

國立臺灣大學工學院工程科學及海洋工程學系

碩士論文

Department of Engineering Science and Ocean Engineering

College of Engineering

National Taiwan University

Master's Thesis

應用於偵測視訊模糊物件的強健性類神經網路

Robust Neural Network for Video Object Detection in Blurred
Environments

徐聖淮

Sheng-Huai Hsu

指導教授：丁肇隆 博士

Advisor: Chao-Lung Ting, Ph.D.

中華民國 113 年 6 月

June, 2024

誌謝



時光荏苒，轉眼間來到了研究所生涯的尾聲，還記得當初進入研究所前的夢想：學習並完成一份深度學習相關的研究，到此終於實現並劃下句點。

本篇論文的完成，首先最想感謝的是我的指導教授丁肇隆老師，非常感謝老師總是在百忙之中撥冗與我們開會，在每次研究遇到問題時都能適時的指引我們，給予我們改進的方法。也非常感謝老師在我們進行論文撰寫時細心的進行修改及補充，以及教導我們嚴謹的研究態度及為人處世的道理。除此之外，也要感謝教育學程的徐式寬老師、陳文進老師，多領域的指導開拓了我的視野，讓我的研究所生涯更加多彩豐富且完整。

感謝實驗室的大家，感謝楷諭學長、佩宜學姊及庭寧學姊在我剛進研究所時給予我課程上的建議與幫忙；感謝宥辰及翊瑄，從碩一開始就一起進行修課學習，課業遇到困難時相互打氣與幫忙；感謝容誠及子霆，在伺服器遇到問題時都能快速地協助解決，並在論文口試時協助進行記錄。在實驗室的各種點滴大小事對我而言都是彌足珍貴的回憶，希望你們在未來的道路上都能有好的發展。

此外，感謝教育學程的楷宗學長，在我進行教學服務時耐心給予課程教學的方法及建議；感謝我的摯友們佑齊、又誠、嘉葳、俊廷，在遇到問題時一起討論解決、在放鬆時一同出遊玩樂。祝你們未來的事業都能蒸蒸日上、順順利利。

最後，我要特別感謝我的家人，感謝你們的支持及鼓勵，讓我經濟無慮、專心投入碩士研究，是最大的功臣與最溫暖的避風港。碩士生涯說長不長、說短不短，因為有各位的幫助，才成就了現在的我，感謝幫助過我的每一個人，再次，感謝。

徐聖淮

2024/06/13

摘要



自 21 世紀電腦運算速度呈爆發性成長後，深度學習方法開始在許多領域進行推行與應用，其中針對物件偵測所設計的類神經網路以 YOLO 系列為大宗。然而在真實場景應用中，常需要面對影像因為錄製者本身的晃動、鏡頭變焦、物體移動，甚至是場景內的霧氣所帶來影像模糊的問題。本研究使用 YOLOX [24] 作為基礎模型，改善了傳統 YOLO 模型應用於單幀模糊影像的表現，並與現有的多幀偵測方法 YOLOV [22] 進行結合，實現在單幀與多幀情境下，均能進行穩定預測的強健性類神經網路。本研究改進了現有靜態物件偵測模型的前處理方法，額外加入了全局灰階高斯模糊影像進行訓練，並優化損失函數以契合模糊影像的預測需求，實現在性能改善的同時，又不需額外時間來進行預測的新模糊影像偵測模型，並兼具新網路模型應用於各情境及新模型的泛用性。

關鍵字：模糊物件偵測、視訊物件偵測、高斯模糊、類神經網路、影像處理

Abstract



Since the explosive growth in computing speed in the 21st century, many applications of deep learning have been implemented across various fields. Currently, the neural networks designed for object detection primarily consist of the YOLO series. However, in real-world applications, images often face challenges such as motion blur from the recorder's movement, camera zoom, object movement, or even image blurring due to fog within the scene. This study utilizes YOLOX [24] as the base model, improving the performance of traditional YOLO models applied to single-frame blurry images. It integrates with existing multi-frame detection methods like YOLOV [22] to achieve robust neural networks capable of stable predictions in both single-frame and multi-frame scenarios. The study enhances the preprocessing methods of existing static object detection models by incorporating new globally grayscale Gaussian blurry images for training. It optimizes the loss function to meet the predictive needs of blurry images, achieving performance improvements without requiring additional time for predicting new blurry image detection models. This approach also ensures the versatility of the new network model across various scenarios and the general applicability of the new model.

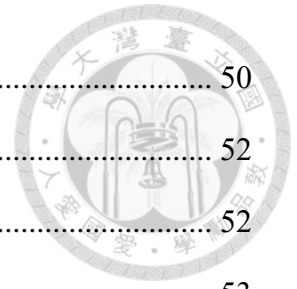
Keywords: Blur Object Detection, Video Object Detection, Gaussian Blur, Neural Networks, Image Processing

目次



誌謝	i
摘要	ii
Abstract.....	iii
目次	iv
圖次	vi
表次	vii
第一章 緒論	1
1.1 研究背景與動機	1
1.2 研究目標	3
1.3 論文架構	4
第二章 相關研究	5
2.1 單幀靜態物件偵測	5
2.2 視訊物件偵測	7
2.3 模糊物件偵測	9
2.4 小結	10
第三章 研究方法	11
3.1 問題定義	11
3.2 研究架構與設計	12
3.3 研究範圍與限制	23
第四章 實驗結果與討論	25
4.1 資料集篩選	25
4.2 單幀情境下識別靜態影像的清晰物件	29
4.3 單幀情境下識別動態影像的模糊物件	40
4.4 多幀情境下識別動態影像的模糊物件	48

4.5 小結	50
第五章 結論與未來展望	52
5.1 研究結論	52
5.2 未來展望	53
參考文獻	54
附錄	56



圖次



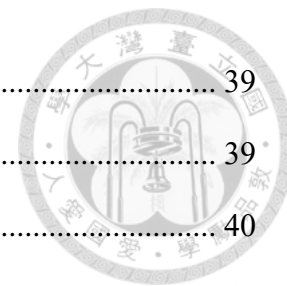
圖 1-1. 靜態物件偵測模型應用於模糊動物辨識的預測表現.....	2
圖 1-2. 動態物件偵測模型應用於模糊動物辨識的預測表現.....	2
圖 2-1. YOLO 與 SSD 的網路模型架構 [19].....	6
圖 2-2. YOLOX 的 Decoupled Head 設計 [24].....	7
圖 2-3. 採用 Decoupled Head 的 YOLOX 預測表現 [24]	7
圖 2-4. 採用多幀參考進行預測的 MEGA 網路架構 [21]	8
圖 2-5. YOLOV 之網路架構 [22]	9
圖 2-6. DeblurGANv2 於 Kohler dataset 中的去模糊表現 [12]	10
圖 3-1. 研究架構圖	14
圖 3-2. YOLOX 所採用之 Mosaic 方法	15
圖 3-3. YOLOX 所採用之 Mixup 方法	16
圖 3-4. 同影片中貓與狐狸的不同模糊表現.....	19
圖 3-5. 不同 Sigma 值之高斯模糊表現.....	20
圖 3-6. 使用局部高斯模糊的影像前處理結果.....	20
圖 3-7. 利用多幀影像偵測模糊物件類神經網路訓練流程.....	22
圖 3-8. 實驗網路架構圖	23
圖 4-1. YOLOX 與 YOLOV 的各類別 AP50 對比.....	28
圖 4-2. 固定學習率對比 Cosine 學習率曲線.....	37
圖 4-3. 本研究方法應用於模糊動物辨識的預測表現.....	51
圖 4-4. 預測錯誤的物件類別	51

表次



表 1-1. 視訊中模糊物件的種類.....	3
表 3-1. ImageNet VID 物件類別.....	12
表 3-2. YOLOX 與 YOLOV 在 ImageNet VID 的驗證集表現.....	13
表 4-1. 硬體設備與軟體開發環境.....	25
表 4-2. ImageNet DET 30 類物件總數量.....	26
表 4-3. ImageNet VID 30 類物件總數量.....	26
表 4-4. ImageNet VID 10% 訓練集物件數量列表.....	27
表 4-5. YOLOX 與 YOLOV 在 ImageNet VID 驗證集的各類別 AP50 表現.....	28
表 4-6. 本研究探討之 10 類物件.....	29
表 4-7. ImageNet DET 訓練集 10 類物件之數量.....	30
表 4-8. ImageNet DET 驗證集 10 類物件之數量.....	30
表 4-9. 本研究之 ImageNet DET 訓練集 10 類物件之數量.....	31
表 4-10. 本研究之 ImageNet DET 驗證集 10 類物件之數量.....	31
表 4-11. 本研究之 ImageNet DET 測試集 10 類物件之數量.....	31
表 4-12. 本實驗之共同超參數.....	32
表 4-13. ImageNet DET Train 1000 物件數量列表.....	32
表 4-14. YOLOX 中不同 Mosaic:Mixup 比例之實驗結果.....	33
表 4-15. ImageNet DET Train 2000 10 類物件之數量.....	34
表 4-16. ImageNet DET Train 4000 10 類物件之數量.....	34
表 4-17. YOLOX 中不同數量資料集的訓練成效.....	34
表 4-18. ImageNet DET Train 1600 物件數量列表.....	35
表 4-19. ImageNet DET Train 2400 物件數量列表.....	35
表 4-20. 多階段資料實驗之訓練成效.....	36
表 4-21. 固定學習率與 Cosine 學習率的對比實驗.....	38

表 4-22. 損失函數中相異之 Reg Weight 的 AP50 表現.....	39
表 4-23. 損失函數中相異之 IOU Loss 算法的 AP50 表現.....	39
表 4-24. 損失函數中相異之 Object Loss 算法的 AP50 表現.....	40
表 4-25. 單幀情境識別實驗之最終設定.....	40
表 4-26. ImageNet VID 驗證集 10 類物件之影片分析.....	41
表 4-27. ImageNet VID 驗證集 10 類物件之模糊物件分析.....	41
表 4-28. ImageNet VID 驗證集 10 類物件之模糊種類分析.....	41
表 4-29. ImageNet VID 測試集 10 類物件數量列表.....	42
表 4-30. ImageNet VID 模糊影片測試集 10 類物件數量列表.....	42
表 4-31. 不同模糊比例之全局高斯模糊實驗.....	43
表 4-32. 全局模糊與全局結合局部模糊的預測表現.....	44
表 4-33. 色彩敏感度與不同模糊程度之比較.....	45
表 4-34. 混合彩色與灰階的模糊物件預測表現.....	46
表 4-35. 彩色與灰階最佳超參數的模糊物件預測表現.....	47
表 4-36. ImageNet DET 4000+ ImageNet VID 10% 10 類之物件數量列表.....	47
表 4-37. 全局結合真實模糊之預測表現.....	48
表 4-38. 多幀情境下不同訓練集所得之預測成效.....	49
表 4-39. YOLOX、YOLOV 及本研究在模糊視訊物件的預測表現.....	50
表 4-40. 本研究與現有模型之各項表現比較.....	50



第一章 緒論



自 21 世紀電腦運算速度呈爆發性成長後，以往遙不可及的類神經網路演算法，從計算速度緩慢變得可行且精確，進而推行了許多深度學習在各領域的應用。現今已有許多針對物件偵測所設計的類神經網路，然而卻有一些場景亟待我們進行改進與優化。本章將說明本研究之背景與動機，而後根據動機訂定研究目標，最後說明本論文之架構。

1.1 研究背景與動機

目前物件偵測所設計的類神經網路有两大类，其中單階段演算法以 YOLO [7]、SSD [19]及 RetinaNet [18]為代表，而二階段演算法則以 R-CNN [15]及 Fast-RCNN [16]為代表。前者演算法同時進行物件檢測與分類，而後者先進行候選物件檢測而後進行分類。然而，在真實的應用場景中，我們需要快速偵測的網路模型來進行實時檢測，因此單階段演算法成為了現今物件偵測的主流，而 YOLO 系列得益於快速且兼具準確率表現的特性，獲得絕大多數使用者的青睞，不過在真實場景應用中仍有許多探索及改善的空間。

以真實的物件偵測場景來說，我們常常需要面對影像因為錄製者本身的晃動、鏡頭變焦、物體移動，甚至是場景內有霧氣所帶來影像模糊的問題，而現今的類神經網路模型在面對此類問題往往有力不從心的情況。如果我們將 YOLO 系列網路模型的代表（YOLOX [24]）應用於真實的動物辨識任務中，則會有圖 1-1 的辨識問題。



圖 1-1. 靜態物件偵測模型應用於模糊動物辨識的預測表現
(以 YOLOX 為代表)

從圖 1-1 中，可發現當動物移動速度過快時會產生模糊的現象，此時現有物件偵測模型即可能發生無法辨識的情況，因此我們需要一個能夠偵測動態物件的模型來處理此類問題。以現有的動態物件偵測模型來說，往往都是藉由單幀預測之結果來進行多幀校正，然而，若單幀預測的表現不盡人意時，那多幀校正的結果亦不理想，如圖 1-2 所示。



圖 1-2. 動態物件偵測模型應用於模糊動物辨識的預測表現
(以 YOLOV [22] 為代表)



綜上所述，可以發現當今的物件偵測模型中，雖然解決了過往偵測速度緩慢的問題，但在真實的視訊物件偵測中仍有許多探索空間。尤其是在模糊視訊物件偵測的任務中，一來可能偵測不到模糊物件、二來就算偵測到物件，也會因為前後進行多幀校正，而導致最終預測結果產生錯誤。

1.2 研究目標

了解當今物件偵測模型在實際應用中所遇到的挑戰後，我們認為處理視訊中物件移動、錄製者本身的晃動、鏡頭變焦等因素（具體類別如表 1-1 所示），所帶來的模糊影像為目前亟需解決的議題。

表 1-1. 視訊中模糊物件的種類

模糊程度	模糊種類
全局模糊	鏡頭移動
	鏡頭失焦
	鏡頭變焦
	鏡頭抖動
局部模糊	物體移動

可以發現即便針對視訊所設計的動態物件偵測模型，仍在模糊物件的預測中有改進的空間。因此根據前述議題，本研究希望達成以下目標：

(1) 提升物件偵測模型應用於模糊物件偵測的準確度

現有物件偵測模型在模糊物件偵測上的表現欠佳，如果不依靠額外模糊資料集進行訓練，則會因為單幀偵測品質不佳，而影響多幀偵測的成效。本研究希望使用額外的前處理方法（Data Argumentation），來生成一些人工模糊的影像以供模型進行訓練，從而改善現有物件偵測模型在偵測模糊物件時，辨識不精準或無法辨識的問題。

(2) 建立泛用的模糊物件偵測法

希望在解決現有模糊物件辨識問題的同時，所提出的方法又具有一定程度的泛用性，能與現今的通用物件偵測模型相容或直接套用。這樣在未來推出新的通用

物件偵測模型時，也能將此方法套用在新的物件偵測模型上，來提升整體的預測表現。



(3) 能與現有動態物件偵測模型進行整合

將本研究所提出之模糊物件辨識方法與當今動態物件偵測模型進行結合，在解決動態物件偵測模型的已知缺陷後，提高模糊物件的辨識準確度。

本研究採用 YOLO 系列中成熟且具有代表性的 YOLOX 網路模型，做為基礎架構進行模糊物件預測的改善。在改善方法且兼具通用性的同時，將與現有的動態物件偵測模型 YOLOV 進行整合，來克服幀與幀之間相互影響的問題，以實現單幀與多幀情境下均能進行穩定預測的強健性類神經網路。

1.3 論文架構

本研究基於上述背景、動機及目標，將研究內容分為五章於論文中進行介紹。首先在第一章緒論，介紹目前視訊物件偵測之發展背景、待改進之議題及針對待改進議題所訂定之研究目標。第二章為相關研究，將探討過去與本研究相關的技術與類神經網路的發展。第三章為研究方法，先根據研究目標來定義研究問題，而後根據研究問題延伸至單幀情境下靜態與動態影像的辨識任務，最後以多幀情境下動態影像的辨識任務進行收斂，並說明研究範圍及限制。第四章為實驗結果及討論，根據第三章所訂定之研究方法進行實驗、比較及探討。第五章為結論及未來展望，總結本研究成果的同時，提出未來發展的建議。

第二章 相關研究



本研究目標為在單幀與多幀情境下，均能進行穩定預測的強健性類神經網路。在單幀之靜態影像偵測部分，我們希望能有一個泛用且預測表現佳的基礎網路模型以供我們進行改進，故於 2.1 節回顧以往使用單幀方法進行物件偵測的網路模型。而在多幀之動態影像偵測上，需要一個預測表現佳且具有相容性，易於套用在單幀偵測網路模型的方法，以達成泛用之成效，故在 2.2 節檢視與視訊物件偵測相關之網路模型。而為了解決模糊物件的偵測問題，也於 2.3 節討論了現有模糊物件偵測所使用的方法，最後於 2.4 節進行簡單總結。

2.1 單幀靜態物件偵測

傳統單幀物件偵測的網路模型主要分為二階段與單階段的偵測方法。Ross 等人[15]在早期提出 R-CNN 網路架構，將辨識的流程分為候選區域生成及特徵提取二步驟。由於 R-CNN 在每次進行區域辨識時，均需要重新進行一次耗時的卷積計算，故 Ross 而後提出了新的 Fast R-CNN 方法[16]，解決了重複計算卷積層所帶來的耗時問題。然而，由於二階段的偵測方法仍過於耗時，在實際的應用中難以做到實時偵測，故 Joseph 等人[7]於同年提出了新的辨識方法，稱之為 YOLO (You Only Look Once)，將圖片拆解成許多網格進行偵測。此方法把二階段偵測簡化為單階段偵測，來加速物件辨識流程，與此同時又有不錯的預測成效。而 Wei 等人[19]也基於 YOLO 的基礎發展出 SSD (Single Shot MultiBox Detector) 網路架構，在多個卷積層所生成的特徵圖中，進行多次預測，而後再利用 Non-Maximum Suppression (NMS) 方法，挑選出合適的物件框，進行最終的物件預測 (其與 YOLO 的異同之處，如圖 2-1 所示。SSD 會將每個卷積層所輸出的圖像也一併進行偵測，YOLO 則無)。至此，由於預測速度與表現均達到可用水準，物件偵測相關的網路模型正式迎來一波發展的熱潮。

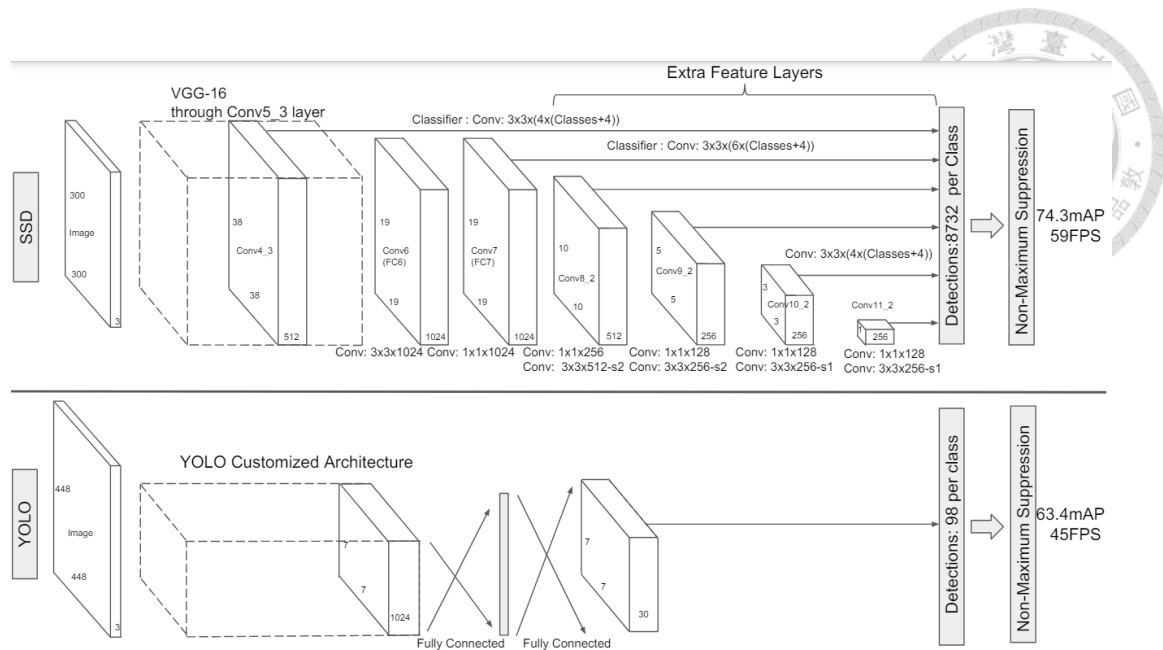


圖 2-1. YOLO 與 SSD 的網路模型架構 [19]

在發展單階段偵測網路模型的過程中，又分為基於錨點（Anchor based）與無錨點（Anchor free）兩類的偵測演算法。前者依據預先定義的錨點框進行訓練，將這些錨點調整到匹配物體的大小及長寬比。後者則不預先定義錨點框，而是讓網路模型直接預測物件框的中心以及長寬。Anchor based 的代表為 YOLO v2 [8]、YOLO v3 [9]、YOLO v4 [1]、YOLO v5 等，而 Anchor free 的代表為 YOLO [7]、YOLOX [24]及後續新的 YOLO 模型。

摒棄 Anchor based 回歸 Anchor free 的 YOLOX 作為新的 YOLO 後繼者，在後端網路有了顯著的改進。他們將傳統的 Coupled Head 改為 Decoupled Head（如圖 2-2 所示），也就是把分類和迴歸任務分開計算，帶來了收斂更快及準確率更高的表現，如圖 2-3 所示。在此模型下的 Anchor free 設計迎來了更好的預測表現，因此可知在不同的設計下 Anchor based 與 Anchor free 的優劣可能有所差異。

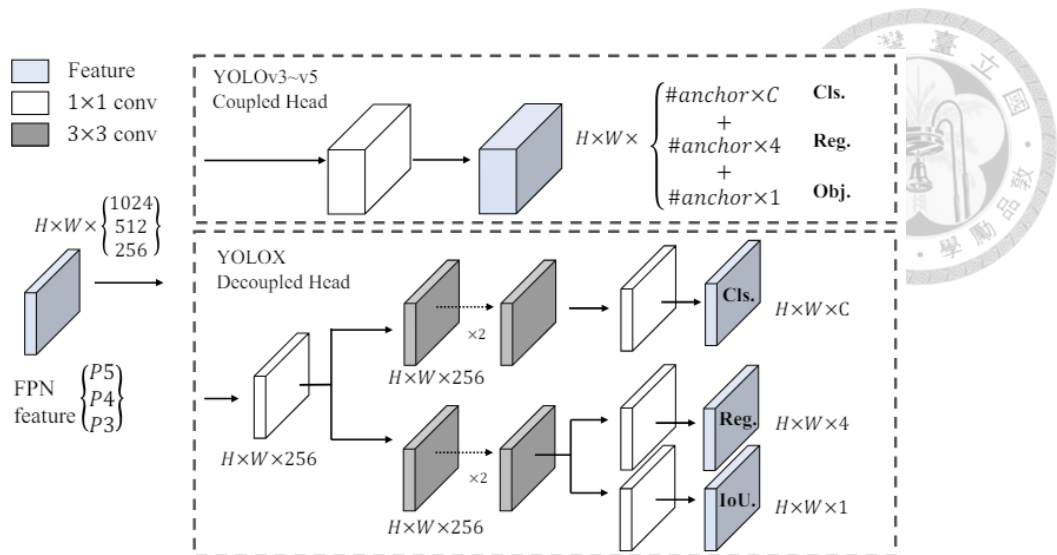


圖 2-2. YOLOX 的 Decoupled Head 設計 [24]

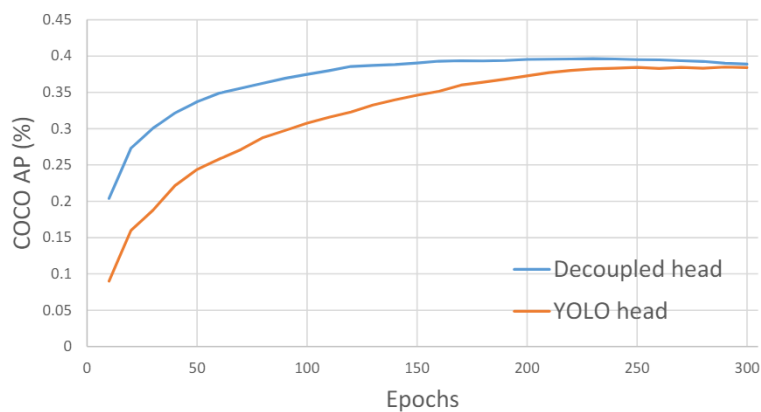


圖 2-3. 採用 Decoupled Head 的 YOLOX 預測表現 [24]

而後我們檢視 YOLOv6 [3]、YOLOv7 [4]及 YOLOv8，發現均採用了 Anchor free 加上 Decoupled Head 的設計方法，他們的做法都是針對損失函數、骨架 (backbone) 及 Head 等進行微調，來踏出 YOLO 系列的最後一哩路。

2.2 視訊物件偵測

在應用於靜態物件偵測時，前小節所提及之單幀靜態物件偵測模型均達到不錯的預測表現，但將其應用於視訊物件偵測時，則因視訊物件偵測中的畫面狀況更為複雜 (額外包含了物件遮蔽、晃動及失焦等，所造成的辨識干擾因素)，導致偵測效果不佳。因此在視訊物件偵測中，往往會關注如何利用多幀影像的資訊來進行



物件辨識的改進。

Xizhou 等人[20]提出了利用光流演算法，進行額外的影像前處理過濾，來計算多幀的光流圖像平均以輔助偵測物件。然而，此方法因為額外進行前處理運算導致預測速度變慢。Alberto 等人[2]在辨識完成後，根據多幀物件框的位置、幾何、外觀及語意進行相似性評分與校正，由於仰賴後處理的方式來進行多幀的預測改進，此方法雖然顯著改進了單幀的預測成效，但無法應用於實時的預測情境中。Yihong 等人[21]以 Mask R-CNN [10]為基礎網路模型進行改進，提出了 Memory Enhanced Global-Local Aggregation (MEGA) 模型，其網路架構如圖 2-4 所示。

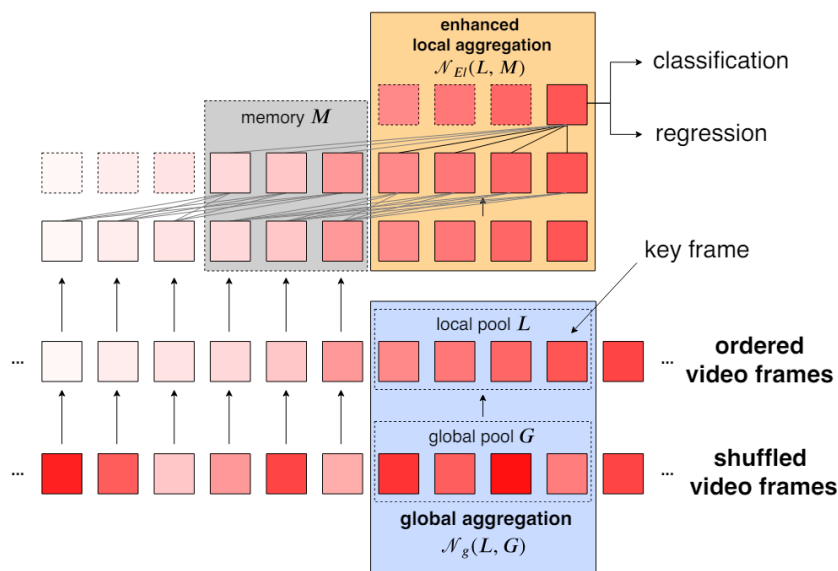


圖 2-4. 採用多幀參考進行預測的 MEGA 網路架構 [21]

圖 2-4 中可看到 MEGA 的網路架構在偵測階段一次擷取多幀資訊，在 backbone 和 RPN (Region Proposal Network) 模塊算出候選框後，透過 memory 與 local aggregation 模塊整合全局與局部的資訊進行預測，使得模型不只能借助單幀的資訊進行預測，也能將多幀的資訊一併考慮進來。然而，由於擷取大量的前後幀資訊，而使得後段的網路層也一併變得龐大，導致預測速度也較不理想。為解決此類問題，Yuheng 等人[22]利用單幀網路模型 YOLOX 為基礎進行改進，將 YOLOX 的後段



接上注意力機制模塊(如圖 2-5 所示),產生了針對視訊物件偵測所優化的 YOLOV, 在模型參數量不增加太多的同時,又能讓模型參考多幀的預測結果進行視訊物件預測。

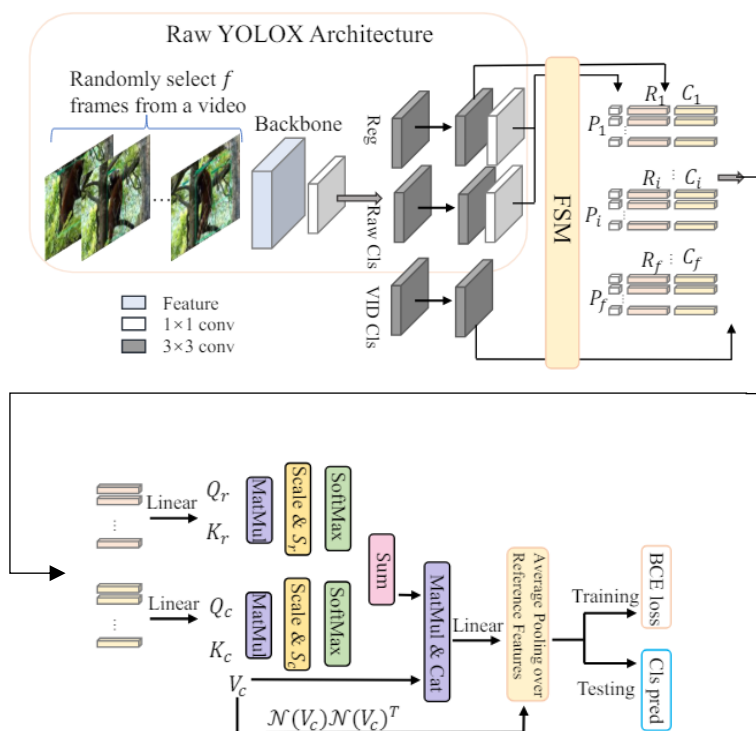


圖 2-5. YOLOV 之網路架構 [22]

隨著近年來語言模型的發展,人們也嘗試將語言模型中的 Transformer 模塊應用於動態物件偵測,以簡化動態物件偵測網路所需的額外網路架構。Lu 等人[11]利用 Transformer 模塊簡化動態物件偵測網路的後段網路架構,有效地消除了許多用於特徵聚合的額外網路層,例如光流、遞迴神經網路及關係網路。然而,此類方法應用於動態物件偵測中,與 YOLO 系列加上額外網路架構的表現並無顯著差異,仍為尚待探索的階段。

2.3 模糊物件偵測

基於模糊物件偵測的研究有二類:一類是基於多幀資訊進行物件偵測的校正,如 2.2 節所提及的方法;二類是將模糊物件去模糊化處理,Shen 等人[17]利用生成對抗網路 (GAN) 將模糊影像重建為清晰影像以進行訓練與預測。王家瑜[25]在靜

態數據集 GoPro 中，使用了 DeblurGANv2 [12] 的去模糊方法，使得模型在輸入前獲得較為清晰的畫面來進行辨識。然而，偵測階段進行額外前處理的方法導致了額外耗時問題，且此類前處理方法仍會有一定程度的偽像（Artifacts）問題，如圖 2-6 所示。可以發現即使是去模糊處理的影像仍有一定的殘影以及失真的影像。



圖 2-6. DeblurGANv2 於 Kohler dataset 中的去模糊表現 [12]

（左為原始模糊影像、右為去模糊影像）

在上述二類模糊物件偵測的方法中，透過去模糊化的前處理方法進行模糊物件偵測，會使得預測速度降低及產生偽像，而參考多幀的方法又多會因為網路層的擴展、前處理或後處理等因素導致預測速度變慢。

2.4 小結

綜觀上述關於單幀靜態物件偵測、視訊物件偵測及模糊物件偵測的方法，可發現他們在各自的應用中均達到一定之成效。然而，單幀方法應用視訊物件中卻產生了不同的問題，而為了解決這些問題，又需要花費額外的時間來進行預測，或者可能因為單幀的預測品質不佳，導致多幀視訊物件偵測的辨識錯誤。

在模糊物件偵測中，又有一部份方法為加入去模糊演算法，來讓網路模型獲得較佳品質的影像進行預測，但也增加了預測時間。因此，本研究希望在訓練階段，使用資料前處理的方法生成模糊物件，讓模型可以針對模糊物件進行辨識，避免在預測階段花費額外的時間進行去模糊化處理。此外，也會提升單幀模糊物件的預測表現，並降低單幀的預測品質不佳而產生後續辨識錯誤的問題。

第三章 研究方法



為清楚說明本研究的研究方法，首先在 3.1 節將說明本研究的問題定義，在了解問題之後，於 3.2 節說明本研究之架構與設計概念。然而，由於研究主題涵蓋廣泛，泛用之結論不易獲得，因此在 3.3 節說明本研究的研究範圍與使用限制。以下將針對上述三個小節分別詳述之。

3.1 問題定義

回顧過往相關研究後，我們發現目前針對視訊物件進行訓練與預測的類神經網路模型有兩類主要缺陷。首先，在視訊中進行單幀物件辨識時，對於模糊物件的預測準確率較低，容易有第一幀預測正確而第二幀預測錯誤的情形發生，而物件模糊造成的原因，包含偵測物件突然快速移動所產生的物件模糊、鏡頭的失焦、攝影機突然晃動，以及物件與背景不夠分明等問題，進而導致的偵測錯誤；再者，就是當需要參考連續多幀的影像資料，來進行預測的類神經網路時，會有第一幀預測錯誤，造成第二幀預測錯誤的結果發生，這是因為第二幀偵測時，會參考第一幀的偵測結果，故會有幀與幀之間相互影響的現象。綜觀上述這兩類缺陷，我們認為過往研究利用多幀資訊來改進視訊物件的偵測固然正確，但也不能忽視單幀資訊對於視訊物件偵測的重要性。

因此，為了改進模糊視訊物件的偵測問題，我們認為應該要先改進單幀情境下的物件偵測準確率，以避免多幀參考偵測時，幀與幀間相互影響的問題，進而改善多幀情境下的物件偵測準確率。如此一來，不只可以讓類神經網路在視訊資訊不充分下（單幀情境），也可以確保一定的準確率，並可在後續進行多幀預測時提升準確率，以期達到泛用的效果。

根據上述提及的缺陷與改進需求，我們將問題定義為：已知單幀資訊下（由動態影片擷取之單幀靜態照片資訊），如何改善模糊物件偵測的準確率？並且將此方法與現有的多幀偵測參考方法做進一步的結合，來克服幀與幀間之相互影響，進而



實現單幀與多幀情境下，均能進行穩定預測的強健性類神經網路。

3.2 研究架構與設計

為了解資料特性，並訓練模型作後續的物件偵測，我們從 ImageNet 取得物件偵測資料集[13] (ImageNet Large Scale Visual Recognition Challenge 2017 Challenges II: Detection, ImageNet DET) 以及影片物件偵測資料集 (ImageNet Large Scale Visual Recognition Challenge 2017 Challenges III: Object detection from video, ImageNet VID) 作為研究資料集，其中 ImageNet DET 所涵蓋的資料為靜態照片，ImageNet VID 所涵蓋的資料為多幀的動態影片。

ImageNet DET 資料集包含了陸地生物、海洋生物以及交通工具等，有 200 類靜態的物件 (詳見附錄)，並且根據每類的物件比例、影像混亂程度及每張照片的物件數量，進行較為均勻且一致的篩選，其中各照片的標記會包含照片中物件的類別、物件框之高度和寬度、物件框左上角以及右下角的直角座標資訊；ImageNet VID 資料集的類別為 ImageNet DET 的子集合，有 30 類運動的物件 (表 3-1)，其篩選方式除了根據 ImageNet DET 的方式外，也會根據物件的運動類型進行篩選，其中各幀的標記會包含物件的類別、物件框之高度和寬度、物件框左上角以及右下角的直角座標資訊。

表 3-1. ImageNet VID 物件類別

飛機	羚羊	熊	自行車	鳥
巴士	汽車	牛	狗	貓
大象	狐狸	大熊貓	倉鼠	馬
獅子	蜥蜴	猴子	摩托車	兔子
紅熊貓	羊	蛇	松鼠	老虎
火車	龜	船	鯨	斑馬

在取得 ImageNet DET 與 ImageNet VID 資料集後，我們從 200 與 30 類物件類別中取交集，從而得到 30 類共通的物件類別，即 ImageNet VID 的物件類別。而為了專注於探討模糊視訊物件的辨識問題，我們將採用 YOLOX 與 YOLOV 模型，

其中 YOLOX 為傳統靜態物件偵測模型，YOLOV 為現今動態物件偵測模型，對 ImageNet DET 與 ImageNet VID 的 30 類物件資料集進行訓練，並使用交併比(IOU) 大於 50%的準確率 (Average Precision with Intersection over Union 50%，AP50) 做為評估指標，在 ImageNet VID 所預先給定之驗證集進行測試，來選取較具代表性的 10 類物件進行訓練、分析與討論，以了解現今傳統靜態物件偵測模型與動態物件偵測模型的不足之處。

為了改善單幀資訊下模糊物件的預測準確率，並與現有多幀的方法進行結合，我們在取得資料集後，搜尋現有應用於 ImageNet DET 與 ImageNet VID 資料集的模型，發現基於 YOLO 系列所衍伸的 YOLOV 模型，不只有良好的準確率表現，也相容於現有 YOLO 家族的模型。因此，在考量準確率以及泛用性後，我們選擇了 YOLOV 的基礎模型 YOLOX，來作為我們的模型調整基準，兩個模型在 ImageNet VID 的偵測表現如表 3-2 所示，其中 S、L、X 代表模型參數的大小 (由小至大)，可發現 YOLOV 在相同參數大小下，透過幀與幀間的參考，可以達到更準確的偵測率。而我們將使用參數較少的 YOLOX-S 來加速訓練流程，之後再將所調整的結果與 YOLOV-S 的測試結果進行比對，以了解兩模型的優點與不足之處。

表 3-2. YOLOX 與 YOLOV 在 ImageNet VID 的驗證集表現

Model	size	mAP@50val	Speed 2080Ti (batch size=1) (ms)
YOLOX-S	576	69.5	9.4
YOLOX-L	576	76.1	14.8
YOLOX-X	576	77.8	20.4
YOLOV-S	576	77.3	11.3
YOLOV-L	576	83.6	16.4
YOLOV-X	576	85.5	22.7

由於類神經網路的初始權重、讀取資料的順序與卷積層算法的順序，均受隨機亂數種子的影響，進而影響最終訓練的結果，而這些隨機的結果，會使我們不易判斷每次調整模型、前處理方法、損失函數和學習率等的影響成效。為了消除這些隨



機性的干擾，我們在訓練模型時，會先使用固定的亂數種子，來固定類神經網路的初始權重、讀取資料的順序，並且設定使用相同的卷積層算法順序，來訓練我們的模型，以確保每次訓練的一致性以及可復現性。

完成前述的準備階段後，實驗將以三個階段進行研究，分別為一、單幀情境下靜態影像中清晰物件的辨識，二、單幀情境下動態影像中清晰及模糊物件的辨識，及三、多幀情境下動態影像中清晰及模糊物件的辨識，整體研究架構，如圖 3-1 所示。

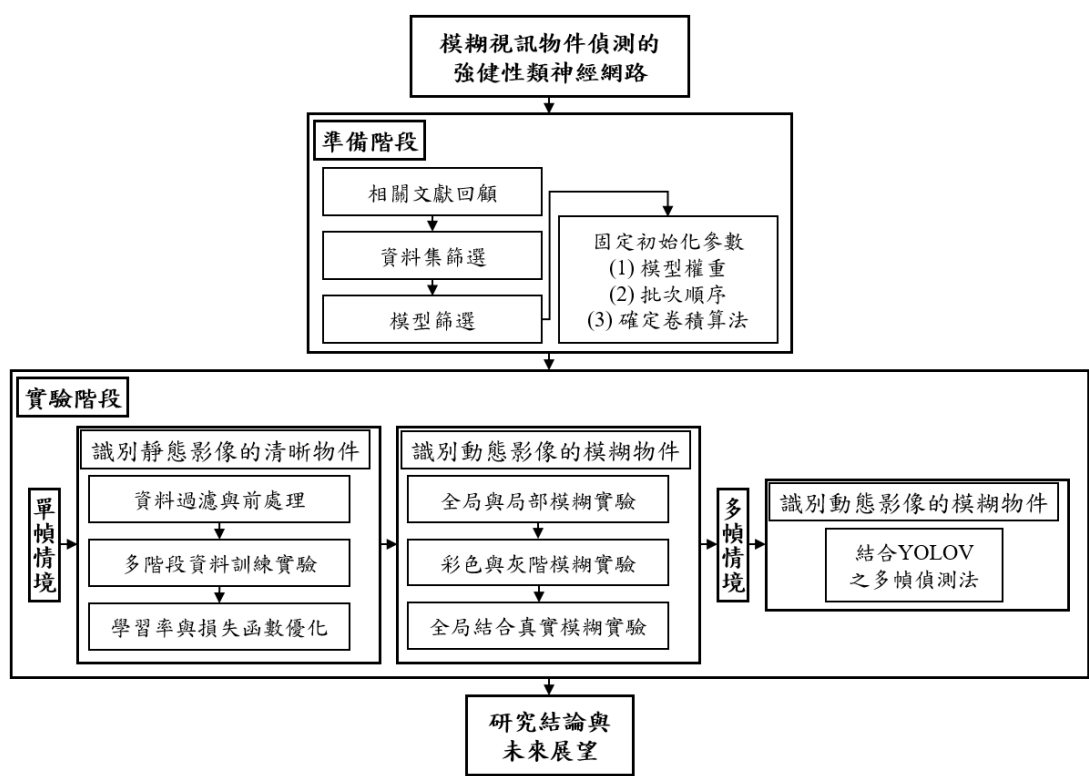


圖 3-1. 研究架構圖

3.2.1 單幀情境下識別靜態影像的清晰物件

首先，將從單幀靜態清晰物件開始進行資料過濾實驗。利用 ImageNet DET 所提供的靜態照片，來模擬現實生活中所遇到的單幀靜態清晰物件影像，我們會先針對 ImageNet DET 進行資料過濾，而後基於 YOLOX 所採用的 Mosaic (Alexey 等人[1]所提出，方法說明詳見圖 3-2) 與 Mixup (Hongyi 等人[5]所提出，方法說明詳



見圖 3-3) 方法進行 Mosaic 與 Mixup 的分離實驗。接著對訓練集進行不同程度的類別數量過濾。我們將會實驗全部資料訓練、去除較多資料訓練以及將資料類別物件數目平衡等三種程度的過濾方法，

透過前述步驟，我們可處理類別物件數目不平衡的問題，並且重新劃分訓練集、驗證集與測試集，以做為 AP50 的評分來源。同時，獲得模型訓練所需之資料臨界值來加速訓練過程。並且在 Mosaic 與 Mixup 的前處理實驗中了解分離操作是否會好於合併操作，得到分離操作的合適比例。

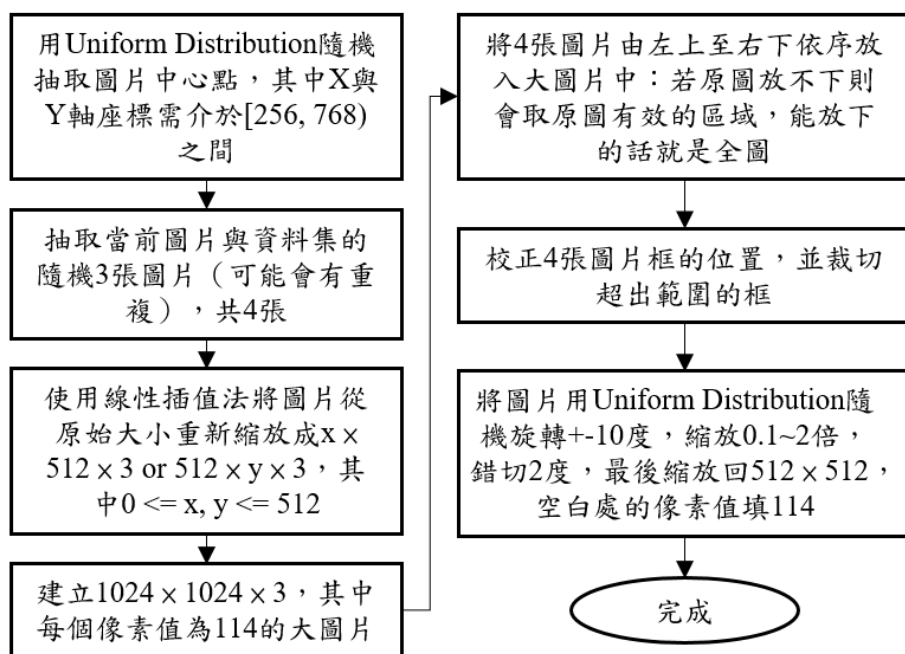


圖 3-2. YOLOX 所採用之 Mosaic 方法

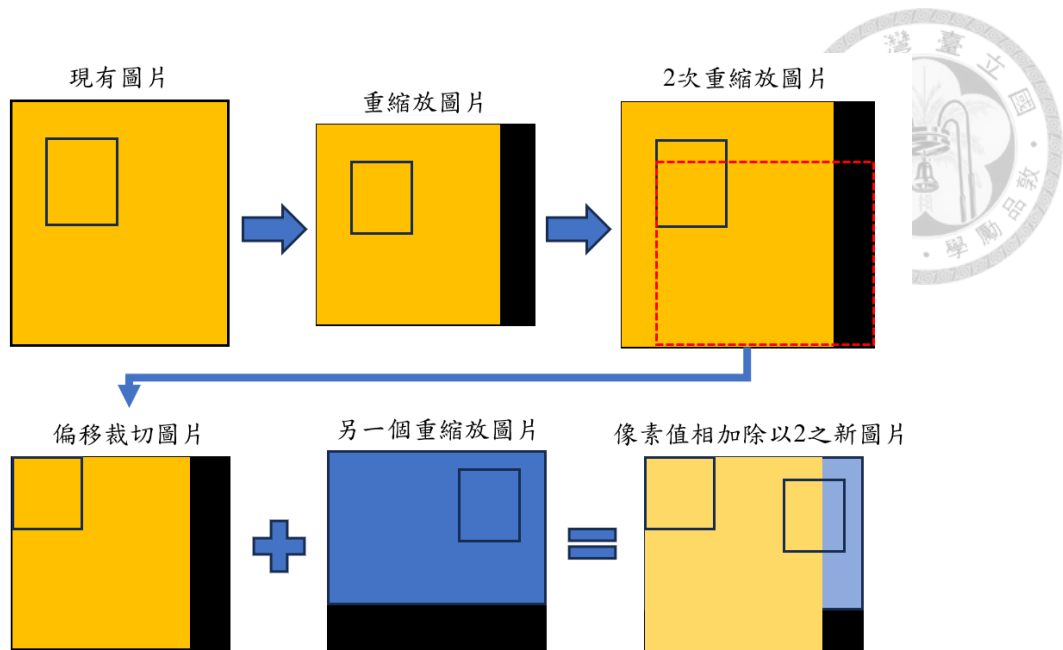


圖 3-3. YOLOX 所採用之 Mixup 方法

完成資料過濾實驗後，我們將進行多階段資料實驗，主要將分為：「一次訓練所有資料」對比「分兩批資料訓練」。進行此實驗的目的是我們想了解是否可以透過分拆資料來加速模型的訓練，使模型利用一部份的資料先快速的收斂到一個階段，然後再利用另一部份的資料來優化先前收斂所得的結果，最後能得到比一次訓練所有資料更好的準確率。

此做法與一般預訓練模型的不同之處，在於我們是用來源相同的資料集進行分批訓練，而非使用不同的資料集進行二次訓練。為了充分了解資料的較佳分拆方式，我們將進行兩個不同訓練模式，分別為「兩批資料為不重複分割的資料」對比「第一批資料為第二批資料的子集合」，以了解模型在第二階段訓練中，是否需要包含第一次的訓練資料，才能得到較佳的收斂結果。

接下來實驗將進行學習率與損失函數的優化，我們檢視 YOLOX 之後，發現他們採用的是 Warmup + Cosine 學習率策略。Warmup 的方法[14]是在一開始訓練模型時，逐步增加學習率的方式，來讓起初不了解資料的模型，能在前幾個迭代(epoch)可以有比較低的參數更新率以進行學習，在了解資料之後，再提升參數更新的速度，避免初始模型更新過快的問題。而 Cosine 學習率策略則是透過逐步下降學習率，



來讓模型在初步了解資料後，進行更精細的參數收斂。因此，我們會先進行 Cosine 學習率對比固定學習率與 Warmup 對比無 Warmup 的實驗，來了解何種方式更新參數，比較適用於我們的辨識任務。

損失函數的部分，YOLOX 採用的損失函數如式 3-1 所示：

$$\begin{aligned} Loss = & Reg\ Weight \times IOU\ Loss + Object\ Loss \\ & + Class\ Loss + L1\ Loss \end{aligned} \quad (3-1)$$

IOU Loss 採用傳統的 IOU 算法 (如式 3-2 所示)，其中 A 為正確之物件框，B 為預測之物件框。並於前面乘以自訂常數 (Reg Weight = 5)，來提升 IOU Loss 在整體損失函數的比重。

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (3-2)$$

物件框的 Object Loss 與類別的 Class Loss 採用二元交叉熵 (Binary Cross Entropy)，對每個物件框進行評分，如式 3-3 所示。其中 y 是二元標籤 1 或 0 (正確與錯誤)， $p(y_i)$ 是輸出屬於正確標籤的機率。

$$BCE\ Loss = -\frac{1}{n} \sum_{i=1}^n y_i \times \log(p(y_i)) + (1 - y_i) \times \log(1 - p(y_i)) \quad (3-3)$$

L1 Loss 則是根據物件框的 x, y 座標值進行計算，並加總進行總損失函數的評分 (數值越小越好)，如式 3-4 所示，其中 $f(x_i)$ 為預測之座標值， y_i 為正確座標值。

$$L1\ Loss(x, y) = -\frac{1}{n} \sum_{i=1}^n |y_i - f(x_i)| \quad (3-4)$$

了解 YOLOX 的損失函數設計原理後，我們也將嘗試對 YOLOX 的損失函數進行改進，讓評分方式更接近應用資料所需要的類型。在此我們設計了三種實驗：

自訂常數 (Reg Weight) 的調整、IOU Loss 以及 Object Loss 的細部優化，來提升模型的預測效果。

自訂常數的調整我們將改變常數的大小，來了解模糊物件偵測所需的最佳常數。IOU Loss 的優化我們將採用傳統的 IOU 算法比對 eIOU 算法[23]。eIOU 比 IOU 多計算了兩個框中心點的歐式距離、框的長寬差異，來更精準的評定物件框差異。Object Loss 的優化採用 Varifocal Loss [6]的方法，來解決物件數量不平衡的問題。

3.2.2 單幀情境下識別動態影像的模糊物件

在識別單幀情境之前，將先針對 ImageNet VID 驗證集的模糊物件進行分析，以瞭解模糊種類（物體移動、鏡頭移動、鏡頭失焦、鏡頭變焦或鏡頭抖動）以及模糊程度（全部或局部模糊）的物件數量與分佈。

完成驗證集的分析並檢視分佈後，將從原始的 ImageNet VID 驗證集中，挑選模糊影片以作為本階段的驗證資料集，並從原始的 ImageNet VID 隨機挑選部分幀，做為模型綜合表現的評估。

接下來將從靜態照片物件辨識開始，以 ImageNet DET 所提供的靜態影像作為訓練集，再使用 ImageNet VID 驗證集所挑選出的子資料集作為驗證，以探索使用靜態影像預測動態影像，所能達到的最佳表現。此實驗的好處是我們的模型不僅在靜態影像下有最佳的模型表現，也能在新的動態影像有較佳的預測成效，實現模型的強健性。

本階段我們會先從全局模糊影像著手，利用 3.2.1 節所得到的最佳方法加入高斯模糊法（其數學表示如式 3-2 所示，其中 $G(x,y)$ 為高斯模糊的分布函數、 $x^2 + y^2$ 為高斯核之模糊半徑、 σ 為常態分布的標準偏差），以模擬鏡頭晃動、鏡頭失焦及鏡頭移動等類型的模糊問題。在資料前處理階段，對部分影像進行模糊化處理，而模糊化的比例將依據前述所提到的 ImageNet VID 驗證集的資料分析結果，盡量接近實際場景下模糊物件在動態影像中的占比，期望所訓練的模型能在真實動態影像



辨識中，對模糊物件能有較好的預測效果。此階段將分為全局結合局部模糊、彩色與灰階模糊對比及全局結合真實模糊三步驟，逐步針對動態影像的模糊物件進行模型優化，達到較佳的綜合表現。

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (3-2)$$

在全局結合局部模糊實驗中，我們會先進行不同模糊比例的實驗，以找到合適的全局模糊與局部模糊比例，模擬鏡頭晃動結合物體移動等類型的模糊問題。由於我們知道高斯模糊的模糊程度取決於高斯核（Gaussian kernel）的大小，而 Sigma 值又決定 Gaussian kernel 的大小，因此會採用比較貼近真實模糊的 Sigma 值進行實驗，以找到模糊物件偵測所合適的較佳解。

在本實驗階段中，我們先檢視了 ImageNet VID 驗證集中的模糊物件，如圖 3-4 所示。可發現在不同幀之間，模糊的程度均有所變化。由於物件的模糊程度會是一個連續性而非離散性的變化，因此我們設定了一個模糊區間範圍來對模糊物件進行不同程度的模擬，經測試後我們發現 Sigma 值介於 4 到 8 之間，有較貼近真實模糊物件的表現，如圖 3-5 所示。故實驗時會先隨機抽取 4 到 8 之間的 Sigma 值，而後再對物件進行模糊化處理。



圖 3-4. 同影片中貓與狐狸的不同模糊表現
(ImageNet VID 驗證集編號：00037001)



圖 3-5. 不同 Sigma 值之高斯模糊表現 (由左而右分別為 2、4 及 8)

完成全局模糊的初步實驗後，我們將根據前階段所分析的物件模糊程度(全部或局部模糊)數量分佈，來進行全局結合局部模糊的綜合前處理實驗，以達到更好的預測成效。局部模糊的處理方式，會根據訓練集中所提供的物件框位置，來對影像進行局部模糊，若影像中有多個物件，則會對影像進行多次的局部模糊，對每個物件均會抽取一次介於 4 到 8 之間的隨機 Sigma 值來進行模糊，如圖 3-6 所示。

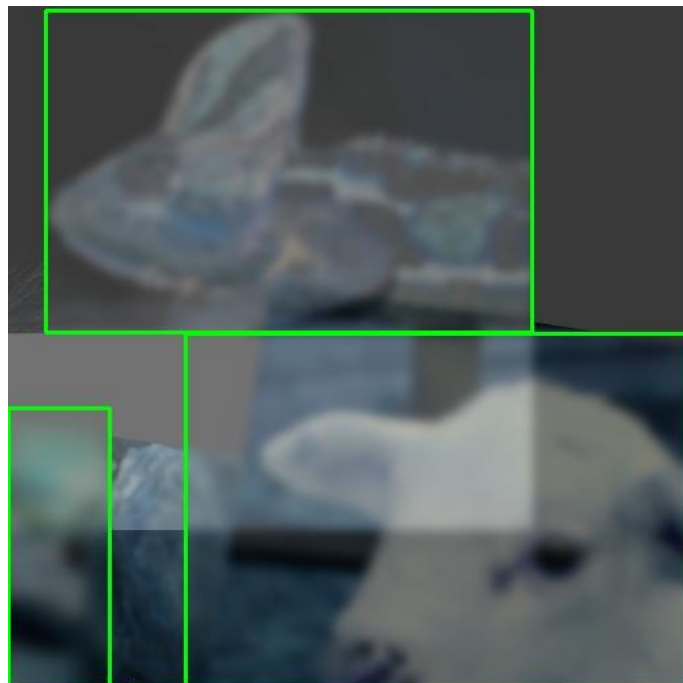
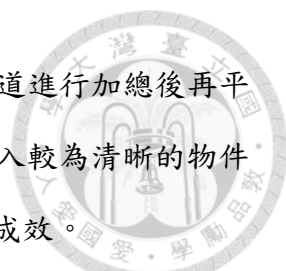


圖 3-6. 使用局部高斯模糊的影像前處理結果

接下來，為了解模糊物件辨識模型對於色彩的敏感度，我們對模糊影像進行彩



色前處理與灰階前處理之比較。我們會將原始影像之 RGB 三通道進行加總後再平均，取代原始影像之色彩數值，達成灰階前處理之效果，並且加入較為清晰的物件 (Sigma 值介於 2 到 4 之間)，來比較模糊物件辨識模型的預測成效。

同時也好奇如果混合彩色與灰階模糊物件一起進行訓練，能否讓模型透過更多元的訓練資料，達成更好的預測成效呢？因此在本階段也會進行對比，而具體的模糊比例，會根據前一階段所找到的合適模糊物件比例，進行彩色與灰階的對半分配來進行混合實驗。

為了避免單一參數所造成的實驗偏差，我們使用 3 個隨機種子製造三種不同的模型初始參數，來進行多次實驗，驗證彩色及灰階模糊的最佳超參數，以提升本階段實驗最佳結果的可信度。

最後會將前階段所得之最佳方法，利用 ImageNet DET 訓練集結合 ImageNet VID 訓練集進行訓練，讓訓練資料包含清晰影像、高斯模糊影像及真實模糊影像。我們會使用在 COCO 資料集上所預先訓練好的模型參數，進行遷移學習，來加速訓練過程，進而得到本階段的最佳預測結果。

3.2.3 多幀情境下識別動態影像的模糊物件

在優化單幀情境下模糊物件的偵測後，接下來把實驗推進到多幀情境下的動態影像辨識實驗。在此將使用 YOLOV 的多幀訓練方法與我們的訓練結果結合，其訓練流程如圖 3-7 所示。

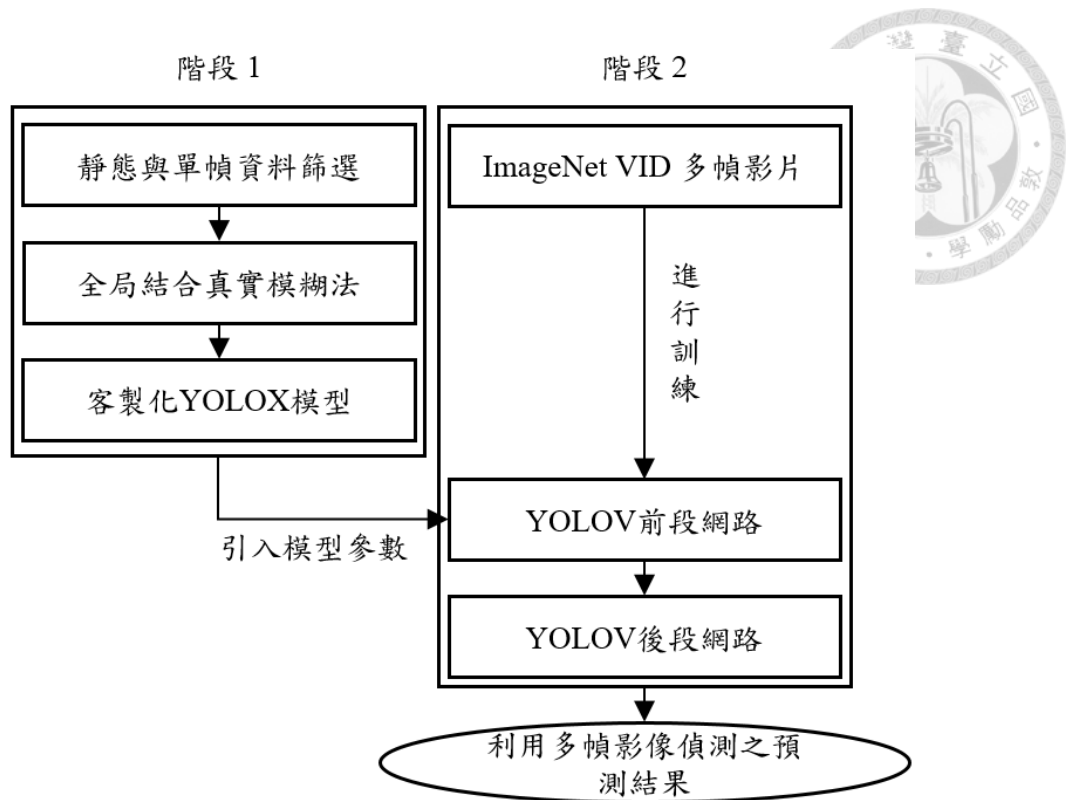


圖 3-7. 利用多幀影像偵測模糊物件類神經網路訓練流程

階段 1 我們會使用 3.2.2 節所得之單幀物件訓練法，來得到單幀情境下之較佳模糊物件偵測網路。由於我們的網路架構與 YOLOV 的前段部分相同，故能直接將訓練所得之模型參數引入 YOLOV 之前段模型，達到兼容功能。階段 2 由於 YOLOV 需要透過幀與幀之間物件的關聯性來預測物件類別，故使用了 ImageNet VID 所提供之多幀影片訓練 YOLOV 的後段網路（此訓練流程不會更新前段網路參數），期望透過我們的前段網路克服幀與幀間相互影響的問題，並結合 YOLOV 之多幀參考的優點，達到多幀情境下模糊物件偵測的最佳預測結果。本研究之網路架構圖如圖 3-8 所示。

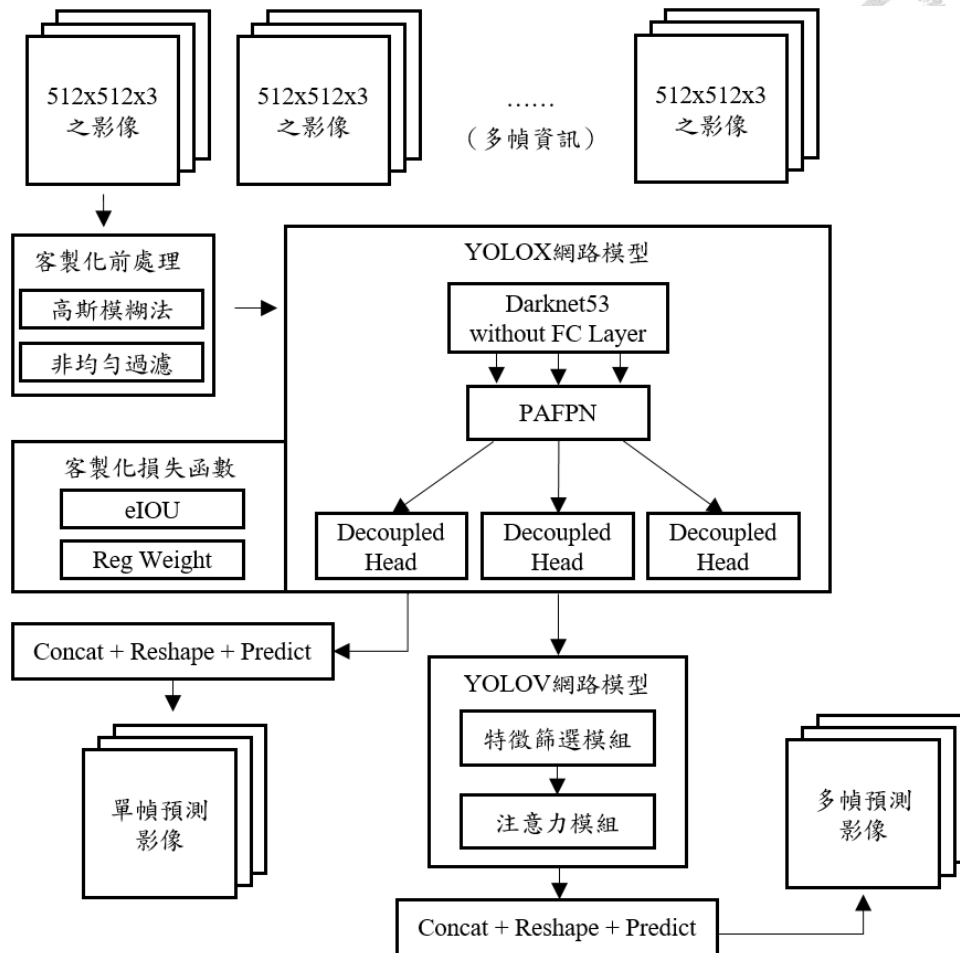


圖 3-8. 實驗網路架構圖

3.3 研究範圍與限制

本研究之模糊視訊物件為室外或室內環境下，進行 10 類動物物件之辨識。在室外及室內環境下，會盡量要求影像能有充足的亮度或對比度，以利我們進行物件輪廓的偵測辨識，且在大多數幀中，動物均在畫面內，以利後續的多幀偵測。此 10 類動物將取形體較為相異的個體，以利我們集中探討模糊物件的偵測問題，而非困難的分類問題。

在類神經網路模型的選擇上，選用目前較具主流的 YOLO 家族模型，進行客製化調整，以契合模糊視訊物件辨識的需求。如此一來，也方便未來將此方法套用在其他 YOLO 系列的模型上，達成較佳的兼容性。本研究所有的訓練結果，均使用固定種子碼將模型參數、前處理及資料讀取順序固定，以利我們在相同條件下進

行實驗對比，並確保可復現性。

模糊視訊物件偵測的現有研究中，多為利用多幀間的物件影像參考，來校正預測結果，然而在各幀之間的物件均無法進行辨識時，可能會導致無法進行單幀的物件偵測，進而無法利用多幀之間的參考來校正預測結果，因此本研究於多幀偵測階段所進行之模糊物件偵測，會至少確保有部分清晰畫面，才會進行多幀動態之物件偵測。

而模糊影像所討論的範圍僅限於拍攝中所帶來的模糊問題，即物件移動、錄製者本身的晃動、鏡頭變焦以及環境背景的霧氣。在視訊壓縮過程中，如 MP4 編碼，所帶來的模糊問題則不予討論。

第四章 實驗結果與討論



根據第三章所述之研究方法進行實驗，其結果將分別在下列章節中說明並討論。實驗所使用之硬體設備與軟體開發環境，如表 4-1 所示。

表 4-1. 硬體設備與軟體開發環境

硬體設備	
CPU	Intel Core i7-11700 (8 core / 16 threads, single core 4.9 GHz / all core 4.4 GHz)
GPU	NVIDIA RTX A6000 (10752 CUDA cores, 1.8 GHz, 48GB GDDR6 memory)
記憶體	DDR4-3200 128GB (32GB x4, Dual channel mode)
硬碟	ADATA SX8200 Pro 2TB (PCIe Gen3x4, M.2 2280, 3D TLC SSD)
軟體開發環境	
作業系統	Ubuntu 20.04.5 LTS
CUDA Toolkit	11.1.0
Python	3.8.10
PyTorch	1.9.0

4.1 資料集篩選

為了解資料的特性以做後續的篩選，我們先從 ImageNet DET 與 ImageNet VID 資料集中，挑選出 30 類共通的類別，然後分別對 30 類的 ImageNet DET 與 ImageNet VID 進行資料分布的檢視，其結果分別如表 4-2 及表 4-3 所示。



表 4-2. ImageNet DET 30 類物件總數量

(共 185410 張照片，214643 個物件)

0: 飛機	1: 羚羊	2: 熊	3: 自行車	4: 鳥
1888	2883	3254	2162	41516
5: 巴士	6: 汽車	7: 牛	8: 狗	9: 貓
3248	12023	1523	78805	3700
10: 大象	11: 狐狸	12: 大熊貓	13: 倉鼠	14: 馬
2250	2862	1008	907	2546
15: 獅子	16: 蜥蜴	17: 猴子	18: 摩托車	19: 兔子
1040	6268	9362	2891	2311
20: 紅熊貓	21: 羊	22: 蛇	23: 松鼠	24: 老虎
1078	2032	9213	965	1254
25: 火車	26: 龜	27: 船	28: 鯨	29: 斑馬
1464	3215	10109	1508	1358

表 4-3. ImageNet VID 30 類物件總數量

(共 1298523 張照片，2005418 個物件)

0: 飛機	1: 羚羊	2: 熊	3: 自行車	4: 鳥
112454	67370	61683	43751	138805
5: 巴士	6: 汽車	7: 牛	8: 狗	9: 貓
36499	141261	64408	149887	70528
10: 大象	11: 狐狸	12: 大熊貓	13: 倉鼠	14: 馬
96012	46278	60577	44062	62253
15: 獅子	16: 蜥蜴	17: 猴子	18: 摩托車	19: 兔子
35036	37789	81463	35595	45075
20: 紅熊貓	21: 羊	22: 蛇	23: 松鼠	24: 老虎
49084	40310	38521	58979	23676
25: 火車	26: 龜	27: 船	28: 鯨	29: 斑馬
118485	50334	64863	46529	83851

從上表資料中，我們發現 ImageNet VID 的平均物件數量約比 DET 資料集多了約 10 倍（185410 張照片 vs. 1298523 張照片），因此我們對 ImageNet VID 中所預先分配好的訓練集進行了 10% 的隨機挑選，以平衡 ImageNet DET 與 ImageNet



VID 的訓練物件數量，挑選結果如表 4-4 所示。

表 4-4. ImageNet VID 10% 訓練集物件數量列表
(共 108613 張照片，173596 個物件)

0: 飛機	1: 羚羊	2: 熊	3: 自行車	4: 鳥
8677	5926	5185	3453	12979
5: 巴士	6: 汽車	7: 牛	8: 狗	9: 貓
3106	11396	5373	12958	5889
10: 大象	11: 狐狸	12: 大熊貓	13: 倉鼠	14: 馬
8510	3729	5304	3770	5508
15: 獅子	16: 蜥蜴	17: 猴子	18: 摩托車	19: 兔子
3300	3180	7057	3408	3887
20: 紅熊貓	21: 羊	22: 蛇	23: 松鼠	24: 老虎
4728	3591	3221	4721	2134
25: 火車	26: 龜	27: 船	28: 鯨	29: 斑馬
10610	4504	6038	3859	7595

完成 ImageNet DET 與 ImageNet VID 的資料集篩選後，我們將 ImageNet DET 與 ImageNet VID 10%訓練集合併，用以訓練 YOLOX 與 YOLOV 模型，並利用在 COCO 資料集上所預先訓練好的權重，作為模型初始參數，來加速訓練並控制變量，最後在 ImageNet VID 驗證集上，測試所得各類別 AP50，其結果如表 4-5 所示，並以長條圖（圖 4-1）結果進行對比，尋找較具代表性的物件類別。

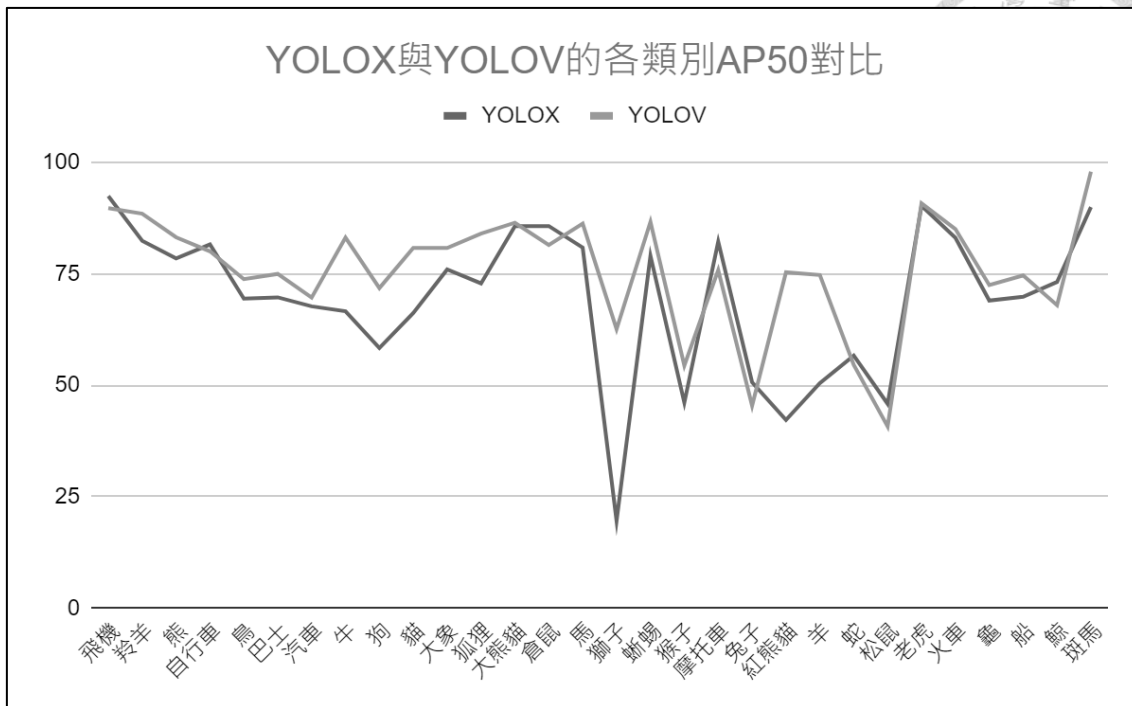


圖 4-1. YOLOX 與 YOLOV 的各類別 AP50 對比

表 4-5. YOLOX 與 YOLOV 在 ImageNet VID 驗證集的各類別 AP50 表現

(取小數點後 2 位四捨五入)

物件類別	YOLOX	YOLOV	物件類別	YOLOX	YOLOV
0: 飛機	92.38	89.66	15: 獅子	19.6	62.54
1: 羚羊	82.34	88.41	16: 蜥蜴	78.98	86.6
2: 熊	78.38	83.13	17: 猴子	46.07	54.38
3: 自行車	81.53	79.99	18: 摩托車	82.16	75.77
4: 鳥	69.39	73.75	19: 兔子	50.65	45.38
5: 巴士	69.67	74.92	20: 紅熊貓	42.18	75.3
6: 汽車	67.66	69.61	21: 羊	50.44	74.68
7: 牛	66.56	83.06	22: 蛇	56.61	54.54
8: 狗	58.32	71.75	23: 松鼠	45.81	40.74
9: 貓	66.16	80.76	24: 老虎	90.17	90.77
10: 大象	75.92	80.75	25: 火車	83.01	84.98
11: 狐狸	72.79	83.97	26: 龜	68.96	72.41
12: 大熊貓	85.64	86.38	27: 船	69.79	74.57
13: 倉鼠	85.65	81.39	28: 鯨	73.11	67.92
14: 馬	80.82	86.18	29: 斑馬	89.9	97.86

在折線圖對比中，我們發現 YOLOX 和 YOLOV 在牛、狗、貓、狐狸、獅子、蜥蜴、猴子、摩托車、紅熊貓、羊、斑馬等 10 類的 AP50 表現上差異較大，而其餘類別兩者之 AP50 差異較不明顯。為了集中探討模糊視訊物件的辨識問題，我們首先在差異較大的類別中，挑選較為動態的物件類別進行討論—即陸上動物，此時我們得到 9 類（摩托車除外）。接著為比對模糊程度影響的因素，從 YOLOV 的結果對比 YOLOX 的結果，其中 AP50 提升較小或變差的類別中，隨機挑選 3 類（挑選結果為大熊貓、兔子、蛇），以檢視現今動態物件偵測模型所無法改進之處。最後與前述 9 類動態類別中的 7 類（隨機挑選）進行合併，最終挑選之 10 個物件類別，如表 4-6 所示。以此 10 個類別進行後續的訓練、分析與討論。

表 4-6. 本研究探討之 10 類物件

(1) 牛	(2) 狗	(3) 貓	(4) 狐狸	(5) 大熊貓
				
(6) 獅子	(7) 蜥蜴	(8) 兔子	(9) 羊	(10) 蛇
				

4.2 單幀情境下識別靜態影像的清晰物件

在劃分訓練集、驗證集與測試集之前，我們先檢視 ImageNet DET 官方所預先分配的訓練集與驗證集各物件的數量，如表 4-7 和表 4-8 所示。



表 4-7. ImageNet DET 訓練集 10 類物件之數量

(共 93751 張照片，102725 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
1366	74517	3514	2681	962
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
1000	5912	2194	1883	8696

表 4-8. ImageNet DET 驗證集 10 類物件之數量

(共 4307 張照片，6037 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
157	4288	186	181	46
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
40	356	117	149	517

在表 4-7 及表 4-8 中，可發現狗的物件數量與其他類別物件的數量有嚴重不均衡的現象，且在驗證集的部分，大熊貓與獅子只有不到 100 個物件可供驗證，這會使驗證時的樣本過少，進而造成 AP50 的可信度降低。為了解決這些問題，首先將 ImageNet DET 官方所預先分配的訓練集與驗證集同類別物件進行合併，再採用隨機亂數調整照片順序，來確保後續抽取樣本的隨機性，接著採取 55%:15%:30% 的照片數量比例，分別切分新的訓練集、驗證集與測試集資料。最後為了避免驗證時狗的物件數量過多，導致 AP50 總和被影響。由於每張照片的物件數量不同，為了方便進行挑選，我們以照片數量為基準，隨機抽取每類最多 1000 張照片來作為新的驗證集、隨機抽取每類最多 2000 張照片來作為新的測試集，新的資料分布如表 4-9、表 4-10 及表 4-11 所示。

表 4-9. 本研究之 ImageNet DET 訓練集 10 類物件之數量

(共 53928 張照片，59766 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
839	43297	2025	1571	554
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
580	3444	1262	1136	5058

表 4-10. 本研究之 ImageNet DET 驗證集 10 類物件之數量

(共 4775 張照片，5264 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
244	1130	553	436	147
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
154	941	350	292	1019

表 4-11. 本研究之 ImageNet DET 測試集 10 類物件之數量

(共 9553 張照片，10475 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
433	2234	1118	854	307
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
306	1883	699	604	2037

此階段完成後，可發現我們在訓練集中並未對狗的物件數量過多進行過濾，這是因為蜥蜴與蛇之物件數量比其他 7 類物件也多出不少，導致我們不確定是否要對這 2 類物件也進行過濾，因此接下來進行資料前處理的實驗，以決定是否要對物件數量過多的類別進行過濾。本實驗所使用的共同超參數，如表 4-12 所示。



表 4-12. 本實驗之共同超參數

訓練超參數	
Input size	512x512
Batch size	32
Warmup epoch	5
Optimizer	SGD
Weight decay	0.0005
Momentum	0.9
Dataloader seed	1
測試超參數	
Test size	576x576 (與 YOLOV 相同之測試設定)
Test confidence	0.001
NMS threshold	0.5

4.2.1 資料前處理

為了加速模型訓練時程，我們先從 ImageNet DET 訓練集中，隨機抽取每類最多 1000 張照片來作為新的訓練集，用以測試除了資料數量以外，前處理的訓練成效（以下簡稱 ImageNet DET Train 1000），抽取的結果如表 4-13 所示。

表 4-13. ImageNet DET Train 1000 物件數量列表

（共 8305 張照片，9446 個物件）

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
836	1095	1065	1092	554
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
580	1016	1061	1131	1016

接下來先以 YOLOX 為架構基礎對 Mosaic 與 Mixup 進行不同比例的分離實驗，實驗中設定了 5 種不同的分離比例，Mosaic 與 Mixup 的處理比例分別為 0:100、25:75、50:50、75:25、100:0，並與原本的合併比例 100:100 進行比較，以了解分離操作是否會好於合併操作，及了解分離操作的最佳比例。

實驗為了避免不同模型初始參數權重所造成的結果差異，我們採用 3 組初始亂數種子碼進行對比（分別為 1、16、47）。在實驗的過程中，也發現了不同的 Mosaic:Mixup 比例會需要不同的收斂迭代次數 (epoch)，因此在表中也標記了不同比例所設定的最大 epoch 數，以及我們在驗證集中產生最佳 AP50 所在的 epoch 數。由於訓練集收斂的結果 AP50 都接近 99%，因此就不在此贅述。實驗結果如表 4-14 所示。

表 4-14. YOLOX 中不同 Mosaic:Mixup 比例之實驗結果

mosaic : mixup 之比例	ImageNet DET 10 類驗證集 之 AP50			ImageNet DET 10 類測試集 之 AP50		
	1	16	47	1	16	47
模型初始參數 之亂數種子碼						
0:100	87.96	89.72	89.43	87.59	89.57	89.32
25:75	88.17	90.00	90.42	87.63	90.12	90.22
50:50	88.92	90.60	90.83	88.84	90.36	89.93
75:25	89.30	90.08	89.97	88.72	89.64	89.90
100:0	88.73	89.05	89.58	87.99	88.62	88.77
100:100	91.02	91.02	90.63	90.88	90.30	90.62

從表中之實驗結果，可發現 5 種不同的分離比例，在比例為 50:50 時測試有最佳的表現，推測應是此分離比例可讓網路模型均勻的讀取較多型態之前處理照片，故達到較佳的收斂結果。不過在與原始的 100:100 重疊操作相比後，發現原始的重疊操作略好於 50:50 分離比例的成效，因此確認原方法的合併操作前處理方法，可在單幀情境的靜態影像有較佳的表現。

在完成分離實驗後，接下來對訓練集進行不同程度的過濾，除了前述實驗的隨機抽取每類最多 1000 張照片做為「資料類別平衡」的代表之外，也比對了隨機抽取每類最多 2000 張照片（以下簡稱 ImageNet DET Train 2000，其資料分布數量如表 4-15 所示），隨機抽取每類最多 4000 張照片（以下簡稱 ImageNet DET Train 4000，其資料分布數量如表 4-16 所示）作為「去除較多資料訓練」的代表，最後與「全部資料訓練」（以下簡稱 ImageNet DET Train all）的結果進行比較，來了解模型收



斂較佳的資料量。在過濾的過程中，可發現有些訓練集類別在過濾 2000 張以上，就開始出現了物件數量不均勻的情況。

表 4-15. ImageNet DET Train 2000 10 類物件之數量
(共 12805 張照片，14262 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
836	2241	2018	1567	554
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
580	2036	1261	1134	2035

表 4-16. ImageNet DET Train 4000 10 類物件之數量
(共 18168 張照片，19948 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
836	4486	2019	1567	554
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
580	3443	1261	1134	4068

完成不同的資料量篩選後，我們採用同樣的驗證集與測試集進行準確率評估，結果如表 4-17 所示。可發現在驗證集中 ImageNet DET Train 4000 有最好的準確率，而在測試集中 ImageNet DET Train 4000 與 ImageNet DET Train all 有相似的表現。

表 4-17. YOLOX 中不同數量資料集的訓練成效

不同數量 之類別	ImageNet DET 10 類驗證集 之 AP50			ImageNet DET 10 類測試集 之 AP50		
	1	16	47	1	16	47
模型初始參數 之亂數種子碼	1	16	47	1	16	47
ImageNet DET Train 1000	91.02	91.02	90.63	90.30	90.62	90.55
ImageNet DET Train 2000	91.87	92.18	91.96	91.69	92.16	91.62
ImageNet DET Train 4000	93.39	92.93	93.27	92.61	92.57	92.46
ImageNet DET Train all	92.43	93.07	93.26	91.52	93.08	92.64



4.2.2 多階段資料訓練實驗

完成資料前處理的實驗後，我們利用前面實驗所得到的最佳資料量 ImageNet DET Train 4000 進行多階段資料的實驗，比較「一次訓練所有資料」與「二批資料訓練」的結果。我們將採用 2:3 的分割方式，將 ImageNet DET Train 4000 分割為 ImageNet DET Train 1600（即每類最多 1600 張照片，如表 4-18 所示）與 ImageNet DET Train 2400（即每類最多 2400 張照片，如表 4-19 所示）。

表 4-18. ImageNet DET Train 1600 物件數量列表

（共 7264 張照片，8012 個物件）

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
346	1798	801	636	224
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
234	1385	503	457	1628

表 4-19. ImageNet DET Train 2400 物件數量列表

（共 10904 張照片，11936 個物件）

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
490	2688	1218	931	330
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
346	2058	758	677	2440

我們取 ImageNet DET Train 1000 及 ImageNet DET Train 4000 作為一次訓練所有資料的代表，並取 ImageNet DET Train 1600+2400（先訓練 ImageNet DET Train 1600 再訓練 ImageNet DET Train 2400）及 ImageNet DET Train 1600+4000（先訓練 ImageNet DET Train 1600 再訓練 ImageNet DET Train 4000）作為二批資料訓練的代表。將此 4 種不同的訓練方法進行對比，來了解其中的較佳方法，結果如表 4-20 所示。



表 4-20. 多階段資料實驗之訓練成效

單階段與多階段之對比	ImageNet DET 10 類驗證集之 AP50			ImageNet DET 10 類測試集之 AP50		
	1	16	47	1	16	47
模型初始參數之亂數種子碼						
ImageNet DET Train 1600+2400 (max epoch 800)	90.44	90.67	90.45	90.25	90.24	90.03
ImageNet DET Train 1600+4000 (max epoch 800)	91.98	92.06	91.71	91.01	91.53	91.46
ImageNet DET Train 1000 (max epoch 850)	91.02	91.02	90.63	90.30	90.62	90.55
ImageNet DET Train 4000 (max epoch 500)	93.39	92.93	93.27	92.61	92.57	92.46

首先我們觀察「兩批訓練資料為不重複的資料」對比「第一批訓練資料為第二批資料的子集合」的模型表現，可發現後者有較佳的表現，故可知第二次訓練中包含第一次的訓練資料，可能得到較佳的收斂結果。接著我們觀察二階段訓練的較佳方法與單階段訓練的對比，可發現在二階段訓練的模型並沒有優於一階段訓練的模型，且 ImageNet DET Train 1600+2400 比資料量更少的 ImageNet DET Train 1000 還來的差，因此我們可以知道如果是相同來源的資料集且進行物件辨識任務的情況下，進行單階段訓練會有較佳的辨識結果。

4.2.3 學習率與損失函數優化

本小節進行 Cosine 學習率與固定學習率的比較實驗，學習率曲線的差異如圖 4-2 所示。X 軸為迭代次數，不同實驗的所需次數不同，故未特別標記；Y 軸為學習率值，本階段的最高學習率設定為 0.01，Cosine 學習率會逐步下降調整學習率



值，而固定學習率則不會。

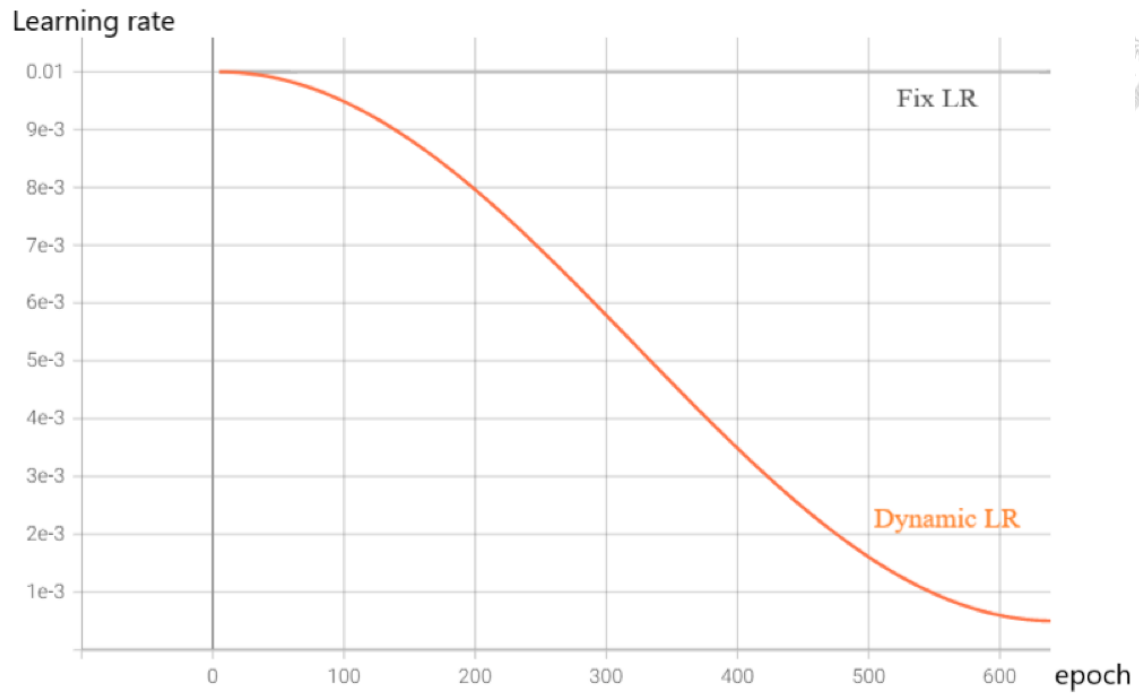


圖 4-2. 固定學習率對比 Cosine 學習率曲線

在此我們以不同 Mosaic 與 Mixup 比例的分離實驗（訓練資料集為 ImageNet DET Train 1000）作為基礎，進行固定學習率與 Cosine 學習率的對比實驗。為了確認 YOLOX 原始的重疊操作是否真的優於分離操作，故再次使用不同的學習率曲線進行實驗與驗證，實驗的結果如表 4-21 所示。

表 4-21. 固定學習率與 Cosine 學習率的對比實驗

mosaic : mixup 之比例	ImageNet DET 10 類驗證集 之 AP50			ImageNet DET 10 類測試集 之 AP50		
模型初始參數 之亂數種子碼	1	16	47	1	16	47
固定學習率 0.01 + Mosaic 與 Mixup 分離						
0:100 (max epoch 850)	85.97	88.30	87.85	85.83	87.99	88.14
25:75 (max epoch 700)	84.97	89.11	88.37	84.55	89.08	88.77
50:50 (max epoch 750)	88.42	89.47	89.59	88.05	89.53	89.27
75:25 (max epoch 600)	86.89	88.88	89.92	86.11	88.53	89.76
100:0 (max epoch 600)	87.31	88.98	89.20	86.80	88.91	88.75
Cosine 學習率 + Mosaic 與 Mixup 分離						
0:100 (max epoch 850)	87.96	89.72	89.43	87.59	89.57	89.32
25:75 (max epoch 700)	88.17	90.00	90.42	87.63	90.12	90.22
50:50 (max epoch 750)	88.92	90.60	90.83	88.84	90.36	89.93
75:25 (max epoch 600)	89.30	90.08	89.97	88.72	89.64	89.90
100:0 (max epoch 600)	88.73	89.05	89.58	87.99	88.62	88.77
Cosine 學習率 + Mosaic 與 Mixup 合併						
100:100 (max epoch 850)	91.02	91.02	90.63	90.88	90.30	90.62

從表中可發現在固定學習率下，50:50 的分離比例為仍有最佳的表現，與先前使用 Cosine 學習率的表現一致，且 Cosine 學習率的各分離比例相對固定學習率均有較佳且一致的表現，因此可確認 Cosine 學習率較適用於我們的辨識任務。另外在進行 Warmup 對比無 Warmup 的實驗時，發現無 Warmup 步驟時，進行訓練的過

程中收斂並不穩定，在有些初始參數下，模型訓練無法收斂，故未進行詳細的結果比較。

在損失函數部分，我們使用固定的種子碼進行 IOU 自訂常數 (Reg Weight) 的調整 (YOLOX 預設為 5，因此我們比較了 1、3、5、7、9 等五種不同的參數)，並使用最佳的挑選資料集進行訓練 (ImageNet DET Train 4000)，以了解合適的 IOU 比重，其結果如表 4-22 所示。

表 4-22. 損失函數中相異之 Reg Weight 的 AP50 表現

(括號中之內容為產生最大 AP50 的 epoch)

	ImageNet DET 10 類驗證集之 AP50					ImageNet DET 10 類測試集之 AP50				
Reg Weight 之常數	1	3	5	7	9	1	3	5	7	9
AP50 分數	92.59	93.28	93.27	93.47	93.10	92.53	92.01	92.46	92.74	92.30

我們發現除了預設的 Reg Weight = 5 之外，1 和 7 的常數也同樣在測試集有不錯的表現，由於差別並不大，因此將 1、5、7 等三種不同的參數用於進行 IOU Loss 中傳統 IOU 與 eIOU 算法的比較，結果如表 4-23 所示。

表 4-23. 損失函數中相異之 IOU Loss 算法的 AP50 表現

	ImageNet DET 10 類驗證集之 AP50			ImageNet DET 10 類測試集之 AP50		
Reg Weight 之常數	1	5	7	1	5	7
IOU	92.59	93.27	93.47	92.53	92.46	92.74
eIOU	92.67	92.45	93.46	92.22	91.98	92.93



從表中可發現 Reg Weight = 7 時，搭配 eIOU 算法可得到最佳的結果。接下來我們進行 Object Loss 的優化，比對了 Binary Cross Entropy Loss (BCE Loss) 與 Varifocal Loss 的成效，其結果如表 4-24 所示。

表 4-24. 損失函數中相異之 Object Loss 算法的 AP50 表現

	ImageNet DET 10 類驗證集之 AP50			ImageNet DET 10 類測試集之 AP50		
	1	5	7	1	5	7
Reg Weight 之常數						
BCE	92.67	92.45	93.46	92.22	91.98	92.93
Varifocal	92.07	92.61	91.64	91.77	92.17	91.36

令人意外的是，考量物件類別數量不平衡的 Varifocal Loss 在 ImageNet DET 10 類測試集中反而使表現較差。因此根據以上實驗，我們決定以表 4-25 之設定作為實驗訓練資料時之最終設定，接著進行單幀情境下識別動態影像中的模糊物件實驗。

表 4-25. 單幀情境識別實驗之最終設定

設定類別	設定內容
篩選資料集	ImageNet DET 4000
Mosaic:Mixup	100:100
訓練模式	單階段訓練法
學習率曲線	Cosine 學習率
損失函數之 Reg Weight	7
IOU 算法	eIOU
Object Loss 算法	Binary Cross Entropy Loss

4.3 單幀情境下識別動態影像的模糊物件

在進行識別單幀情境物件實驗前，我們先檢視原始 ImageNet VID 中 10 類驗證集的模糊物件組成及成因，如表 4-26 及表 4-27 所示。檢視的過程中，發現驗證

集有錯誤的類別標籤（如鸚鵡標成狗、以及貓標記成狗），故刪除了這些錯誤標籤影片，最後統計出各類別有效影片之數量，如表 4-27 中「有效影片數量」。模糊影片的數量占有效影片數量的 32.6%及全局模糊影片數量占模糊影片數量的 87.3%。各類別中除了羊均為清晰影片，其餘類別均有模糊影片的組成。我們也根據全局模糊的種類，進一步統計不同模糊類型的影片數量，如表 4-28 所示。

表 4-26. ImageNet VID 驗證集 10 類物件之影片分析

總影片數量	有效影片數量	清晰影片數量	模糊影片數量	全局模糊數量	局部模糊數量
195	193	120	63	55	8

表 4-27. ImageNet VID 驗證集 10 類物件之模糊物件分析

類別	牛	狗	貓	狐狸	大熊貓	獅子	蜥蜴	兔子	羊	蛇
有效影片數量	23	61	32	17	14	5	12	13	4	12
清晰影片數量	17	41	26	10	8	3	5	8	4	6
模糊影片數量	6	18	6	7	6	2	7	5	0	6
無效影片數量	0	2	0	0	0	0	0	0	0	0

表 4-28. ImageNet VID 驗證集 10 類物件之模糊種類分析

模糊程度	模糊種類	影片數量
全局模糊	鏡頭移動	22
	鏡頭失焦	17
	鏡頭變焦	8
	鏡頭抖動	8
局部模糊	物體移動	8

接下來從我們原始 ImageNet VID 驗證集的 30 類別中，先隨機抽取 10000 幀作為測試基準。然而由於本研究僅討論其中 10 個類別，故將其餘 20 個類別資料去除，最後僅餘 3735 幀（各類別分布如表 4-29 所示）。而後根據表 4-27 之分析從原始驗證集中抽出模糊物件的影片做為新的測試集，由於有部分類別的影片過少，故挑選模糊影片的全部幀（各類別分布如表 4-30 所示），以驗證本階段的模型成效。

表 4-29. ImageNet VID 測試集 10 類物件數量列表

（共 3735 張照片，4868 個物件）

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
724	1134	654	475	470
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
141	332	323	268	347

表 4-30. ImageNet VID 模糊影片測試集 10 類物件數量列表

（共 23930 張照片，30798 個物件）

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
1726	5409	2781	2852	5027
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
664	3444	3032	2431	3432

4.3.1 全局與局部模糊實驗

本階段我們採用高斯模糊法對影像進行模糊化處理，首先使用 4.2 節所述之最佳方式進行訓練，並在前處理階段對影像進行全局高斯模糊。由於我們不確定恰當的模糊比例，故使用固定的種子碼試驗了 10%、15%、20%、30%、40%及 60%等 6 種不同的模糊比例，以了解訓練集需要多少模糊資料進行訓練，才能讓模型對模糊物件有較佳的偵測能力。其實驗結果如表 4-31 所示。

表 4-31. 不同模糊比例之全局高斯模糊實驗

(測量基準為 AP50)

模糊比例	10%	15%	20%	30%	40%	60%
ImageNet DET 驗證集	93.48	93.61	92.91	93.33	92.81	91.45
ImageNet VID 測試集	61.77	64.83	62.96	63.15	62.52	57.22
ImageNet VID 模糊測試集	63.71	65.70	64.03	64.74	62.17	59.58
驗證集之最佳 epoch	449	373	374	448	403	451

從表 4-31 我們可以發現，模糊比例在超過 40%，模型的預測性能開始出現顯著衰退，而最佳性能的前三模糊比例為 15%、20%及 30%，因此我們取測試集中表現最好的 15%模糊比例，進行全局結合局部模糊的綜合前處理實驗。

回顧在 ImageNet VID 驗證集的分析中，由於我們發現全局模糊與局部模糊的影片數量比例為 87.3:12.7，故取相對接近且局部模糊數量又不至於太少的比例 4:1，進行綜合前處理實驗。比較全局模糊與全局結合局部模糊的預測表現，其結果如表 4-32 所示。

表 4-32. 全局模糊與全局結合局部模糊的預測表現

(測量基準為 AP50)

模糊程度	全局	全局+局部	全局	全局+局部	全局	全局+局部
種子碼	1		16		47	
ImageNet DET 驗證集	93.64	93.28	93.16	93.28	93.59	93.29
ImageNet VID 測試集	64.83	63.59	62.03	60.92	63.37	63.76
ImageNet VID 模糊測試集	65.70	63.56	63.61	63.17	63.38	63.61
驗證集之最佳 epoch	373	428	408	371	382	462

從結果可發現採用了全局結合局部模糊的前處理方法，反而會使模型的預測表現變差，推測應該是因為局部模糊的影像與真實局部模糊的影像有落差，導致模型預測表現反而變差了。因此接下來我們均採用全局模糊方法進行彩色與灰階模糊的實驗。

4.3.2 彩色與灰階模糊實驗

而後將以最佳三個模糊比例的中間比例（20%）進行色彩敏感度實驗：進行彩色前處理與灰階前處理的對比，並加入較為清晰的物件（Sigma 值介於 2 到 4 之間）進行比較，結果如表 4-33 所示。

表 4-33. 色彩敏感度與不同模糊程度之比較

(測量基準為 AP50)

色彩	彩色		灰階	
Sigma 模糊區間	2~8	4~8	2~8	4~8
ImageNet DET 驗證集	93.48	92.91	93.43	93.54
ImageNet VID 測試集	61.33	62.96	63.21	61.69
ImageNet VID 模糊測試集	63.41	64.03	66.36	65.44
驗證集之最佳 epoch	440	374	412	425

我們可發現將模糊物件進行灰階前處理之後，模型在 ImageNet VID 模糊測試集有較佳的表現，而在加入較為清晰的物件 (Sigma 值介於 2 到 4 之間) 後，彩色前處理方法的模型表現變差，灰階前處理方法的模型表現變好。推測是因為灰階物件模型需要更清晰的資料來學習物件的輪廓，以達成在真實環境下較佳的預測成效。

了解將模糊物件使用灰階前處理進行訓練會有較佳成效後，我們開始探索能否讓模型透過更多元的訓練資料，達成更好的預測成效呢？因此我們使用彩色與灰階模糊物件各半的前處理方法，前處理的比例為 10% 的彩色模糊物件與 10% 的灰階模糊物件 (根據前段的 20% 模糊比例進行對半分配)，來了解這樣的多元前處理方法是否能把模型的預測成效，更進一步提升達成更好的效果，其結果如表 4-34 所示。

表 4-34. 混合彩色與灰階的模糊物件預測表現

	彩色	灰階	彩色+灰階
Sigma 模糊區間	2~8		
ImageNet DET 驗證集	93.48	93.43	93.20
ImageNet VID 測試集	61.33	63.21	63.33
ImageNet VID 模糊測試集	63.41	66.36	65.47
驗證集之最佳 epoch	440	412	370
Sigma 模糊區間	4~8		
ImageNet DET 驗證集	92.91	93.54	93.10
ImageNet VID 測試集	62.96	61.69	62.65
ImageNet VID 模糊測試集	64.03	65.44	63.48
驗證集之最佳 epoch	374	425	437

先從模糊區間範圍較廣的 Sigma 2 到 8 進行討論，我們可發現彩色混合灰階方法優於彩色方法，但在模糊測試集上表現仍然弱於灰階模糊方法。而看到模糊區間範圍較窄的 Sigma 4 到 8，彩色混合灰階方法在模糊測試集上的表現弱於彩色方法，成為表現最差的一組實驗結果。因此知道更多元的訓練資料，有時候反而會讓模型混淆，從而導致預測表現變差。

最後，我們使用三組亂數種子（1、16 及 47）製造三種不同的模型初始參數，來進行彩色與灰階最佳超參數（彩色之 Sigma 模糊區間取 4 到 8、模糊比例取 15%，灰階之 Sigma 模糊區間取 2 到 8、模糊比例取 20%）對比的多次實驗，實驗結果如表 4-35 所示。

表 4-35. 彩色與灰階最佳超參數的模糊物件預測表現

(測量基準為 AP50)

	彩色	灰階	彩色	灰階	彩色	灰階
種子碼	1		16		47	
ImageNet DET 驗證集	93.61	93.43	93.16	93.06	93.59	93.32
ImageNet VID 測試集	64.83	63.21	62.03	63.49	63.37	62.28
ImageNet VID 模糊測試集	65.70	66.36	63.61	66.77	63.38	64.11
驗證集之最佳 epoch	373	412	408	418	382	403

在三種不同的模型初始參數中，灰階模糊物件前處理方法均在模糊測試集中比彩色模糊物件前處理方法有更好的預測表現，而在靜態物件（ImageNet DET 驗證集）與綜合動態物件（ImageNet VID 測試集）中，兩者有著較為相近的表現。

4.3.3 全局結合真實模糊實驗

在 4.3.2 節中，我們知道灰階模糊法的表現優於彩色模糊法的表現，故在接下來的實驗中，使用表現較佳的灰階模糊處理方法，進行全局模糊化的處理。真實模糊資料的部份，我們採用 ImageNet VID 10%訓練集的其中 10 類物件作為訓練資料，並將其與全局模糊處理後之 ImageNet DET Train 4000 訓練集合併（以下簡稱 ImageNet DET 4000+ ImageNet VID 10%）進行訓練，資料分佈如表 4-36 所示。

表 4-36. ImageNet DET 4000+ ImageNet VID 10% 10 類之物件數量列表

(共 53815 張照片，70380 個物件)

0: 牛	1: 狗	2: 貓	3: 狐狸	4: 大熊貓
6209	17444	7908	5296	5858
5: 獅子	6: 蜥蜴	7: 兔子	8: 羊	9: 蛇
3880	6623	5148	4725	7289



而後我們使用在 COCO 資料集上所預先訓練好的模型參數進行遷移學習，目的是為了加快訓練流程。新的卷積層中則使用 1、16 及 47 三組種子碼給予新的隨機參數，進行全局結合真實模糊的前處理實驗，結果如表 4-37 所示。

表 4-37. 全局結合真實模糊之預測表現

(測量基準為 AP50)

	全局模糊	遷移學習之全局模糊	全局結合真實模糊	全局模糊	全局結合真實模糊	全局模糊	全局結合真實模糊
種子碼	1		16		47		
ImageNet DET 驗證集	93.43	93.79	92.12	93.06	92.37	93.32	92.07
ImageNet VID 測試集	63.21	63.93	68.41	63.49	68.46	62.28	70.48
ImageNet VID 模糊測試集	66.36	64.56	67.54	66.77	66.60	64.11	68.22

由於全局結合真實模糊的前處理實驗採用了遷移學習方法，為了統一變量，我們也在全局模糊法中使用了遷移學習，在種子碼為 1 的統一模型參數實驗中，我們發現使用遷移學習的全局模糊法並未有顯著變化，且在 ImageNet VID 模糊測試集中，反而有預測表現變差的情形，故未於後續的兩組種子碼使用遷移學習法進行訓練。就結果而言，可發現全局模糊結合真實模糊之訓練方法，在 ImageNet DET 所代表的靜態驗證集上有些微衰退，而在 ImageNet VID 所代表的動態測試集上，均有穩定較佳的預測成效。故將採用此方法在接下來的多幀情境下進行物件識別的改進。

4.4 多幀情境下識別動態影像的模糊物件

首先測試單幀情境下，使用僅有靜態物件之 ImageNet DET Train 4000 資料集，對比含有動態物件之 ImageNet DET 4000+ ImageNet VID 10% 資料集進行階段 1 訓

練，並將所得之模型參數引入 YOLOV 前段網路，使用 ImageNet VID 全部的 10 類資料集進行階段 2 訓練（註：為了方便與 YOLOV 進行比較，本階段所取之最佳 epoch 為 YOLOV 作者所採用的最大 AP50-95 所發生的 epoch，再轉為計算當下的 AP50 值），測試集使用 ImageNet VID 模糊測試集進行測試，結果如表 4-38 所示。

表 4-38. 多幀情境下不同訓練集所得之預測成效

階段 1 之訓練集	ImageNet DET Train 4000		ImageNet DET 4000+ ImageNet VID 10%	
階段 2 之訓練集	ImageNet VID 全部之 10 類訓練集			
方法	YOLOV	Ours	YOLOV	Ours
階段 1 之 AP50	58.87	66.36	66.45	67.54
階段 2 之 AP50	57.11	63.18	73.58	76.12

上表中，可發現直接利用 ImageNet DET Train 4000 資料集所訓練而得的模型參數，在進入階段 2 後的 AP50 有不升反降的現象。這是因為階段 2 之訓練使用新的 ImageNet VID 資料集，且階段 1 所訓練而得之前段模型參數已經被固定，導致這些固定的模型參數在偵測新的 ImageNet VID 資料集時，表現不佳。從而使得階段 2 所訓練的後段模型參數也不正確。

另一方面，倘若我們利用 ImageNet DET Train 4000 加上 ImageNet VID Train 10% 資料集來訓練階段 1 之前段模型，讓前段模型在偵測新的 ImageNet VID 資料集時有較佳的表現，則階段 2 所訓練而得之後段模型也會得到正確的結果，故測試集的 AP50 得以正常提升。

最後我們統一在階段 1 使用 ImageNet DET 4000+ ImageNet VID 10% 資料集，使用 ImageNet VID 模糊測試集進行測試，以了解本研究所設計之最終網路模型，對於模糊視訊物件的預測表現，並將其與現有的傳統靜態物件偵測模型代表 (YOLOX) 及現今動態物件偵測模型代表 (YOLOV) 進行比較，其結果如表 4-39 所示。

表 4-39. YOLOX、YOLOV 及本研究在模糊視訊物件的預測表現

(測量基準為 AP50)

種子碼	YOLOX-S	YOLOV-S	Ours
1	66.45	73.58	76.12
16	65.86	74.09	76.89
47	66.58	76.92	77.87
78	65.89	74.64	78.39
91	64.52	74.24	76.82
五項平均 AP50	65.86	74.694	77.218

實驗中我們使用 5 種不同的種子碼生成 5 種不同的模型參數來比較不同模型初始參數所造成的訓練偏差，可發現透過多樣的前處理、損失函數優化及學習率調整後，我們的模型網路達到了穩定與較佳的預測成效。

4.5 小結

在得出應用於模糊視訊物件偵測的最佳模型後，我們將本研究之網路模型與現有的傳統靜態物件偵測模型 (YOLOX) 及現今動態物件偵測模型 (YOLOV) 進行性能表現的比較，如表 4-40 所示。

表 4-40. 本研究與現有模型之各項表現比較

Model	Params (M)	GFLOPS	Time (ms)	AP50 (%)
YOLOX-S	8.94	21.69	8.3	65.86
Ours with single frame	8.94	21.69	8.5	67.54
YOLOV-S	9.83	25.72	8.9	74.69
Ours with multi frame	9.83	25.72	8.9	77.22

回顧在第一章中，圖 1-2 所提到模糊物件辨識問題，我們的方法在模糊場景下成功的預測了蛇類的標籤，如圖 4-3 所示。新的方法成功在原本預測為蜥蜴且 4 張圖像中只有 2 張產生物件框的情況下，改進為 4 張均產生物件框且正確預測類別。

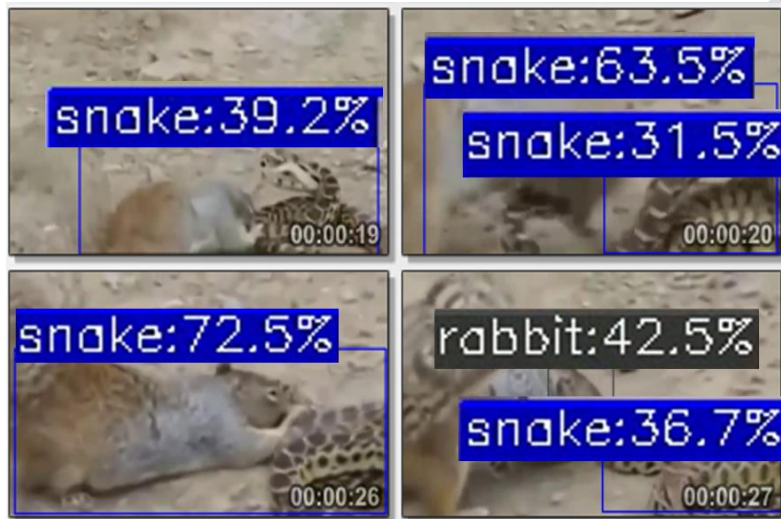


圖 4-3. 本研究方法應用於模糊動物辨識的預測表現

在瀏覽預測結果的過程中，也發現一些預測錯誤的部分。在圖 4-4 中，可看到我們的方法會將類似蛇類的枕頭辨識為蛇類，雖然不會影響準確率的部分，但從 recall 的角度來看的話卻下降了。我們推測是因為使用了模糊前處理的方法，使得模型對於特定物件特徵更敏感所造成的現象。



圖 4-4. 預測錯誤的物件類別

(左為 YOLOX-S，右為本研究之單幀預測方法)

總體而言，本研究在模型參數與現有網路模型一致的情況下，在相近的預測時間內實現了更好的模糊視訊物件預測表現。

第五章 結論與未來展望



基於前述章節之實驗結果與討論，我們將先歸納整理本研究之結論，而後根據研究成果提出本研究議題之未來展望。

5.1 研究結論

本研究使用 ImageNet 所提供的 DET 及 VID 資料集作為訓練、驗證及測試基準，利用現有靜態影像物件偵測模型 YOLOX 結合模糊物件生成的資料前處理方法進行調整與優化，改進為適用於視訊模糊物件偵測的網路模型。本研究主要的改進之處有下列三項。

(1) 資料篩選與前處理

在一開始的篩選中，只取每類 4000 張左右的照片，並採用單階段的訓練，這是因為我們發現更多的資料無益於網路模型的訓練及預測表現。而在前處理的部份，我們採用了全局影像的灰階高斯模糊法取代彩色高斯模糊法，進而取得了比原先更佳的模型預測表現（AP50 準確率提升約為 1.2%）。

(2) 損失函數的優化

採用了更高的 IOU 權重來提高物件框的準確度、並使用 eIOU 算法取代 IOU 算法來提升物件框評分的可靠性。相比於原先的 IOU 權重及傳統 IOU 算法，此新方法在模型參數及其餘隨機參數完全一致的情況下提升了 0.47% 的 AP50 準確率。

(3) 兼具與新模型相容的可能性

將本研究所調整之模糊視訊物件網路模型與現有動態影像物件偵測模型（YOLOV）進行整合，在改進模糊物件預測準確率的同時，也兼具網路模型應用於各情境及新模型的泛用性。

整體而言，我們的方法在單幀情境下於 ImageNet VID 模糊物件測試集中實現了比原先基礎模型 YOLOX 高出了 1.68% 的準確率；而在多幀情境下，比目前動

態物件偵測模型 YOLOV 高出了 2.53% 的準確率，且在預測速度上持平目前 YOLOX 及 YOLOV 的預測速度，這讓我們在實際應用於真實場景中獲得性能改善的同時，又不會需要增加額外的時間來進行預測。

根據前述研究，我們可發現多幀情境下現有的動態物件偵測模型，會因為受到預測品質較差的單幀結果，導致多幀推理的錯誤。如果我們提升單幀預測品質，則可以解決此問題。

5.2 未來展望

目前所設計的方法為單幀情境下，進行影像模糊處理前處理，以加強網路模型在模糊物件的預測成效。然而，基於高斯模糊所生成的模糊影像與真實影像仍有不同，真實影像可能會在影像中的不同區域有不同程度的模糊，或者是在鏡頭失焦的情境下遠近不同的物體，可能會同時存在清晰與模糊的現象。

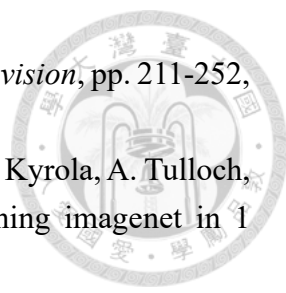
在目前的實驗中，局部模糊的影像的前處理方法，在模糊影像中反而有較差的預測表現，這可能是因為局部模糊影像的生成方法不夠貼近真實物件的局部模糊現象。因此，若是能調整模糊前處理的方法，加入 Motion Blur 來模擬物體運動的模糊、加入 Turbulence Blur 來模擬場景中霧氣所帶來的模糊，使得生成的影像更多元且接近真實模糊的影像，相信模型會在單幀情境下有更好的預測表現。

此外，此多幀的參考方法仍有改進空間。目前所採用的 YOLOV 後段模型，僅使用靜態模型所輸出的 IOU 及類別進行多幀的結合推論。若是能在後段模型中把部分多幀影像資訊一併進行考量，且盡量的避免使用過多資訊導致性能損失，也是一個未來可供探索的議題。

參考文獻



- [1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [2] A. Sabater, L. Montesano, and A.C. Murillo, "Robust and efficient post-processing for video object detection," In *Proceedings of the 2020 IEEE International Conference on Intelligent Robots and Systems*, pp. 10536-10542, 2020.
- [3] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, , M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," *arXiv preprint arXiv:2209.02976*, 2022.
- [4] C. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7464-7475, 2023.
- [5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
- [6] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "Varifocalnet: An iou-aware dense object detector," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8514-8523, 2021.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788, 2016.
- [8] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271, 2017.
- [9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," In *Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969, 2017.
- [11] L. He, Q. Zhou, X. Li, L. Niu, G. Cheng, X. Li, W. Liu, Y. Tong, L. Ma, and L. Zhang, "End-to-end video object detection with spatial-temporal transformers," In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1507-1516, 2021.
- [12] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," In *Proceedings of the IEEE international conference on computer vision*, pp. 8878-8887, 2019.
- [13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, H. Zhiheng, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, "Imagenet large scale

- 
- visual recognition challenge,” *International journal of computer vision*, pp. 211-252, 2015.
- [14] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y.-Q. Jia, and K. He, “Accurate, large minibatch SGD: Training imagenet in 1 hour,” *arXiv preprint arXiv:1706.02677*, 2017.
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587, 2014.
- [16] R. Girshick, “Fast R-CNN,” In *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448, 2015.
- [17] S. Zheng, Y. Wu, S. Jiang, C. Lu, and G. Gupta, “Deblur-yolo: Real-time object detection with efficient blind motion deblurring,” In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1-8, 2021.
- [18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” In *Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988, 2017.
- [19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” In *Proceedings of the Computer Vision–ECCV 2016: 14th European Conference*, pp. 21-37, 2016.
- [20] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-guided feature aggregation for video object detection,” In *Proceedings of the IEEE international conference on computer vision*, pp. 408-417, 2017.
- [21] Y. Chen, Y. Cao, H. Hu, and L. Wang, “Memory enhanced global-local aggregation for video object detection,” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 10337-10346, 2020.
- [22] Y. Shi, N. Wang, and X. Guo, “YOLOV: Making still image object detectors great at video object detection,” In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2254-2262, 2023.
- [23] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, “Focal and efficient IOU loss for accurate bounding box regression,” *Neurocomputing*, pp. 146-157, 2022.
- [24] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [25] 王家瑜, “基於模糊影像之物件偵測,” 碩士, 資訊工程研究所, 國立臺灣大學, 臺北市, 2022.

附錄



ImageNet DET 類別列表

手風琴	飛機	螞蟻	羚羊	蘋果	狢狢	洋薊	斧頭	嬰兒床	背包
百吉餅	平衡木	香蕉	OK 繡	班卓琴	棒球	籃球	浴帽	燒杯	熊
蜜蜂	甜椒	長椅	自行車	活頁夾	鳥	書架	弓	領結	碗
胸罩	墨西哥捲餅	巴士	蝴蝶	駱駝	開罐器	汽車	購物車	牛隻	大提琴
蜈蚣	鏈鋸	椅子	風鈴	雞尾酒調酒器	咖啡機	電腦鍵盤	電腦滑鼠	瓶塞刀	奶油
槌球	拐杖	黃瓜	杯子	尿布	數位時鐘	洗碗機	狗	家貓	蜻蜓
鼓	啞鈴	電風扇	大象	散粉	無花果	文件櫃	花盆	長笛	狐狸
法國號	青蛙	平底鍋	大熊貓	金魚	高爾夫球	高爾夫車	鱈梨醬	吉他	吹風機
髮膠	漢堡	鐵錘	倉鼠	口琴	豎琴	寬邊帽	包心菜	安全帽	河馬
高低單槓	馬	熱狗	iPod	木屑	水母	無尾熊	杓子	瓢蟲	台燈
筆記型電腦	檸檬	獅子	口紅	蜥蜴	龍蝦	連身泳衣	沙鈴	麥克風	微波爐
牛奶罐	迷你裙	猴子	摩托車	蘑菇	釘子	頸圈	雙簧管	橙子	水獺
鉛筆盒	削鉛筆機	香水	人	鋼琴	鳳梨	乒乓球	大水罐	披薩	塑膠袋
盤架	石榴	冰棒	豪豬	電動鑽	椒鹽脆餅	印表機	冰球	沙袋	錢包
兔子	球拍	射線	紅熊貓	冰箱	遙控器	橡皮擦	橄欖球	尺	鹽
薩克斯風	蠍子	螺絲刀	海豹	綿羊	滑雪板	臭鼬	蝸牛	蛇	雪地摩托車
雪梨	肥皂液分配器	足球	沙發	鏟子	松鼠	海星	聽診器	爐子	濾網
草莓	擔架	太陽眼鏡	游泳褲	豬	注射器	桌子	磁帶播放器	網球	蟬蟲
領帶	老虎	烤麵包機	交通燈	火車	長號	小號	龜	顯示器	單輪車
吸塵器	小提琴	排球	華夫餅製作器	洗衣機	水瓶	船舶	鯨魚	酒瓶	斑馬