train / validation 比例為 49:1 訓練所得的模型）對被分類為 positive 的 test data 進一步做物件位置判別。其架構如下圖所示。



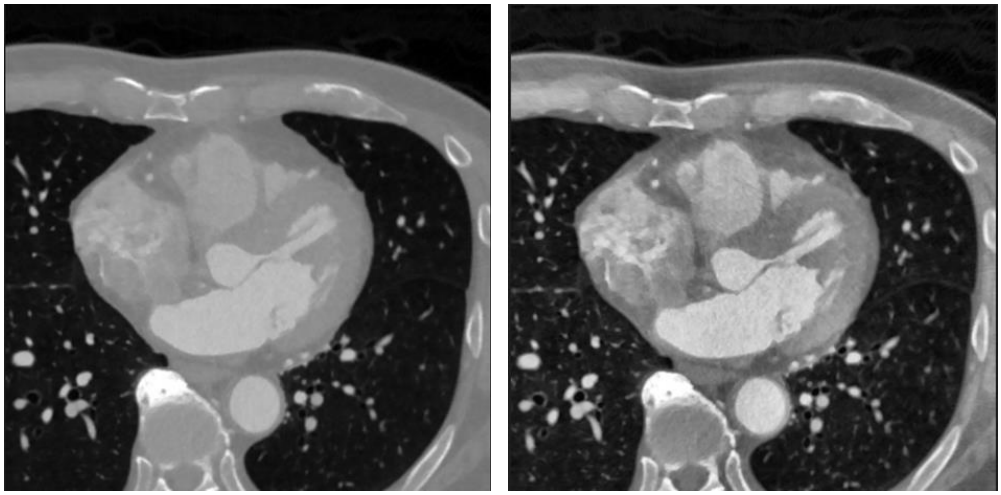
不過，實驗的結果顯現，對資料 classify 後，有時能提高準確率，有時會降低準確率，因此在後續做的實驗，部分會再嘗試是否要進行 classify 的操作。

2.1.7. Histogram equalization

Histogram equalization 是一種提升影像對比的方法，透過重新分配像素強度值，使影像中的重要結構更明顯。我們嘗試將 histogram equalization 作為影像前處理與 data augmentation，觀察其對模型表現的影響。

2.1.7.1. 作為資料前處理（Data preprocessing）

作為前處理時，我們會先將原始影像轉換成 histogram equalization 後的影像，再送入模型訓練。我們使用 OpenCV 的 CLAHE（Contrast Limited Adaptive Histogram Equalization）來提升 CT 影像的局部對比，讓解剖結構更清楚。CLAHE 的作法是將影像切成多個小區塊（tiles）分別進行強度分布調整，同時透過 clip limit 限制對比過度放大，以避免雜訊被放大。在本次實作中，我們設定 clip limit = 2.0，且 tile grid size = 8×8，前後對比效果如下方所示，其中左圖是處理前，右圖是處理後的結果。



然而，該方法使 public leaderboard 準確率由 0.93713821 降至 0.90168443，因此我們改以 data augmentation 方式進一步測試。

2.1.7.2. 作為資料增強（Data augmentation）

在 data augmentation 的設定下，我們將 CLAHE 處理後的影像與原始影像 一併加入訓練資料，使模型同時學習原始影像與對比增強後影像的特徵（CLAHE 的效果如 2.1.7.1 所示）。

但實驗結果顯示，採用此作法後，模型在 training 的 validation 上最佳 mAP 僅有 0.9222，明顯低於 baseline 在 validation 可達 0.969 的表現，**因此我們最終決定在本研究中完全移除 histogram equalization（CLAHE） 相關的前處理與 augmentation**。

2.1.8. K-fold validation 與模型的 ensembling

在嘗試多種方法後，我們發現模型的準確率表現多停留在 0.94 左右，因此我們開始思考是否能透過「整合多個表現普通的模型」，來得到一個更穩定、效果更好的模型。基於此想法，我們採用 K-fold validation 的方式進行訓練，並將不同 fold 訓練出的模型做 ensemble。

由於我們在訓練時採用 train/validation = 40:10 的設定，因此我們選擇使用 5-fold 來切分與訓練，最後會得到 5 個模型。接著在預測階段，我們將這 5 個模型的預測結果整合，並使用 Weighted Box Fusion（WBF）將多個模型的 bounding box 融合成最終的輸出。

2.1.8.1. Weighted box fusion

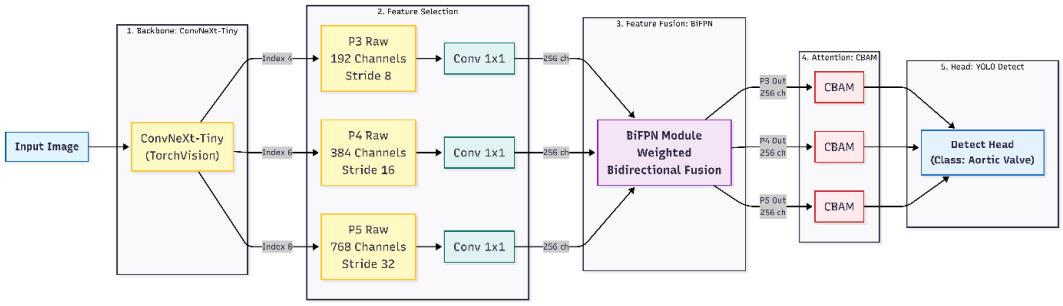
Weighted Box Fusion [8]是一種將各個模型預測的 bounding box 合併的方法，他的概念是：針對不同模型預測出的位置相近，但 confidence 不同的 bounding box，根據各個 bounding box 的 confidence 提供一個 weight，再將它們依照 weight 合併成一個最

終的 bounding box。WBF 會先針對同一張影像、同一類別的預測框，依據 IoU 是否大於某個門檻（例如 IoU ≥ 0.5）進行分組。若某個 bounding box 與群組中的代表框之 IoU 超過門檻，則會被視為同一組，代表它們很可能是在預測同一個主動脈瓣位置。

分組完成後，WBF 會將同一組的 bounding box 進行融合，其 bounding box 座標會以加權平均的方式計算，權重通常與 confidence 成正比，使高 confidence 的預測對最終結果具有較大影響。因此，融合後的 bounding box 位置會更接近高 confidence 的預測結果，從而提升定位穩定性並降低單一模型預測偏移所造成的誤差。

2.2. 模型架構的調整

在此部分，我們將會嘗試將 YOLOv12 的 backbone 替換成 ConvNeXt，再將 YOLOv12 的 neck 替換成 BiFPN + CBAM 的架構，架構圖如下圖所示：



2.2.1. 將 YOLOv12 的 backbone 改為 ConvNeXt

替換的做法主要是修改 yolo12.yaml 檔案，該檔案原本標示了 yolov12 的架構，若使用者在呼叫模型時，可以直接呼叫 yolo12.yaml，就能獲取未經過預訓練的 yolo12 模型。

我們將 yolo12.yaml 中，屬於 backbone 的部分替換成 TorchVision 提供的 ConvNeXt 模型，並且設定為使用預訓練的權重，讓我們的模型不必再額外使用大規模的資料集，如 COCO dataset，來進行預訓練。而除了導入 ConvNeXt 作為我們的 backbone 外，也仿照原本的架構，修改 yolo12.yaml 中的 neck，作為加入 BiFPN + CBAM neck 之前，測試 ConvNeXt 所使用的 neck。我們會從 ConvNeXt backbone 的不同 stage 輸出之 feature map，分別作為 P3、P4、P5 feature，以支援小到大尺度物件的偵測，再將具有最強語意的 P5 層進行 FPN 向上採樣，接著再做 PAN 向下採樣，讓高層的語意和底層的位置資訊結合，最後將三層的資訊彙整，送往 output layer，來取得物件偵測的結果。

我們將修改後的 YOLOv12 架構以主辦方提供的主動脈瓣資料集再進行 fine-tuned，並且訓練時，導入 20%的不含主動脈瓣的影像，最後在 public leaderboard 上得到不錯的結果，public leaderboard 的 mAP 達到 0.94371780，相比直接使用此資料進行訓練的原版 YOLOv12 模型，有小幅的提升。

2.2.2. 將 YOLOv12 的 neck 改為 BiFPN+CBAM

為了將 YOLOv12 的 neck 改為 BiFPN + CBAM，我們需要修改 YOLOv12 的.yaml 架構設定。由於 Ultralytics 並未提供 BiFPN 模組，因此我們需自行撰寫客製化的 BiFPN 以配合自行設計的模型架構。我們使用的 BiFPN 相較於 EfficientDet 論文中的 BiFPN 更為簡化：EfficientDet 的版本具有較密集的 skip connection、每個節點融合的輸入更多（多尺度如 P3～P7），特徵重用能力更強；每個融合節點的輸入來源較少（多為兩路融合），且僅採用 P3～P5 三個尺度，因此整體特徵重用與跨尺度連接較簡化。實作上，我們將 BiFPN 模組程式碼置於 ultralytics/nn/modules/bifpn.py，並在 ultralytics/nn/modules/__init__.py 中註冊該模組，讓.yaml 可以正確呼叫 BiFPN。完成後即可在模型設定檔中直接使用 BiFPN 作為 neck 的特徵融合模組。

另外，由於 ConvNeXt 輸出的多尺度特徵（例如 P3、P4、P5）通道數不一定相同，為了能順利送入 BiFPN 進行融合，我們先透過 1×1 convolution 將 P3、P4、P5 的通道數統一調整為 256 channels，再將這三個尺度的特徵輸入 BiFPN 做多尺度融合，確保融合過程的維度一致。

至於 CBAM 的部分，由於 Ultralytics 已在模組中提供 CBAM block，因此我們不需要另外實作，只需在 .yaml 中直接使用即可。我們將 BiFPN 的三個輸出特徵接上 CBAM 進行注意力強化，最後再傳入 detection head 進行主動脈瓣的 bounding box 預測。

然而在實際結果上，修改成 BiFPN + CBAM 後的模型表現仍與原本 YOLOv12m 有一定差距，public leaderboard 的成績約為 0.910，未能如預期提升整體 mAP50 表現。

3. Results

本部分將會說明於第二部分提出的方法，其完整的實驗結果數據。其中，下列呈現的數據都是四捨五入制小數點後第四位的 mAP50 成績。

3.1. 超參數的設定、資料前處理以及訓練方式的調整

3.1.1. Augmentation

Augmentation 將套用 Rotation, Translation, Horizontal flip, Scaling 四種 Augmentation 方式，其中的成績如下表所示：

Model	Augmentation	Public leaderboard	Private Leaderboard
yolov12n	Yes	0.9017	0.8795
Yolov12n	No	**0.9371**	**0.9313**

發現結果不使用 Augmentation 的結果比較好，因此後續的實驗將不採用 Augmentation。

3.1.2. 使用不同版本的 YOLOv12 進行訓練

我們嘗試使用 Ultralytics 所提供的 YOLOv12m，以及原論文作者所提供的 YOLOv12m，它們的成績如下表所示：

Model	Public leaderboard	Private Leaderboard
yolov12m (Ultralytics)	**0.9426**	**0.9354**
yolov12m (paper)	0.9247	0.9264

3.1.3. 使用不同參數量的 YOLOv12 進行訓練

對於 Ultralytics 提供的五種不同尺寸的 YOLOv12 模型，我們實驗了其中的三種尺寸：YOLOv12n、YOLOv12s、YOLOv12m 的模型，其中由於 YOLOv12s 在訓練時，validation 的結果較差，因此我們沒將它上傳到競賽網站來觀看它的成績。下表將呈現 YOLOv12n 與 YOLOv12m 的成績：

Model	Public leaderboard	Private Leaderboard
yolov12n	0.9371	0.9313
yolov12m	**0.9403**	**0.9399**

可以發現，yolov12m 的表現是個參數量的模型中，表現最佳的，因此我們在後續的實驗中，若自行設計的實驗想與 YOLOv12 原版進行準確率的比較，則皆使用 yolov12m 作為實驗的對照組。

3.1.4. Train/Validation 資料比例調整

我們以 YOLOv12m 模型來實驗三種的 Train/Validation 資料比例，觀察提供更多資料給模型訓練，能否有較佳的表現。它們的成績如下表所示：

Model	Train/Val	Public leaderboard	Private Leaderboard
yolov12m	49/1	0.9395	**0.9439**
yolov12m	40/10	**0.9426**	0.9354
yolov12m	30/20	0.9421	0.9437

3.1.5. 讓模型學習到「沒有主動脈瓣」的圖片的特徵

我們進行了兩組實驗，分別是「使用完整 datasets（含所有沒有主動脈瓣的圖片）」以及「不含主動脈瓣的圖片數量僅占含主動脈瓣圖片數量的 20%」兩種訓練方式。它們的成績如下表所示：

Model	Training datasets	Public leaderboard	Private Leaderboard
yolov12m	all images	0.8233	0.7412
yolov12m	20% images without aortic valve	0.9252	0.9238
yolov12m	only images with aortic valve	**0.9403**	**0.9399**

3.1.6. 另行訓練 classifier 來先行分類 test data

此實驗先以 training datasets 訓練 YOLOv11n-cls，再將 test data 送入此模型，分類出圖片是否具有心臟主動脈瓣，接著使用 yolov12m 物件偵測模型辨識 classifier 判別具有心臟主動脈瓣的 test data。另外，此實驗使用的是具有最佳 Private leaderboard 成績的 yolov12m、train : val 比例為 49:1 的模型，選用此比例，是因為本實驗是在競賽截止後進行的，因此我們選用 private leaderboard 成績最高的模型來進行實驗。它們的成績如下表所示：

Model	Classify	Public leaderboard	Private Leaderboard
yolov12m	Yes	0.9355	**0.9341**
yolov12m	No	**0.9371**	0.9313

從結果可以發現，兩個模型在 Public leaderboard 與 Private leaderboard 各有優勢。

3.1.7. Histogram equalization

我們嘗試將 Histogram Equalization 同時作為 data preprocessing 與 data augmentation，其結果整理如表所示：

Model	Histogram Equalization	Validation	Public leaderboard	Private Leaderboard
yolov12m	Augmentation	0.9518	0.9017	0.8795
yolov12m	Data preprocessing	0.9222	-	-
yolov12m	No	**0.9690**	**0.94031**	**0.9399**

由表可以發現，無論是作為 augmentation 或前處理，表現皆低於 yolo12m。其中 data preprocessing 的 validation 僅 0.9222，明顯偏低，因此我們並未將此版本提交至 leaderboard；最終也決定不採用 histogram equalization 於本研究流程中。

3.1.8. K-fold Ensemble Model

在模型表現提升受限的情況下，我們改採 K-fold ensemble 搭配 Weighted Box Fusion 進行模型整合。本次使用 K=5，將多個 fold 模型的預測框進行融合，以提高最終預測的穩定性與準確率。結果如下表所示：

Model	K-Fold Ensemble	Public leaderboard	Private Leaderboard
yolov12m	Yes	**0.9446**	**0.9585**
yolov12m	No	0.9403	0.9399
Self-designed model	Yes	0.9344	0.9361
Self-designed model	No	0.9105	0.9230

從結果可觀察到，加入 K-fold ensemble 後，不論是 yolov12m 或自建模型皆有明顯提升，其中 yolov12m + K-fold ensemble 的 private leaderboard 可達 0.9585，是本次實驗中準確率最高。

3.2. 模型架構的調整

3.2.1. 將 YOLOv12 的 backbone 改為 ConvNeXt

此實驗探討僅將 backbone 換成 ConvNeXt，而 neck 與 head 的架構仍仿照或直接使用 YOLOv12 原本的架構進行。其中，我們在此實驗使用的資料集包含了 20%的無主動脈瓣影像，並且 train 與 validation 資料的比例為 40:10。實驗的結果如下表所示：

Model	Public leaderboard	Private Leaderboard
ConvNeXt backbone + YOLOv12 neck and head	**0.9437**	0.9363
yolov12m	0.9403	**0.9399**

3.2.2. 完整套用自行設計的架構：ConvNeXt + BiFPN + CBAM

此實驗將使用自行設計的架構。另外，因為進行此實驗時，已經確定使用 k-fold ensemble 能使預測準確率提高許多，因此以下實驗結果皆套用 k-fold ensemble。此外，由於從 3.1.6.的結果可見，使用 classify 有可能使預測準確率提升，所以此實驗也同時將使否使用 classify 作為另一個變因。此實驗的成績如下表所示：

Model	Classify	Public leaderboard	Private Leaderboard
Self-designed model	Yes	0.9514	0.9405
Self-designed model	No	0.9345	0.9361
yolov12m	Yes	**0.9514**	0.9513
yolov12m	No	0.9446	**0.9585**

從結果可以發現，自行設計的模型架構結果不如預期，仍無法超越 yolov12m 的效能。但是若將 k-fold ensemble 在進行 ensemble 前的五個模型準確率做比較，可以看見成績如下（僅放上 private leaderboard 的結果）：

Model	1st fold	2nd fold	3rd fold	4th fold	5th fold
Self-designed model with classify	0.9134	0.9058	**0.9256**	0.9122	0.9156
yolov12m with classify	0.9324	0.9357	0.9088	0.9310	0.9221

可以發現，自行設計的模型在 3rd fold 的表現能贏過 yolov12m。

4. Discussion and conclusions

上述的實驗，如 classify、K-fold ensemble 以及完整的自行設計架構(3.2.2.)，皆是在競賽結束後，請競賽方幫我們測試結果，因此不能納入競賽的排名當中。而綜合上

述實驗的結果，我們整理出以下幾個，分別在 private leaderboard 與 public leaderboard 表現最佳的模型以及其排名：

Model	train/val ratio	Public leaderboard	Private leaderboard	Note	rank
ConvNeXt backbone + YOLOv12 neck and head	40/10	**0.9437**	0.9363	Use 20% unlabeled image in training datasets	200/536
yolov12m	49/1	0.9395	**0.9439**		122/536

若將競賽結束後，才繳交的模型成果也納入考量，則具有最佳成績的模型為：

Model	Classify	Public leaderboard	Private Leaderboard
yolov12m	Yes	**0.9514**	0.9513
yolov12m	No	0.9446	**0.9585**

從結果可以發現，對於此競賽所需辨識的電腦斷層醫療影像，事實上使用 yolov12 模型就能得到優秀的結果，而在使用 k-fold ensemble 來整合多個模型的結果後，能夠得到更好的成效。而訓練方式的調整則會影響結果，但本次競賽我們選用的調整都沒有得到太好的成效；架構的調整也同樣沒有對結果有太大的影響。未來的若有機會再參與相關競賽，或許可以先從觀看其他專注於「電腦斷層影像物件偵測」的相關研究，再從這些研究中思考該如何調整架構與實驗方式。

5. Statement of individual contribution

王宸澤 B113040047: 50%

洪理川 B113040056: 50%

6. Reference

[1] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in CVPR, 2022.

[2] Y. Tian, Q. Ye, and D. Doermann, “YOLOv12: Attention-Centric Real-Time Object Detectors,” arXiv preprint arXiv:2502.12524, Feb. 2025.

[3] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” in CVPR, 2020.

[4] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” in ECCV, 2018.

[5] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in CVPR, 2018, pp. 7132–7141.

[6] H. Tekin, Ş. Kılıç, and Y. Doğan, “DiagNeXt: A Two-Stage Attention-Guided ConvNeXt Framework for Kidney Pathology Segmentation and Classification,” J. Imaging, vol. 11, no.

12, Art. no. 433, Dec. 2025.
[7] A. Roy et al., “MedNeXt: Transformer-Driven Scaling of ConvNets for Medical Image Segmentation,” in MICCAI, 2023.
[8] R. Solovyev, W. Wang, and T. Gabruseva, “Weighted Boxes Fusion: Ensembling Boxes from Different Object Detection Models,” arXiv preprint arXiv:1910.13302, 2019.