2024 年 明新學報 47 卷 e024002
e024002 Vol. 47,2024
Journal of Minghsin University of Science and Technology

基於動物辨識的 YOLOv5 與 YOLOv8 之性能比較

許瀟傑[1] 吳亞芬[2*] 孫子恩[3] 林義楠[4]

1.2.3.4:台灣,新北市,明志科技大學,電子工程系

摘要

本文探索比較分析 YOLOv5 與 YOLOv8 的性能及模型結構。研究中使用 Anaconda3 的 PowerShell 建立虛擬環境，運行 YOLOv5 和 YOLOv8 的程式碼，採用相同的深度模型大小（YOLOv5s、YOLOv8s）、數據集（分為兩類：dog 及 cat），並以固定的預測集進行比較分析。研究結果證實 YOLOv5 在整體準確率達到 79.7%。相比之下，YOLOv8 在整體準確率達到 82.8%。總結而言，本文研究發現 YOLOv8 在準確率上優於 YOLOv5，且其運行速度也遠優於後者。最後系統模擬驗證，採用一種基於數學理論和圖形化特性的系統建模工具 Petri Net，分別對 YOLOv5 和 YOLOv8 的模型結構進行建模與驗證，並深入了解其兩者的架構完整性。透過分析比較證實，不同版本 YOLO 模型之間的性能差異，探討了有關準確性和速度之間的取捨，以及 YOLOv8 或未來版本的性能與改進方向。

關鍵詞：YOLOv5、YOLOv8、深度學習、物件偵測、派翠網路

投稿日期:2023.12.26
修改日期:2024.03.05
接受日期:2024.04.11
刊登日期:2024.04.16

* 通訊作者：吳亞芬（教授） 地址：新北市泰山區工專路 84 號 Tel：(02)29089899 轉 5311
E-mail：yfwu@mail.mcut.edu.tw

2024 年 明新學報 47 卷 e024002
e024002 Vol. 47, 2024
Journal of Minghsin University of Science and Technology

Performance comparison between YOLOv5 and YOLOv8 based on animal recognition

Hsiao-Chieh Hsu[1] Ya-Fen Wu[2*] Tzu-En Sun[3] Yi-Nan Lin[4]

1.2.3.4: Taiwan, New Taipei City, Ming Chi University of Technology, Department of Electronic Engineering

Abstract

This study explores and compares the performance and model structures of YOLOv5 and YOLOv8. Using Anaconda3's PowerShell, a virtual environment is established to run the code for YOLOv5 and YOLOv8. The study employs models of the same depth (YOLOv5s, YOLOv8s) and datasets (divided into two categories: dogs and cats), and performs comparative analysis using a fixed prediction set. The research results confirm that YOLOv5 achieves an overall accuracy of 79.7%, while YOLOv8 achieves an overall accuracy of 82.8%. In summary, this study finds that YOLOv8 outperforms YOLOv5 in accuracy and operates significantly faster. Finally, system simulation validation is conducted using a system modeling tool based on mathematical theory and graphical characteristics, Petri Net. The modeling and validation of the model structures of YOLOv5 and YOLOv8 are carried out to gain insight into their architectural integrity. Through analysis and comparison, this study confirms performance differences between different versions of YOLO models, discusses the trade-offs between accuracy and speed, and explores the performance and improvement directions of YOLOv8 or future versions.

Keywords: YOLOv5, YOLOv8, deep learning, object detection, Petri Net

Received：26 Dec 2023
Revised：5 Mar 2024
Accepted：11 Apr 2024
Published：16 Apr 2024

* 通訊作者：吳亞芬（教授） 地址：新北市泰山區工專路 84 號 Tel：(02)29089899 轉 5311
E-mail：yfwu@mail.mcut.edu.tw

一、介紹

YOLO(You Only Look Once)是即時物件偵測中重要的技術。首次於2015年6月由Joseph Redmon和Ali Farhadi共同提出和開發的[1]。隨後在過去幾年裡生成了許多版本，每個版本都做出了些許地改動，這些改動包括架構設計、損失函數的修改和邊界框調整等。

YOLOv5是由Ultralytics公司在2020年年底開發的，其旨在提供更高效和更精確的目標檢測解決方案。目前，這項演算法在各種領域和應用中被廣泛使用，包括電腦視覺與圖像識別、自動駕駛與無人系統、醫療影像分析等。該演算法通過改進YOLOv4並採用新的模型結構和訓練技術，提高了速度和精度。與過往的版本不同，YOLOv5一次性地推出了多個版本(如s、m、l、x)，並使用Python在 PyTorch框架下進行編寫[1][4]。

YOLOv8是Ultralytics公司在2023年1月開發的，與之前的YOLO版本相比，YOLOv8引入了一些新的設計思想和技術，以提高模型的精度和速度。該演算法不僅可以在通用的目標檢測任務中表現良好，還可以應用於各種領域，如自動駕駛、工業檢測、物體識別等。該演算法在模型結構、數據增強等方面進行了優化，使得最終輸出獲得出色的結果。

Petri Net是由德國數學家Carl Adam Petri於1960年代提出的一種基於數學理論且具有圖形化特性的系統建模工具。其主要應用在建立模型、模擬以及分析工作流程網路的過程。Petri Net主要由四個基本元素構成如表1所示，包括位置(place)、轉變(transition)、位置與轉變之間的流向關係(arc)以及用於紀錄目前狀態的記號環(token) [5-7]。前三項構成了Petri Net的靜態結構，而結合記號環則使得Petri Net能夠表示動態行為。這種工具的應用範疇廣泛，尤其在描述和分析系統的同時提供了清晰的圖形化表示。Petri Net的建模能力和數學基礎使其成為工程、電腦科學等領域中一個強大的工具，用於處理複雜的流程、同步和異步事件，以及狀態轉換的動態行為。

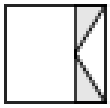
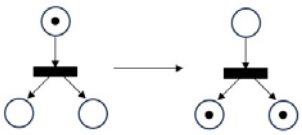
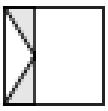
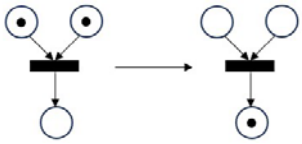
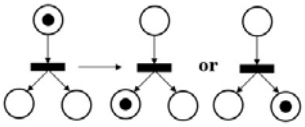
表 1. Petri Net 元素介紹

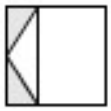
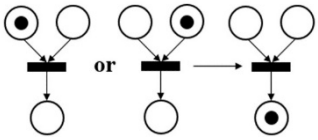
中/英文名稱	圖形	解釋
位置 / place	◯	表示系統物體或事件的狀態

變遷 / transition		表示系統中的物體或事件狀態的變化
有向弧 / arc		表示系統中物體或事件的轉換方向
記號 / token		表示系統中的移動物體或事件

WoPeD(Workflow Petri Net Designer)是一個開源軟體，遵循 LGPL(GNU Lesser General Public License)條款，由巴登-符騰堡州立聯合大學(Cooperative State University Karlsruhe)開發。該軟體的主要目的在於建立、模擬和分析工作流程網路模型[5-6]。這工具在學術領域中扮演著重要角色，為研究人員和學者提供了強大的資源，以深入探討工作流程網路的結構和行為。透過這個軟體，使用者能夠進行系統性的模型建立、模擬實驗和分析，有助於更好地理解和優化複雜的工作流程。此外 WoPeD 還額外在 Petri Net 的基礎上進行了邏輯擴充如表 2 所示，這不僅增進 Petri Net 在驗證系統方面的精確性和可靠性還擴展了對工作流程研究的理解，近一步提高其在描述現實世界問題時的適用性和效能。

表 2. WoPeD 邏輯元件介紹

And-split Transition		表示將同一資料或事件分為不同流程進行 
And-join Transition		表示多個事件或資料同時滿足後才可進行下一流程 
XOR-split Transition		表示將單一資料或事件分支成多個可能的流程，但只允許其中一條分支進行。 

XOR-join Transition		用於表示在系統的不同分支路徑合併成單一流程時的行為 

二、 架構比較

本節深入研究YOLOv5以及引進新一代模型YOLOv8的架構設計與相關內容。作為一種專注於目標檢測的模型，YOLOv5的架構特點及其獨特優勢將被詳細探討，並呈現其對目標檢測任務所帶來的顯著進展。接著，借鑑於YOLOv5的優點與前瞻性研究，新興的YOLOv8被提出，其目的在進一步提升目標檢測的性能與效率。該篇章旨在全面探討這兩種模型的架構特性，深入分析其優勢與差異性。

(一) YOLOv5架構

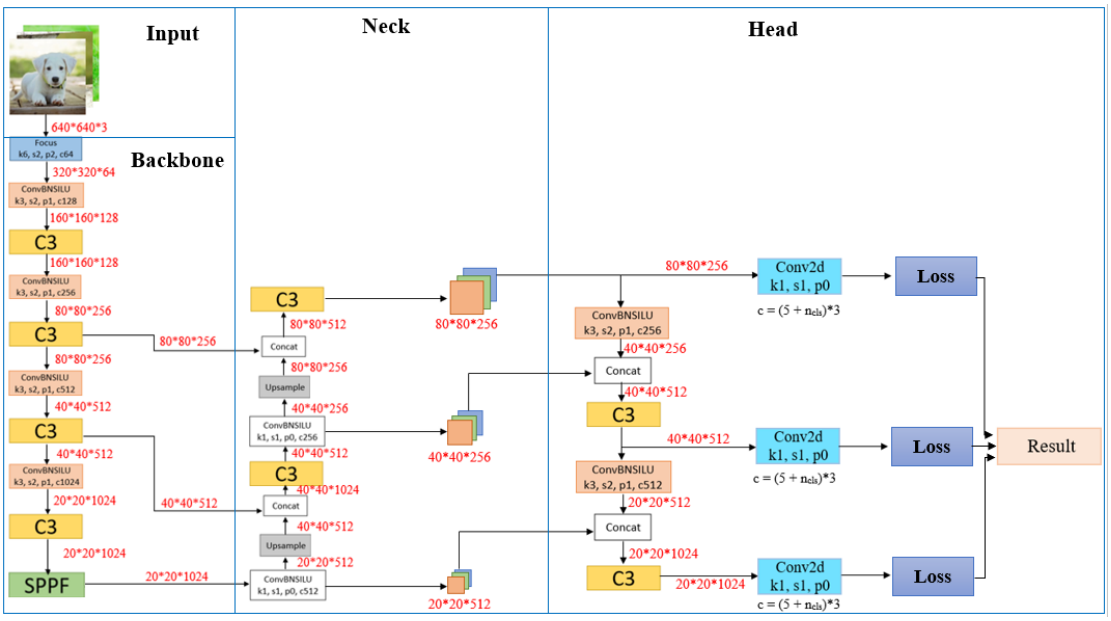


圖 1. YOLOv5 架構

YOLOv5 目標檢測模型的架構是由 Input 層、Backbone 層、Neck 層以及 Head 層結構組成[3]，如圖 1 所示。

1. Input 層匯入圖片後將會經歷 Mosaic 數據增強、動態邊界框計算、自動圖片縮放等初步的圖片處理[2]。

(1) Mosaic 數據增強：

每次讀取四張圖片後，接著進行隨機翻轉、縮放、剪裁、排佈和色域變化的方式進行拼接成 Mosaic 數據，為模型提供了更多樣化且豐富的資料。透過整合這些圖片，模型能學習更多變和複雜的場景，進而提升對不同情況下的辨識能力。

(2) 動態邊界框計算：

不同的數據集都對應著各自初始設定的長寬邊界框。這些邊界框被視為訓練過程中的參考基準，模型將基於初始邊界框輸出預測框，然後與真實邊界框進行比較，計算兩者之間的差異。接著，通過反向更新模型參數，不斷疊代優化，這有助於提高模型檢測的效率與準確度。

(3) 自動圖片縮放：

在以往的 YOLO 版本中，演算法習慣性地將圖片縮放至相同尺寸後再進行檢測。然而，這種方法會導致輸入圖像兩側出現不同尺寸的黑邊，進而影響模型推理速度，使得資料冗餘。相較於這種方式，YOLOv5 引入了一項新功能，能夠自動調整原始影像，使其周圍增加最少量的黑邊，以符合模型的需求。這個改進有效地減少了運算量，提高了目標檢測的速度，同時增加了模型的通用性和適應性。

2. Backbone 層是將處理好的圖片進行特徵點提取。此架構採用了 Focus 結構、ConvBNSILU、C3、SPPF [2]。

(1) Focus 結構：

透過圖像的切片處理和卷積運算，可獲得採樣的特徵圖。如 408*408*3 的圖片輸入至 Focus 結構後，該圖片經過切片處理轉變為 204*204*12 的特徵圖。接著，進行 64 個卷積核的卷積操作，最終得到尺寸為 204*204*64 的特徵圖。通過這樣的處理，可以更好地捕捉到圖像中的各種特徵，並且能夠更有效地進行後續的分類、檢測和識別任務。

(2) ConvBNSILU：

對輸入圖像進行卷積操作、批量歸一化和引入非線性激活函數 SiLU。這對於提升模型的表示能力和學習圖像特徵的效率具有重要意義。

(3) C3：

此為 CSPDarknet53 中的特定模塊。透過多層卷積操作來處理輸入的特徵圖，有助於更準確地檢測和識別圖像中的目標物體。

(4) SPPF:

SPPF 是一種擴展自 SPP 的技術，旨在更有效地利用不同尺寸的感受野(Receptive Field)來捕捉圖像中物體的多尺度特徵。透過 SPPF，我們能夠更全面地在不同尺度下理解和學習圖像的細節特徵，有助於提升物體檢測的精準度與清晰度。

3. Neck 層用了 Concat、Upsample、ConvBNSILU、C3，主要是將特徵圖進行特徵

融合、尺寸變換、特徵增強等操作[2]。

(1) Concat：

Concat（Concatenate 的簡稱）是一種特徵融合的方法。透過將兩個或多個數據結構按照某個特定維度或軸連接在一起，形成一個更大的數據結構，以更全面地捕捉和整合圖像中的信息。這有助於提升目標檢測的精確度和豐富度。

(2) Upsample：

上採樣是一項用於將低解析度特徵圖提升至高解析度的技術，藉此促進跨層級特徵融合。這項技術有助於增進神經網絡對目標的檢測和定位能力，使網絡能夠更有效地理解和捕捉不同尺度的特徵，進而提升目標檢測的精確性和性能。

4. Head 負責生成目標檢測的預測結果[2]。

(1) 目標類別預測：

預測層負責生成一個類別概率分布，用於表示每個檢測框所屬的目標類別。通常情況下，這個概率分布使用 Softmax 函數進行歸一化，確保每個檢測框僅屬於一個類別。

(2) 邊界框座標預測：

給出物體邊界框的位置，包括框的中心座標、寬度和高度，用以準確定位物體在圖像中的位置。

(3) 置信度分數預測：

用於評估每個方框是否涵蓋目標物體的指標。通常呈現在 0 到 1 之間的數值，數值越高代表模型對於方框內包含目標的確信程度越高。

(二) YOLOv8架構

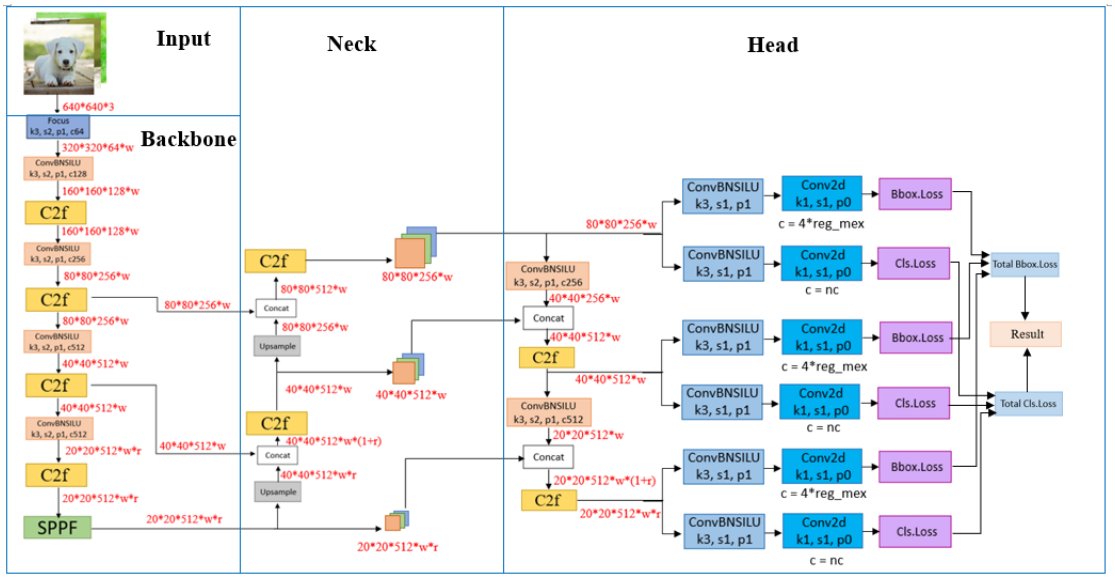


圖 2. YOLOv8 架構

YOLOv8，如圖2所示，與其前身YOLOv5在整體架構上保持了一致性，均包含Input、

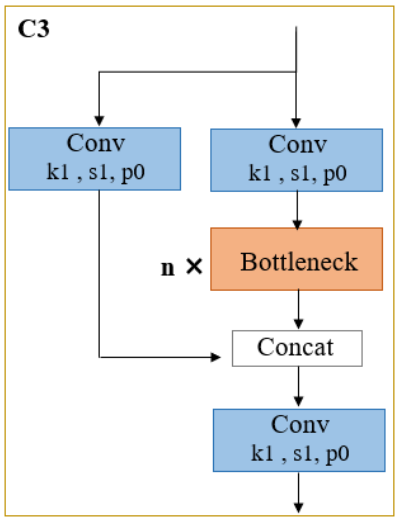
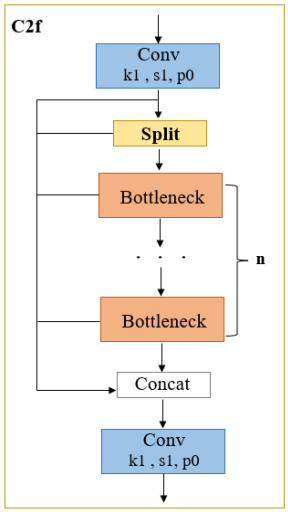
Backbone、Neck、Head。只在細微上進行了改動，這些改動旨在進一步優化目標檢測性能。

1. Backbone

(1) 將卷積核從 6*6 調整為 3*3，以進行初步特徵處理。輸入圖像經過 3*3 的卷積操作後，解析度從 640*640 被降低到 320*320。此操作旨在進行初步的特徵提取，同時對圖像進行解析度縮減。

(2) 將所有的 C3 模塊替換成梯度信息更豐富的 C2f 模塊 [1-2]，如表 3 所示，進一步增加跳層連結。C2f 模塊的設計受到 C3 模塊和 ELAN 模塊的啟發，以確保 YOLOv8 在保持輕量化的同時，獲取更加豐富的梯度流信息。ELAN 模塊是 YOLOv7 所引入的一種模塊，其能夠補獲更為豐富的梯度信息，進而實現更高的精確度和更合理的推理時間延遲。

表 3. C3 結構與 C2f 結構差異圖

C3 結構	C2f 結構
	

2. Neck

(1) 將所有的C3模塊替換成梯度信息更豐富的C2f模塊[1-2]，進一步增加跳層連結。

(2) 取消兩個 1*1 卷積層(ConvBNSILU)的操作，改為直接將 Backbone 不同階段的輸出特徵通過 Concatenation(拼接)操作後，直接送入 Upsample 層。這種修改旨在簡化模型架構，省略特定層次的卷積處理，使得不同階段的特徵能夠直接被整合和上採樣，以提高計算效率和減少參數數量。

3. Head

從原先的耦合變成解耦，將目標分類和偵測框架目標回歸兩個行動解耦，並分別進行預測，這種解耦合的操作有助於加速模型的收斂速度並提高目標檢測的

精度[1-2]。

三、實驗準備

本節進行規劃流程與實驗，以確保其穩定性和可比較性。為了建立基準，本文將Yolov5與Yolov8的模型深度與訓練圖庫固定，使其能在相同的環境下進行模擬。此外，訓練圖庫使用了LabelImg軟體來獲得.txt的資料檔，以確保其符合YOLO模型的格式，使最終能夠獲取具有高度標準化的實驗數據。這一步驟旨在確保數據的一致性和可比性，以便在實驗過程中獲得可靠、可重複且具有參考價值的結果。

(一) 實驗流程

本文透過 Anaconda 建立了兩個獨立的虛擬環境，分別載入了 Yolov5s 與 Yolov8s 的模型，以及 PyTorch 深度學習框架。這些模型是進行物體檢測的基本工具，可用於辨識圖像中的物件與其位置。本文中的訓練圖庫使用了 100 張的圖像數據集，其中包含了 50 張 dog 的圖像以及 50 張 cat 的圖像。此訓練圖庫透過 LabelImg 軟體[6]對圖像進行了特徵邊界框的標記，並生成符合 YOLO 格式的資料檔，如表 4 所示，其標記的結果涵蓋了目標類別(dog 和 cat 兩種類別；0 為 dog，1 為 cat)、標註的邊界框在圖片中的 X_{center} (標記框歸一化後的中心 X 座標)，如表 5 中公式 1.3 所示、Y_{center} (標記框歸一化後的中心 Y 座標)，如表 5 中公式 1.4 所示、$YOLO_w$ (標記框歸一化後相對應的寬)，如表 5 中公式 1.5 所示、$YOLO_h$ (標記框歸一化後相對應的高)[4]，如表 5 中公式 1.6 所示，實際目標特徵標記框座標，如圖 3 所示。此外，為了評估所建立的模型性能，另準備了 10 張 dog 和 10 張 cat 的驗證圖像，用於模型準確性的評估與驗證。在 Yolov5s 和 Yolov8s 的獨立虛擬環境中，啟動了訓練過程，使用先前標記好的訓練圖像。這兩種模型將都使用 batch-size=8 的情況下進行一段時間的訓練，並生成多張訓練結果的圖片，這些結果將有助於評估模型對於 dog 和 cat 的檢測效果。

表 4. YOLO 格式的文件

類別	X_{center}	Y_{center}	$YOLO_w$	$YOLO_h$
0	0.546667	0.283750	0.326667	0.512500
1	0.510000	0.261667	0.513333	0.301111

表 5. YOLO 格式中各項數值的計算公式

$w = X\ max - X\ min$	(公式 1.1)
$h = Y\ max - Y\ min$	(公式 1.2)
$X_{center} = \frac{(X\ min + X\ max)}{2} * \frac{1}{w}$	(公式 1.3)

$Y_{center} = \frac{(Y\ min + Y\ max)}{2} * \frac{1}{h}$	(公式 1.4)
$YOLO_{w} = \frac{(X\ max - X\ min)}{w}$	(公式 1.5)
$YOLO_{h} = \frac{(Y\ max - Y\ min)}{h}$	(公式 1.6)

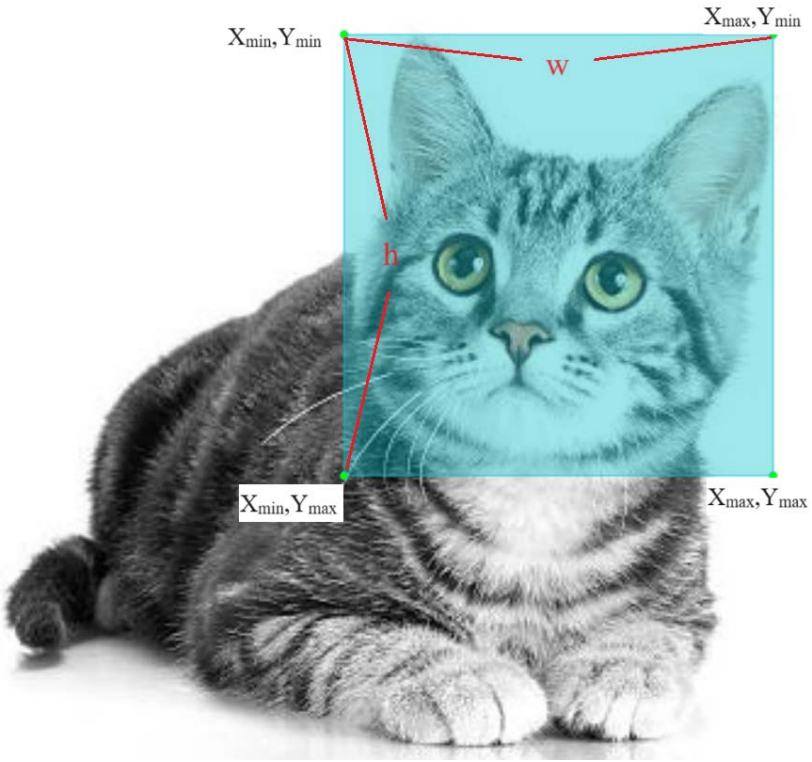


圖 3. 目標特徵標記框座標

四、 實驗數據比較

本節主要論述聚焦於對比YOLOv5和YOLOv8兩個目標檢測模型的輸出結果以及在WoPeD中Petri Net結果的比較分析。

(一) 比較項目解析

(1) 輸出結果比較：

評估並對比 YOLOv5 和 YOLOv8 兩種目標檢測模型的輸出結果。將透過相同訓練圖庫與實驗條件下的準確率和回歸率等性能指標，深入分析兩者的檢

測結果。同時，將探討這兩種模型的強項和弱項。

(2) 在 WoPeD 中的 Petri Net 結果比較：

分析並對比將 YOLOv5 與 YOLOv8 模型繪製到 WoPeD 中所產生的 Petri Net 結果。將針對兩種模型在 WoPeD 環境下 Petri Net 結果的差異進行深入探討。分析模型的效率、速度等關鍵特徵。

(二) YOLOv5 與 YOLOv8 的輸出結果與比較

根據 YOLOv5 的輸出結果，如圖 4 所示，整體準確率達到 79.7%。在該結果中，dog 類的準確率為 90%，cat 類的準確率達到 69.4%，如表 6 所示。目標檢測的整體回歸率為 95%；當 IoU 為 0.5 時，平均精度達到 96.4%；在 IoU 從 0.5 到 0.95 範圍內的平均精確度為 76.4%，如圖 5 所示。

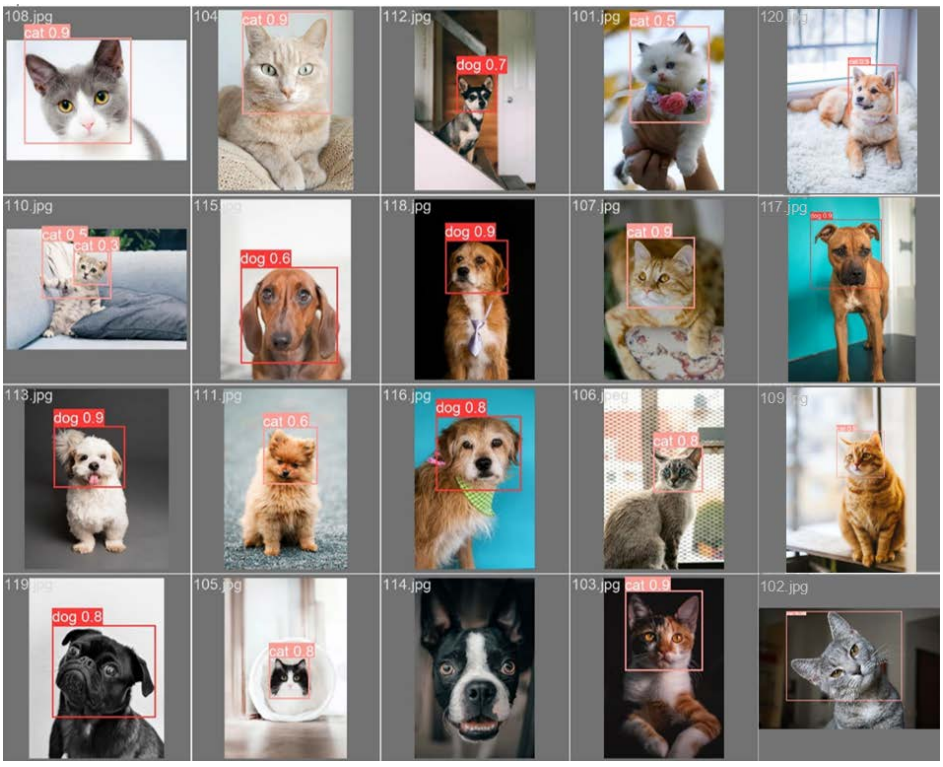


圖 4. YOLOv5 測試集結果

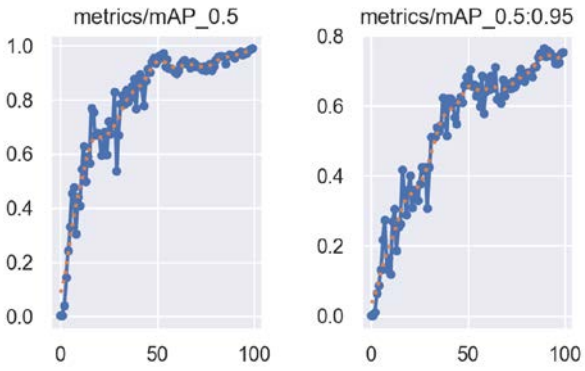


圖 5. YOLOv5 訓練結果線性回歸圖

根據 YOLOv8 的輸出結果，如圖 6 所示，整體準確率達到 82.8%。在該結果中，dog 類的準確率為 89.1%，cat 類的準確率達到 76.5%，如表 6 所示。目標檢測的整體回歸率為 98.9%；當 IoU 為 0.5 時，平均精度達到 97.4%；在 IoU 從 0.5 到 0.95 範圍內的平均精確度為 78.9%，如圖 7 所示。

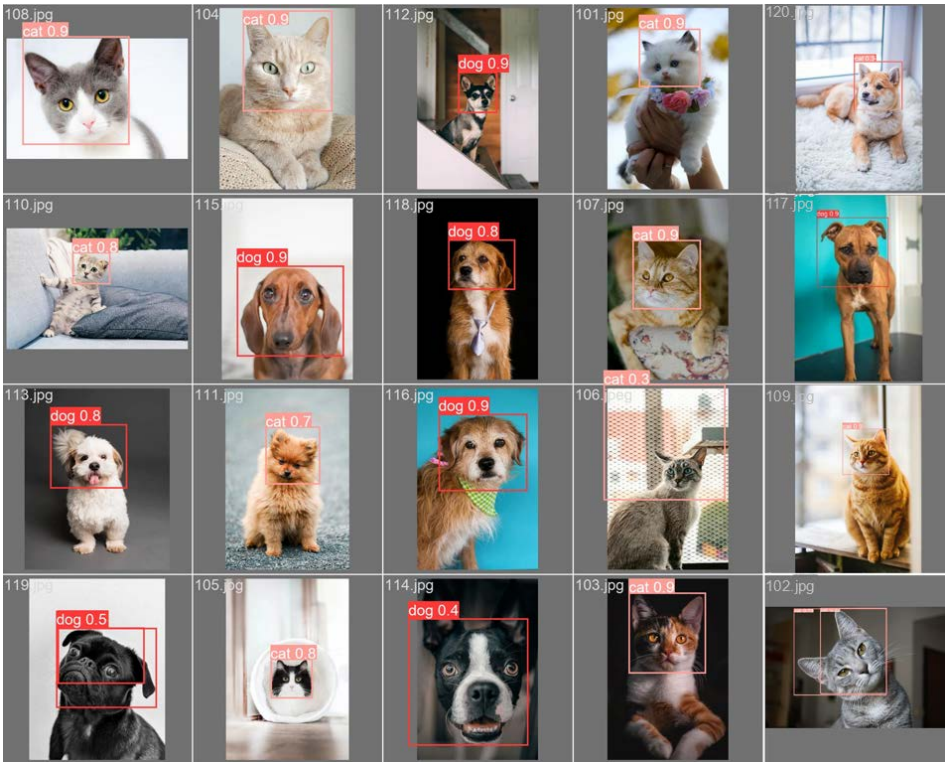


圖 6. YOLOv8 測試集結果

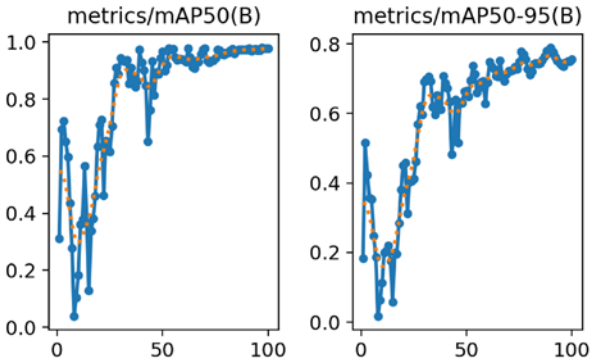


圖 7. YOLOv8 訓練結果線性回歸圖

(1) 準確率(Precision)是指在所有模型預測為正類(Positive)的樣本中，實際為正類的樣本所占的比例。其計算公式為 Precision = TP/(TP+FP)，其中 TP 代表真正例(True Positive) [2][6]，亦即將 dog 類預測為 dog 類。FP 則代表誤判為正例的負類樣本(False Positive)，亦即將 cat 類預測為 dog 類。

(2) 回歸率(Recall)則表示在所有實際為正類的樣本中，被模型正確預測為正類的樣

本所占的比例。其計算公式為 Recall = TP/(TP+FN)，其中 TP 仍為真正例(True Positive)，FN 則為未能正確辨別的正類樣本(False Negative) [2][6]，亦即將 dog 預測為 cat。這兩者是用於評估分類模型的重要指標，準確率衡量了模型預測為正類的準確性，而回歸率則衡量了模型對正類樣本的覆蓋程度，兩者在模型評估和優化中扮演著不可或缺的角色。

(3) mAP50 通常代表在目標檢測任務中，使用 IoU(交集比聯集) [4]為 0.5 時計算的平均精度。此指標反映了模型在預測結果與實際標籤匹配時，若預測邊界框與實際邊界框的 IoU 大於 0.5，則將其視為有效檢測結果。而 mAP50 至 95 則涵蓋了 IoU 從 0.5 至 0.95(間隔 0.05)範圍內的平均精度，對應著在不同 IoU 值下(0.5、0.55、0.6、0.65、0.7、0.75、0.8、0.85、0.9、0.95)的平均 mAP。該指標廣泛應用於評估目標檢測模型在不同 IoU 條件下的性能表現。

(4) 基於所列數據，如表 6 所示，YOLOv8 相較於 YOLOv5 在 P(準確率)、R(回歸率)、mAP50(在 IoU 為 0.5 時的平均精度)、mAP50~95(在 IoU 為 0.5 到 0.95 範圍內的平均精度)等性能指標均表現優異。YOLOv5 的正規化速率及總運行時長明顯高於 YOLOv8，這代表 YOLOv5 不僅在準確性上有所弱於 YOLOv8，且模型的推理速度和效率方面皆不如 YOLOv8。這些結果凸顯了兩個模型間在性能提升和運行效率之間的差別。YOLOv8 通過將 C3 模塊更改成 C2f 模塊，增加跳層連結進而達成輕量化，使 YOLOv8 在準確率與回歸率不降的同時還能降低運算的時間成本。

表 6. YOLOv5 與 YOLOv8 的輸出結果比較

Items	YOLOv5	YOLOv8
Precision(P)	90%	89.1%
Recall(R)	69.4%	76.5%
Mean Average Precision 50 (mAP 50)	96.4%	97.4%
Mean Average Precision 50~95 (mAP 50~95)	76.4%	78.9%
運算時間成本 (平均每筆資料所需時間比例)	10.7	1

(三) Petri Net 建模與驗證

YOLOv5 使用 Petri Net 建模時，共涉及了 35 個 Places、28 個 Transitions(11 個 Operators)以及 68 個 Arcs。建模後的驗證結果顯示架構流程符合 Petri Net 基本規範，且流程上並無錯誤，如圖 8、圖 9、表 7 所示。

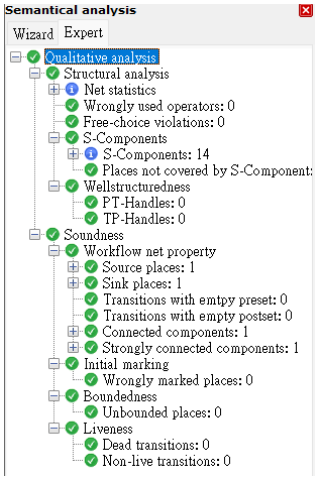


圖 8. YOLOv5 的 Petri Net 驗證結果

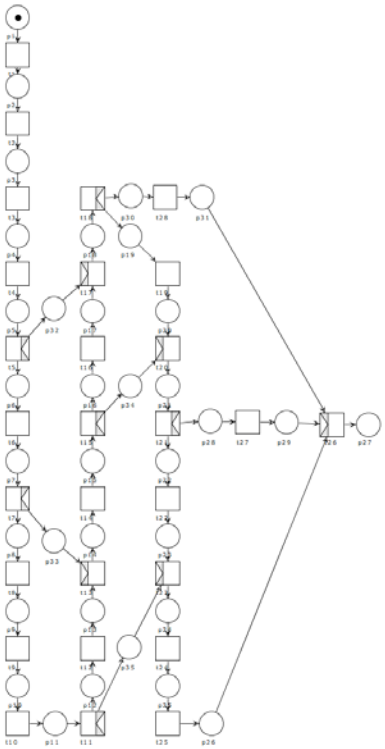


圖 9.YOLOv5 的 Petri Net 建模

表 7. YOLOv5 的 Place 和 Transition

位置點 (Place)	內容	轉移點 (Transition)	內容
P1	輸入原始圖像 (長 640*寬 640*卷積核數 3)	T1	Focus 結構

P2	獲得圖像 (長 320*寬 320*卷積核數 64)	T2	ConvBNSILU
P3	獲得圖像 (長 160*寬 160*卷積核數 128)	T3	C3
P4	獲得圖像 (長 160*寬 160*卷積核數 128)	T4	ConvBNSILU
P5	獲得圖像 (長 80*寬 80*卷積核數 256)	And-Split T5	C3
P6	獲得圖像 (長 80*寬 80*卷積核數 256)	T6	ConvBNSILU
P7	獲得圖像 (長 40*寬 40*卷積核數 512)	And-Split T7	C3
P8	獲得圖像 (長 40*寬 40*卷積核數 512)	T8	ConvBNSILU
P9	獲得圖像 (長 20*寬 20*卷積核數 1024)	T9	C3
P10	獲得圖像 (長 20*寬 20*卷積核數 1024)	T10	SPPF
P11	獲得圖像 (長 20*寬 20*卷積核數 1024)	And-split T11	ConvBNSILU
P12	獲得圖像 (長 20*寬 20*卷積核數 512)	T12	Upsample
P13	獲得圖像 (長 40*寬 40*卷積核數 512)	And-join T13	Concat
P14	獲得圖像 (長 40*寬 40*卷積核數 1024)	T14	C3
P15	獲得圖像 (長 40*寬 40*卷積核數 512)	And-Split T15	ConvBNSILU
P16	獲得圖像 (長 40*寬 40*卷積核數 256)	T16	Upsample

P17	獲得圖像 (長 80*寬 80*卷積核數 256)	And-join T17	Concat
P18	獲得圖像 (長 80*寬 80*卷積核數 512)	And-split T18	C3
P19	獲得圖像 (長 80*寬 80*卷積核數 256)	T19	ConvBNSILU
P20	獲得圖像 (長 40*寬 40*卷積核數 256)	And-join T20	Concat
P21	獲得圖像 (長 40*寬 40*卷積核數 512)	And-split T21	C3
P22	獲得圖像 (長 40*寬 40*卷積核數 512)	T22	ConvBNSILU
P23	獲得圖像 (長 20*寬 20*卷積核數 512)	And-join T23	Concat
P24	獲得圖像 (長 20*寬 20*卷積核數 1024)	T24	C3
P25	獲得圖像 (長 20*寬 20*卷積核數 1024)	T25	Conv2d
P26	獲得 Loss	T26	將各個 Loss 總和
P27	獲得生成目標檢測的預測結果	T27	Conv2d
P28	獲得圖像 (長 40*寬 40*卷積核數 512)	T28	Conv2d
P29	獲得 Loss		
P30	獲得圖像 (長 80*寬 80*卷積核數 256)		
P31	獲得 Loss		
P32	獲得圖像 (長 80*寬 80*卷積核數 256)		
P33	獲得圖像		

	(長 40*寬 40*卷積核數 512)		
P34	獲得圖像 (長 40*寬 40*卷積核數 256)		
P35	獲得圖像 (長 20*寬 20*卷積核數 512)		

YOLOv8 使用 Petri Net 建模時，共涉及了 47 個 Places、37 個 Transitions(14 個 Operators)以及 92 個 Arcs。建模後的驗證結果顯示架構流程符合 Petri Net 基本規範，且流程上並無錯誤，如圖 10、圖 11、表 8 所示。

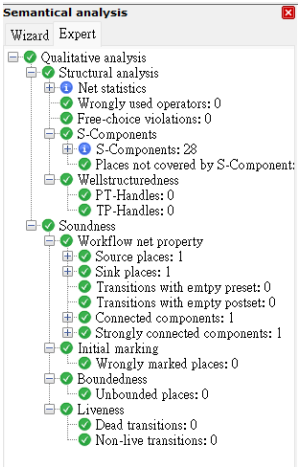


圖 10 YOLOv8 的 Petri Net 驗證結果

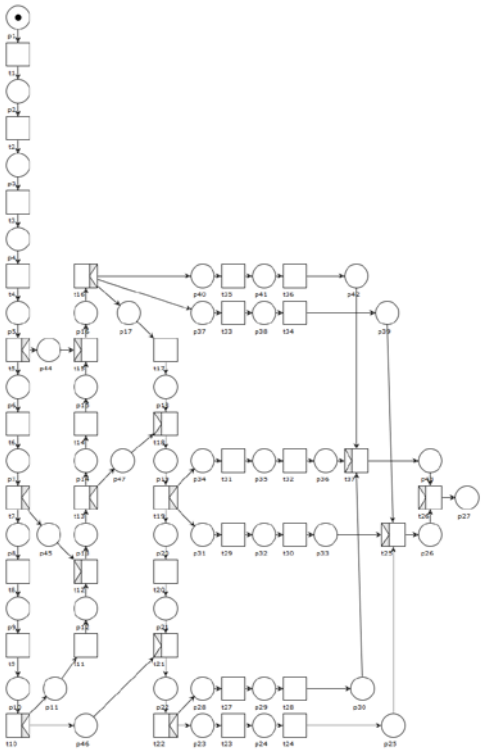


圖 11 YOLOv8 的 Petri Net 建模

表 8.YOLOv8 的 Place 和 Transition

位置點 (Place)	內容	轉移點 (Transition)	內容
P1	輸入圖像 (長 640*寬 640*卷積核數 3)	T1	Focus 結構
P2	獲得圖像 (長 320*寬 320*卷積核數 64*特徵圖通道數 w)	T2	ConvBNSILU
P3	獲得圖像 (長 160*寬 160*卷積核數 128*特徵圖通道數 w)	T3	C2f
P4	獲得圖像 (長 160*寬 160*卷積核數 128*特徵圖通道數 w)	T4	ConvBNSILU
P5	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)	And-split T5	C2f
P6	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)	T6	ConvBNSILU
P7	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	And-split T7	C2f
P8	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T8	ConvBNSILU
P9	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 r)	T9	C2f
P10	獲得圖像	And-split T10	SPPF

	(長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 r)		
P11	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 r)	T11	Upsample
P12	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w*額外維度數 r)	And-join T12	Concat
P13	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w*額外維度數 1+r)	And-split T13	C2f
P14	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T14	Upsample
P15	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)	And-join T15	Concat
P16	獲得圖像 (長 80*寬 80*卷積核數 512*特徵圖通道數 w)	And-split T16	C2f
P17	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)	T17	ConvBNSILU
P18	獲得圖像 (長 40*寬 40*卷積核數 256*特徵圖通道數 w)	And-join T18	Concat
P19	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	And-split T19	C2f
P20	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T20	ConvBNSILU

P21	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w)	And-join T21	Concat
P22	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 1+r)	And-split T22	C2f
P23	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w 額外維度數 r)	T23	ConvBNSILU
P24	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 r)	T24	Conv2d
P25	獲得 Cls.Loss	And-split T25	將各個 Cls.Loss 總和
P26	獲得總 Cls.Loss	And-join T26	將 Cls.Loss 和 Bbox.Loss 總和
P27	獲得生成目標檢測的預測結果	T27	ConvBNSILU
P28	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 r)	T28	Conv2d
P29	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w*額外維度數 r)	T29	ConvBNSILU
P30	獲得 Bbox.Loss	T30	Conv2d
P31	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T31	ConvBNSILU
P32	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T32	Conv2d
P33	獲得 Cls.Loss	T33	ConvBNSILU

P34	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T34	Conv2d
P35	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)	T35	ConvBNSILU
P36	獲得 Bbox.Loss	T36	Conv2d
P37	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)	And-split T37	將各個 Bbox.Loss 總和
P38	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)		
P39	獲得 Cls.Loss		
P40	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)		
P41	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)		
P42	獲得 Bbox.Loss		
P43	獲得總 Bbox.Loss		
P44	獲得圖像 (長 80*寬 80*卷積核數 256*特徵圖通道數 w)		
P45	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)		
P46	獲得圖像 (長 20*寬 20*卷積核數 512*特徵圖通道數 w)		

P47	獲得圖像 (長 40*寬 40*卷積核數 512*特徵圖通道數 w)		

基於 WoPeD 上對 YOLOv5 和 YOLOv8 在 Petri Net 結果的驗證，如表 9 所示，YOLOv8 在 Places、Transitions 和 Arcs 的數量上均遠超過 YOLOv5。這些結果證實了 YOLOv8 模型架構呈現出更高層次的複雜性。然而，除了結構差異外，本文進一步比較兩個模型的運行時間。對比運行時間顯示出明顯的差異，YOLOv5 的運行時間明顯較短，表示其具有較高的運行效率。相對地，YOLOv8 進行目標檢測任務需要更長時間。但因本次構建兩個模型時，並未更細分 C3 與 C2f 模塊的內部構造。於是將此結果對比前文的訓練結果會發現 YOLOv8 能快於 YOLOv5 的訓練時間最主要的原因應該就是 C3 與 C2f 的差別，C2f 的跳層連結架構具有更多的層次、參數及神經元連接，使其 YOLOv8 運行時間的減少也許就是由於這種模塊複雜性所帶來的結果。

表 9. YOLOv5 與 YOLOv8 的 Petri Net 比較

Items	YOLOv5	YOLOv8
Places	35	47
Transitions(包含 Operators)	28	37
Arcs	68	92
模型運行時間 (一筆數據跑完的時間比例)	1	1.29

五、 結論

本文深入探討比較YOLOv5s和YOLOv8s兩種物體檢測模型的性能及模型結構，並透過實際的訓練和驗證圖像的應用，深入了解其在不同任務上的表現。以下是比較之主要結論：

(一) 性能比較：

根據實驗結果YOLOv8s在dog和cat的數據集上的檢測準確率優於YOLOv5s。具體而言，YOLOv8s整體準確率達到82.8%。相比之下，YOLOv5在相同數據集上的表現較差。

(二) 速度比較：

雖YOLOv8s不僅在檢測準確率上較為優秀，其訓練速度也明顯優於YOLOv5s。

(三) 模型結構比較：

透過對模型結構的深入分析，發現 YOLOv8 在 Backbone 和 Neck 部分進行了一系列的改進，包括使用了更小的卷積核、引入新的模塊、優化特徵融合等。這些變化可能是導致其在檢測性能上優於 YOLOv5 的原因之一。

(四) 模型結構建模與驗證：

引入 Petri Net 作為一種系統建模工具，對 YOLOv5s 和 YOLOv8s 的模型結構進行了建模與驗證。這一步驟有助於更深入地了解模型的結構完整性，並為未來的優化提供了方向。

綜合上述，本文提供了對於 YOLOv5 和 YOLOv8 在物體檢測任務上性能和結構方面的詳盡比較。在實際應用中，應根據具體需求權衡不同模型的優勢，這有助於選擇最適合特定任務的物體檢測模型。未來的研究方向可以包括進一步優化模型結構、提升速度和準確性的平衡，以及應用更多先進的技術進行模型訓練和評估。

六、 參考文獻

[1] Liu Chen, Li Shixin, Meng Fanrun, Chen Fankai. “A Review of Unmanned Aerial Vehicle Target Detection Based on Deep Learning[J]”. Computer Science and Application, 2023, 13(5): 1092-1099. DOI: https://doi.org/10.12677/CSA.2023.135107.

[2] Zhang Chuanqing, Chen Yangshan, Liu Xiding, Wang Jianqiang, Zhao Shuguang. Research on Recyclable Waste Identification Method Based on YOLOv8[J]. Computer Science and Application, 2023, 13(5): 1019-1025. DOI: https://doi.org/10.12677/CSA.2023.135099.

[3] Q. Li, Z. Ge, F. Wang, X. Luo, Y. Yang and T. Yu, "Small Target Repair Parts Detection Algorithm Based on Improved YOLOv5," 2022 10th International Conference on Information Systems and Computing Technology (ISCTech), Guilin, China, 2022, pp. 420-433, DOI: 10.1109/ISCTech58360.2022.00073.

[4] Jr-Jena Huang, Cheng-Yingb Yang, Yi-Nana Lin, Victor R.L. Shen, Chia-Tsaia Lin, Frank H.C. Shen, ” Novel Emotion Recognition System Using Edge Computing Platform with Deep Convolutional Networks,” Journal of Intelligent & Fuzzy Systems, vol. 45, no. 2, pp. 2669-2683, 2023. DOI: 10.3233/JIFS-223801

[5] Yi-Nan Lin, Sheng-Kuan Wang, Gwo-Jen Chiou, Cheng-Ying Yang, Victor R. L. Shen, Zhi Yang Su, ”Development and Verification of an IoT-enabled Air Quality Monitoring System Based on Petri Nets,” Wireless Personal Communications volume 131, pages63–87, 2023. DOI: https://doi.org/10.1007/s11277-023-10416-1

[6] Yang, CY., Lin, YN., Wang, SK. et al. An Edge Computing System for Fast Image Recognition Based on Convolutional Neural Network and Petri Net Model. Multimed Tools Appl 83, 12849–12873 (2024). DOI: https://doi.org/10.1007/s11042-023-15388-9

[7] C. -Y. Yang, Y. -N. Lin, V. R. L. Shen, F. H. C. Shen and C. -C. Wang, "A Novel IoT-Enabled System for Real-Time Face Mask Recognition Based on Petri Nets," in IEEE Internet of Things Journal, vol. 11, no. 4, pp. 6992-7001, 15 Feb.15, 2024, DOI: 10.1109/JIOT.2023.3313583.