



基於YOLO模型的寵物貓咪行為偵測系統

Pet cat behavior recognition based on YOLO model

指導教授：陳榮靜 教授

研究生：洪修德

中華民國 110 年 7 月

The logo of Chaoyang University of Technology is a circular seal. It features a gear-like outer ring with the university's name in English, "CHAORYANG UNIVERSITY OF TECHNOLOGY", and the year "1994" at the bottom. Inside the ring is a stylized gear or sunburst design.

朝陽科技大學資訊管理系
Department of Information Management
Chaoyang University of Technology

碩士論文

Thesis for the Degree of Master

基於YOLO模型的寵物貓咪行為偵測系統
Pet cat behavior recognition based on YOLO model

指導教授： 陳榮靜 教授 (Rung-Ching Chen)

研究生： 洪修德 (Hsiu-Te Hung)

中華民國 110 年 7 月

July, 2021

中文摘要

近年來，越來越多家庭會飼養寵物，因為經濟不景氣，許多現代人不敢結婚生小孩，但是又希望家裡有人能陪伴，所以許多人會以飼養寵物來代替生小孩，如何照顧寵物在現今是非常重要的課題。在台灣，根據農委會統計，西元 2017 年台灣人飼養寵物高達 250 萬口，平均 10 個人會養 1 隻寵物，而預估 2021 年將有 300 萬個家庭至少擁有一隻寵物，所以這些動物也與我們生活上的科技息息相關。許多家庭因為長時間在外工作而把寵物單獨留在家裡，但是又怕家裡的寵物太孤單或是太頑皮而亂咬家俱，所以會使用一些寵物電子產品來與寵物互動，例如：自動餵食機、互動攝影機、電子球和寵物監視器等等。

隨著時代的進步，科技日新月異，機器學習與人工智慧越來越多應用於交通、物流和居家方面等，而在寵物方面，寵物監控近年來也相當熱門。市面上的寵物攝影機功能繁多，能夠與寵物遠端互動和自動餵食等，但是這些攝影機並不會第一時間通知飼主家中的寵物正在做什麼，若家中寵物誤食或是上廁所太久，飼主無法在第一時間知道，則可能會對寵物造成傷害。

所以本篇研究針對居家寵物即時監控的問題，藉由樹莓派當作監控系統，提出了一種基於 YOLOv3-Tiny 的識別系統，利用 YOLOv3-Tiny 方法具有快速檢測及較好的邊界框預測。對於房間內採集了 1410 張貓咪動作的樣本進行標記並訓練，為了測試 YOLOv3-Tiny 性能，我們也訓練了 YOLOv3 模型進行比較，根據輸入圖像的類別，輸出六個寵物動作類別的結果。分別為睡覺、吃飯、坐下、行走、上廁所和翻垃圾桶，透過影像辨識建立串流到使用者手機和電腦，當貓咪上廁所過久或是亂翻垃圾桶時，會即時傳訊息到飼主的手機，達到即時預防的寵物遠端監控系統。

關鍵詞： 樹莓派、YOLOv3-Tiny、影像辨識

Abstract

In recent years, more and more families will keep pets. Because of the economic downturn, many modern people dare not marry and have children, but hope that someone in the family can accompany them. Therefore, many people will keep pets instead of having children. How to take care of pets is a very important topic nowadays. In Taiwan, an estimated 3 million families have at least one pet, and these animals are also closely related to the technology in our lives. Many families leave their pets alone at home because they work outside for a long time, but they are afraid that their pets are too lonely or too naughty to bite the furniture, so they use some pet electronic products to interact with their pets, such as automatic feeding machines, Interactive cameras, electronic balls and pet monitors, etc..

With the progress of the times and the rapid development of science and technology, machine learning and artificial intelligence are increasingly used in transportation, logistics and homes. In terms of pets, pet monitoring has become very popular in recent years. The cameras on the market have various functions, which can interact with pets remotely and automatically feed. However, these cameras do not immediately inform the owners what their pets are doing. If the pet in the house eats by mistake or goes to the toilet for too long, the owner cannot know the first time, and it may cause harm to the pet.

This research aims at the real-time monitoring of domestic pets. Using the Raspberry Pi as the monitoring system, a recognition system based on YOLOv3-Tiny is proposed. The YOLOv3-Tiny method is used for rapid detection and better bounding box prediction. The 1410 cat behavior photos collected in the room were labeled and trained. In order to test the performance of YOLOv3-Tiny model, we also trained the YOLOv3 model for comparison. According to the category of the input image, the results of six pet action categories were output. They are sleeping, eating, sitting, walking, going to the litter box, and search on a trash can. Through

image recognition, create a stream to the user's mobile phone and computer. When the cat goes to the toilet for too long or search on a trash can, the message will be sent to the user's mobile phone in real time to achieve the remote pet monitoring system for real-time prevention.

Keyword: Raspberry Pi, YOLOv3-Tiny, image recognition

致謝

本篇論文首先要感謝我的指導老師陳榮靜老師，謝謝您在這兩年多的碩士班的求學過程中努力栽培指導，從一開始的論文選題、資料蒐集到撰寫論文的期間，當我遇到困難或難題時，老師總是能毫不吝嗇地提供許多想法和方法，讓我度過難關，非常感謝老師精心的指導和付出。

在求學過程中，我非常感謝一路互相支持和幫忙我的許多同學，而其中諺庭同學幫助我非常多，當我在研究上遇問題時，總是能在第一時間伸出援手幫助我。另外，還有兩位學長也非常地幫助我，俊元學長和喬恩學長，謝謝你們當我遇到課業以外的問題時，能幫助和協助我。

最後要感謝我的家人，提供我衣食無缺的生活，總是在背後默默的支持我，讓我能夠安心和專心地在課業上不受影響，最後，我再一次真誠地向曾經幫助過我的老師、同學和家人表示感謝！

目錄

中文摘要.....	I
Abstract.....	II
致謝.....	IV
目錄.....	V
圖目錄.....	VI
表目錄.....	IX
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	2
1.3 研究目的.....	2
1.4 論文架構.....	3
第二章 文獻探討.....	3
2.1 寵物照顧.....	3
2.2 樹莓派.....	4
2.3 深度學習.....	6
2.4 YOLO.....	7
2.5 模型評估.....	10
2.6 IFTTT.....	11
第三章 研究方法.....	13
3.1 研究架構.....	13
3.2 行為偵測.....	18
3.3 YOLOv3-Tiny.....	20
第四章 實驗結果與分析.....	22
4.1 實驗環境.....	22
4.2 數據集.....	22
4.3 實驗結果.....	23
第五章 結論與未來展望.....	48
5.1 結論.....	48
5.2 未來展望.....	49
參考文獻.....	49

圖目錄

圖 1 樹梅派 4B 和攝像頭	6
圖 2 Darkmet-53 模型	9
圖 3 YOLOv3 邊框預測	10
圖 4 研究架構	13
圖 5 YOLOv3-Tiny-cfg 參數	15
圖 6 YOLOv3-Tiny-data 參數	15
圖 7 訊息發送程式碼-1	16
圖 8 訊息發送程式碼-2	17
圖 9 IFTTT 連結 Notify	17
圖 10 樹莓派安裝 VNC Server	18
圖 11 寵物異常行為偵測流程圖	19
圖 12 YOLOv3-Tiny 主幹網路	21
圖 13 BBox Label Tool 手動標示圖片	22
圖 14 YOLOv3-Tiny 9000 次培訓步驟-1	24
圖 15 YOLOv3-Tiny 9000 次培訓步驟-2	25
圖 16 YOLOv3-Tiny 9000 次培訓步驟-3	25
圖 17 YOLOv3-Tiny 9000 次培訓步驟-4	26
圖 18 YOLOv3-Tiny 9000 次培訓步驟-5	26
圖 19 YOLOv3-Tiny 9000 次培訓步驟-6	27
圖 20 YOLOv3-Tiny 9000 次培訓步驟-7	27
圖 21 YOLOv3-Tiny 9000 次培訓步驟-8	28
圖 22 YOLOv3-Tiny 9000 次培訓步驟-9	28
圖 23 YOLOv3-Tiny 9000 次培訓步驟-10	29
圖 24 YOLOv3-Tiny 12000 次培訓步驟-1	29

圖 25 YOLOv3-Tiny 12000 次培訓步驟-2.....	30
圖 26 YOLOv3-Tiny 12000 次培訓步驟-3.....	30
圖 27 YOLOv3-Tiny 12000 次培訓步驟-4.....	31
圖 28 YOLOv3-Tiny 12000 次培訓步驟-5.....	31
圖 29 YOLOv3-Tiny 12000 次培訓步驟-6.....	32
圖 30 YOLOv3-Tiny 12000 次培訓步驟-7.....	32
圖 31 YOLOv3-Tiny 12000 次培訓步驟-8.....	33
圖 32 YOLOv3-Tiny 12000 次培訓步驟-9.....	33
圖 33 YOLOv3-Tiny 12000 次培訓步驟-10.....	34
圖 34 YOLOv3-Tiny 15000 次培訓步驟-1.....	34
圖 35 YOLOv3-Tiny 15000 次培訓步驟-2.....	35
圖 36 YOLOv3-Tiny 15000 次培訓步驟-3.....	35
圖 37 YOLOv3-Tiny 15000 次培訓步驟-4.....	36
圖 38 YOLOv3-Tiny 15000 次培訓步驟-5.....	36
圖 39 YOLOv3-Tiny 15000 次培訓步驟-6.....	37
圖 40 YOLOv3-Tiny 15000 次培訓步驟-7.....	37
圖 41 YOLOv3-Tiny 15000 次培訓步驟-8.....	38
圖 42 YOLOv3-Tiny 15000 次培訓步驟-9.....	38
圖 43 YOLOv3-Tiny 15000 次培訓步驟-10.....	39
圖 44 YOLOv3 模型的平均損失和 mAP.....	41
圖 45 YOLOv3-Tiny 模型檢測速度	42
圖 46 YOLOv3 模型檢測速度	42
圖 47 行走類別被偵測為翻垃圾桶類別	43
圖 48 行走檢測結果	43
圖 49 進食檢測結果	44





圖 50 睡覺檢測結果.....	44
圖 51 坐下檢測結果.....	45
圖 52 翻垃圾桶檢測結果.....	45
圖 53 上廁所檢測結果.....	46
圖 54 翻垃圾桶回傳至手機結果.....	46
圖 55 上廁所超過 30 秒回傳至手機結果.....	47
圖 56 透過手機使用 VNC 即時察看結果.....	47
圖 57 透過電腦使用 VNC 即時察看結果.....	48



表目錄

表 1 混淆矩陣	10
表 2 YOLOv3-Tiny batches9000 各項指標結果	40
表 3 YOLOv3-Tiny batches12000 各項指標結果	40
表 4 YOLOv3-Tiny batches15000 各項指標結果	41

第一章 緒論

在本章中將說明本研究的研究背景、研究動機、研究目的與本計畫書的架構。

1.1 研究背景

近年來，隨著科技的日新月異，無論是網路或是硬體都有很大的進步，對於資訊也越來越隨手可得，許多人們的生活也逐漸改變，想要獲取資訊不在侷限於書本或是人與人之間的交談，大部分的資訊都可以透過網路來取得，也因為這樣，人們為了提高生活品質，對於物聯網科技有了更多的研究與應用。

現今的社會中，許多人對於物聯網這個概念已經不在陌生，物聯網的概念很簡單，透過生活上的物品裝上感測器達到自動化，將沒有意識的物品變得有意識。以智慧冰箱為例，當冰箱中的蘋果已經沒了，冰箱會自動發出訊息至家主的手機中，提醒是否需要添購新的蘋果並顯示水果行的資訊。物聯網不僅漸漸地融入大家的生活也為人們帶來許多便利，成為不可或缺的一部分。

目前物聯網具有很大的市場規模與潛力，因為物聯網可以應用的範圍非常廣泛，從汽車、醫療、工業、零售、運輸到智慧家庭都可以看到物聯網的蹤影，其中智慧家庭使用物聯網技術最為廣泛，目前市場上不斷有廠商陸續推出有關於智慧家庭、居家安全與智慧保全等產品，像是 LifeSmart[1]智慧家庭系統提供許多功能，包括影像監看、環境檢測、家電控制、情境照明控制、智慧門鎖及藍芽應用設備等，所有設備皆可透過平板或智慧手機連接進行偵測與控制，達到智慧家庭的概念。

而智慧家庭現在不只使用在人身上，近年來由於家庭飼養寵物的比例逐漸增加，所以智慧家庭也可以服務在寵物身上，市面上已經有許多

關於寵物監控的產品，像是 Pawbo[2]寵物互動攝影機，提供雙向語音、互動遊戲和點心餵食等，一個小小的監視器就提供語音、影像辨識和遠端監控等功能，本篇論文也將利用樹莓派連結攝影機來遠端監控寵物。

1.2 研究動機

許多家庭都會飼養寵物，如何照顧寵物成為現代人重要的課題之一，在台灣，飼養狗的家庭比飼養貓的比例來的多，大部分的家庭都會將狗圈養在自家的門外，少部分的小型犬或是貓咪會飼養在家中，而未訓練過的寵物可能會亂大小便和亂咬家中的家俱，造成飼主的困擾，而若家中空氣不流通、光照時間和雜物過多等環境因素都有可能造成寵物會有異食癖[3]以及泌尿道疾病[4]等情況發生。市面上雖然有許多寵物監控產品能夠透過攝影機查看寵物正在做什麼，但是飼主不可能將注意力隨時隨地都注意在寵物監控視頻上，而若飼主剛好離開寵物監控視頻，寵物誤食、亂咬電電線或是亂翻垃圾桶，都有可能會對寵物帶來危害。所以本篇論文針對寵物遠端監控的問題，提出了一種利用樹莓派作為智慧攝影機並結合影像辨識，能夠即時查看寵物正在做甚麼，以及當飼主沒有在監控視頻前，若寵物發生亂翻垃圾桶或是上廁所過久等情況都會第一時間發送訊息至飼主的手機，提醒飼主寵物的情況，達到即時預防的寵物監控系統。

1.3 研究目的

本研究希望蒐集家中貓咪的各種行為動作的照片，並透過樹莓派與YOLOv3-Tiny 深度學習來分析貓咪的行為動作準確率，透過影像識別與深度學習，當貓咪發生亂翻垃圾桶或是上廁所太久都會將訊息傳至飼主的手機，最終實現即時預防的寵物監控系統。

1.4 論文架構

本論文架構如下：

- 第 1 章 在本章中將說明本研究的研究背景、研究動機與研究目的。
- 第 2 章 本章為文獻探討，將分別說明寵物照顧、樹莓派、深度學習、YOLO、IFTTT。
- 第 3 章 本章將說明本研究所提出的方法，將分別說明研究架構、行為偵測、YOLOv3-Tiny。
- 第 4 章 本章將說明研究的實驗環境、數據集、實驗結果與分析。
- 第 5 章 本章將說明本研究的結論與未來研究。

第二章 文獻探討

在本章中將分別說明寵物照顧、樹莓派、深度學習、YOLO、IFTTT。

2.1 寵物照顧

因為近年來物價不斷上漲，薪資也不見調漲，造成如今經濟不景氣，許多現代人不敢輕易結婚生子，根據 World Population Review[5]報導，台灣是 2019 年全球生育率最低的國家，生育率僅 1.218，預計台灣在 2022 年將出現人口負成長的情況，也因為這樣的關係，導致現在社會已經進入了高齡化和少子化。大部分的現代人不想結婚生子又希望家中有人能夠陪伴，所以選擇飼養寵物來代替生子，花費也不會造成太大的負擔。目前飼養狗的比例高於飼養貓，雖然飼養狗的人較多，但是近年來飼養貓的成長率卻是高於飼養狗。根據時事股份有限公司[6]統計，2020 年下半年將會出現黃金交叉，飼養寵物數將超過新生兒數，於 2021 年時預估寵物將高達 295 萬隻，超過 283 萬新生兒，因為如此，寵物的商機也漸漸萌芽。

由於飼養寵物的家庭逐漸增加，代表著寵物的相關產業也漸漸興起，根

據財政部[7]統計，從 2008 年寵物相關產業僅 3734 家，至 2018 年增加至 6486 家，成長幅度增加 73.7%，銷售總額從 154.5 億元成長至 265.8 億元，由此可知寵物帶來了龐大的商機。許多飼主將寵物如小孩般一樣細心呵護，對於寵物的飲食、娛樂和健康都希望能有較好的待遇，而業者們也看準了這個商機。在飲食方面，大部分的飼主不只希望寵物能夠吃得飽也要吃得健康，而陪心寵糧[8]能夠滿足飼主的需求，根據飼主寵物的品種、年齡、體重和活動量等，評量寵物所需的營養成分，提供客製化的糧食，讓寵物能夠吃得安心。在娛樂方面有也許多選擇，像是百威犬舍[9]能夠在良好的環境與寵物互動並教導寵物能夠做出簡單的行為動作，像是坐下、等待和臥倒等動作。在許多人的觀念中，認為寵物美容只是時尚的一環，但是其實寵物美容也能夠保持寵物的清潔衛生，我們好好[10]針對狗的皮膚、牙齒和指甲都有優質的衛生清潔服務，若不想將帶寵物出門，FURKIDS[11]行動寵醫也提供到府看診的服務，讓不適應在外看診的寵物也能在家安心診療。

雖然現今能夠照顧寵物的管道不勝枚舉，但是還是有許多飼主疏忽寵物的健康，根據動物保護處[12]統計，貓最多的死因是腎衰竭，而狗則是癌症居多，貓咪腎衰竭主要因為長期飲水不足、泌尿道感染和異食癖的關係所導致，貓咪若發生泌尿道感染，最明顯的症狀是上廁所次數頻繁以及沒有排出尿液，而若家中雜物過多則會導致寵物有異食癖的情況發生，間接造成貓咪腎衰竭的風險提升，如何有效的照顧寵物相當重要。

2.2 樹莓派

樹莓派最初是為教育而設計的一款 Linux 系統的單板機電腦，體積約信用卡大小且價格便宜。近年來樹莓派不斷的改進，許多人將樹莓派應用於各個層面，例如：教育、醫療、交通、安全和居家等。劉家名[13]將樹莓派和手勢控制板做結合，提出了一種雙重認證的手勢辨識架構，收集手勢特徵並對圖片進行預處理再將資料傳至伺服器進行深度學習的預測，最終在四

種和六種手勢中都有良好的準確率。在居家方面，Raju A Nadaf[14]等學者提出了一種具有家庭入侵檢測系統的智能鏡，以樹莓派作為模型並使用 Haar[14]分類器對臉部和眼睛進行檢測，當入侵者闖入時，系統將會拍下入侵者的臉部並發送訊息通知使用者，此外，該智能鏡如同鏡子般能夠用來梳裝也不易被入侵者察覺。

本篇論文所使用的樹莓派和攝像頭如圖 1，硬體採用 Raspberry Pi 4 Model B V1.2 版，其硬體規格處理器採用 Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz；GPU 採用 Dual Core VideoCore VI® Multimedia CoProcessor；Open GL ES 3.0; hardware-accelerated OpenVG; H.265(4Kp60) high-profile decode；記憶體為 2GB LPDDR4-2400 SDRAM (與 GPU 共享)；乙太網路採用 Gigabit Ethernet；無線網路採用雙頻 2.4G Hz 及 5G Hz IEEE802.11ac，藍芽 5.0，BLE；顯示接口採用兩個 micro-HDMI(4Kp60)；其他接口為 2 個 USB 3.0 ports 及 2 個 USB 2.0 ports；儲存採用 microSD 64G；相較於 Raspberry Pi 3，第四代的樹莓派在各方面都有明顯的升級，CPU 快了 3 倍，在 USB 埠方面，首次使用 USB 3.0，而視訊輸出方面，從原本的 Composite RCA 和 HDMI v1.3 升級為兩組 HDMI v2.0。

本研究所使用的攝像頭，採用 Raspberry Pi Camera Module V2.1，其規格為 800 萬像素；3280x2464 像素靜態照片；SONY IMX219 感光晶片；攝像速率支援 1080p 30FPS、720p 60FPS、640x480p 90FPS。作業系統使用 NOOBS 安裝 Raspbian 系統。



圖 1 樹梅派 4B 和攝像頭

2.3 深度學習

深度學習是從機器學習中的人工神經網路發展出的新領域，而隨著深度學習的發展，許多學者應用於影像辨識、自然語言處理和生物醫學等領域都有很好的效果。Mark F. Hansen[15]等人使用 CNN 演算法對於農場的豬隻臉部進行辨識，減少人力與時間上的花費，在準確率上也高達 96.7%；Hung Nguyen[16]等人提出了一種自動化的野生動物檢測系統，使用 CNN 模型對野生動物進行檢測，在檢測大部分的野生動物準確率都高達 90%，較為常見的動物準確率高達 96%；Xi Shu-shu[17]等人利用 SSD 網路對狗的品種進行檢測，SSD 網路採用多尺度特徵進行檢測，在檢測精度上有良好的表現，該研究中檢測泰迪犬平均準確率高達 96.7%；Fang Wu[18]等學者使用傳統 CNN 演算法對犬種進行辨識，在實驗中，從 Kaggle 數據及提取了 5000 張與狗相關的照片，針對常見的 50 種品種中，準確率高達 85%，而在另外 120 種不常見的品種中，準確率僅為 64%，明顯下降許多；Punyanch

Borwarnginn[19]等學者認為目前家庭最多飼養的寵物為狗，由於狗的種類眾多，每種犬種都有特殊的疾病和健康狀況，所以為了提供適當的治療，識別品種極其重要，也因此本文中利用區域性二值模式(LBP)和方向梯度直方圖(HOG)對傳統 CNN 演算法進行改進，對犬種進行識別，在實驗結果表明，傳統的 CNN 演算法準確率為 79.25%，而改進過後的方法準確率高達 96.75%，精度有非常顯著的提升。

上述的研究所提出深度學習對於動物的影像辨識都有良好的貢獻，故本篇論文將會使用深度學習的方式進行實驗。

2.4 YOLO

YOLO 最早於 2015 年，由 Joseph Redmon 等人在 You Only Look Once: Unified, Real-Time Object Detection[20]這篇論文所提到，YOLO 是基於單個神經網路的目標檢測系統，與其他演算法不同，YOLO 將圖片分為多個單元，並預測每個單元的邊界框座標和所屬類別的概率，YOLO 最大的優點在於速度，每秒可處理 45 幀。而在之後 Joseph Redmon 等人對 YOLO 進行了改進，提出了 YOLOv2[21]與 YOLOv3[22]。YOLOv2 的全文為 YOLO 9000: Better, Faster, Stronger，在論文中提到該模型能夠檢測 9000 種物體，使用新的特徵提取器稱為 Darknet-19，藉此提高模型的檢測速度和準確率。而 YOLOv3 為了達到更好的分類效果，在 YOLOv2 提出的 Darknet-19 的基礎上做了改良，採用了更深的卷積層神經網路 Darknet-53，主要由 1x1 和 3x3 的卷積層組成，共有 53 層，如圖 2 所示。隨著網路層數不斷加深，採用了一般類神經網路加深時常用的 ResNet[23]結構來解決梯度問題，這使得訓練深層網路難度大大減小，精度提升也較為明顯。

YOLOv3 也使用 FPN [24](Feature Pyramid Networks)多層級預測架構以提升小物體預測能力，特徵層從單層 13x13 變成了多層 13x13、26x26 和 52x52，單層預測 5 種 bounding box 變成每層 3 種 bounding box (共 9

種)。使用 FPN 的架構可以讓低層較佳的目標位置和高層較佳的語義特徵融合，並且在不同特徵層獨立進行預測，使得小物體檢測改善效果十分明顯。

在邊框預測模型中，YOLOv3 延續了 YOLOv2 的做法，採用原公式，其中 (tx, ty, tw, th) 就是模型的預測輸出（網路學習目標）。 (cx, cy) 是單元格的座標偏移量（以單元格邊長為單位）， σ 表示 sigmoid 函數， (pw, ph) 是預設的 Anchor Box 的邊長， (bx, by, bw, bh) 就是最終得到的預測出的邊界框的中心座標和寬高，如圖 3 所示。而在分類預測，YOLOv3 不在使用 softmax 做分類，而是使用獨立的 logistic 做二分類並搭配 binary cross-entropy 來衡量分類上的損失，這種方法的好處是可以處理重疊的多標籤問題。

因為 YOLO 在物體偵測上具有良好的檢測速度與檢測精度，許多學者利用 YOLO 模型來進行影像辨識上的研究，Cuixiao Liang[25]等人，使用 YOLOv3 模型對夜間自然環境中的荔枝果實進行檢測，此研究為了在夜間檢測荔枝能夠取得較好的檢測結果，對於夜間探照燈的亮度水平和距離範圍進行交叉驗證，以確定亮度和照明為最佳組合，最終的檢測結果也有良好的準確率。Fan Wu[26]等人將 YOLOv3 做改良，利用 DenseNet[27]在模型參數上的優勢代替 YOLOv3 的主幹網路進行特徵提取，緩解了原始網路中檢測不準確和邊界框重疊的問題，而形成 YOLO-Densebackbone 卷積神經網路，與傳統的 YOLOv3 相比，改進的算法檢測精度提高了 2.44%；Yan Hongwen[28]等學者以真實養殖場景下的豬隻為研究對象，提出了一種融合通道注意力模塊(CAB)和空間注意力模塊(SAB)的 DAT-YOLO 模型，對 Tiny-YOLO 進行改進，DAT-YOLO 保留了 Tiny-YOLO 的多尺度特徵提取塊以保證對不同大小的臉部有較強的檢測性能，由於融合 CAB 和 SAB，可在不顯著提升計算量和參數量的前提下提升模型特徵提取性能，在臉部姿態類別中，原始的 Tiny-YOLO 平均準確率為 73.99%，而改進後的 DAT-YOLO 平均準確率為 82.38%。

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

圖 2 Darknet-53 模型

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w e^{t_w} \quad (3)$$

$$b_h = p_h e^{t_h} \quad (4)$$

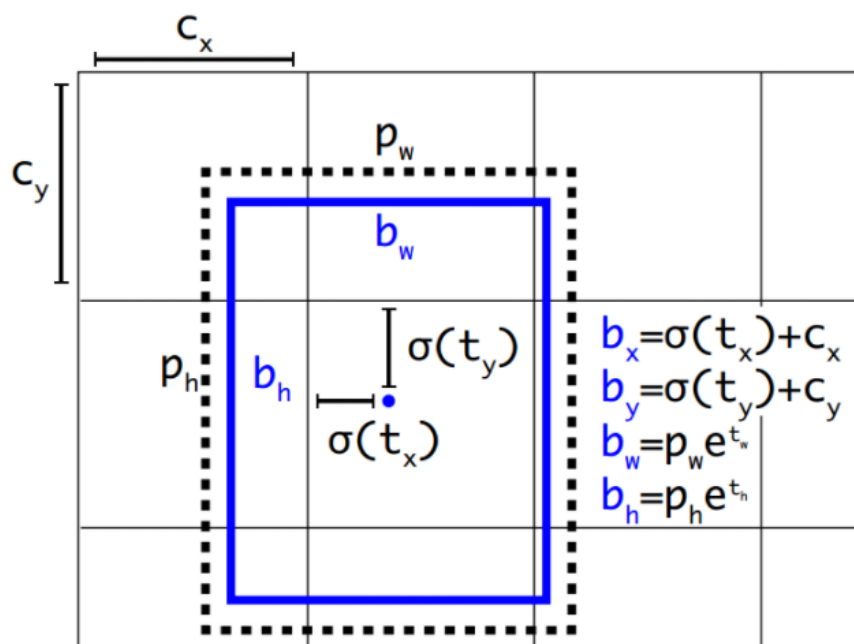


圖 3 YOLOv3 邊框預測

2.5 模型評估

在許多研究中，會使用各種方法來評量一個分類模型的好壞，在此研究中，使用最常見的混淆矩陣來參考各項指標，混淆矩陣是一種可視化的工具，目的是判斷模型中是否混淆了不同的類別，這也是 YOLO 模型中最常用的評估方法，如表 1 所示。

表 1 混淆矩陣

		True condition	
		Positive	Negative
Predicted condition	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

精確率(Precision)是代表預測為正確的樣本中有多少個是真正的正樣本。

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

召回率(Recall)是代表所有正樣本中，有多少個是正確判斷出來的正樣本。

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

F1 score 是測量精度的常用指標，常用於判斷算法的精度，同時考慮精確率和召回率兩個指標，體現了該算法的準確率。

$$F1 \text{ Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

mAP 是計算每個類別的精度並取平均值，k 代表的模型的類別，n 代表在該類別下的數量。

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (8)$$

IoU 是測量數據集中對象精度的標準。最重要的是計算預測的邊界框與真實邊界框的重疊區域，將重疊區域除以聯集區域得出最終分數，如果大於 0.5，則認為它是一個好模型。

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (9)$$

2.6 IFTTT

智慧家庭已經是目目前眾所皆知的技術之一，在現今的社會中，越來越注重便利、安全和隱私，而智慧家庭為家庭帶來更好的安全性、舒適性和便利性，有了智慧家庭系統，屋主可以輕鬆的監控家庭情況和控制家用電子產品，例如：照明設備、空調、門窗和掃地機器等。智慧家庭一直以來都是很熱門的話題，所以許多學者正在研究和開發一套完整的智能家居系統，像是 Martina Botticelli[29]等學者提出一種能夠提供監控環境參數、控制門窗和安全服務的智能居家系統，該系統在門窗上裝置感測器，當屋主不在屋內且偵

測到有陌生人移動門窗，系統將會發送警告訊息至屋主的手機，屋主也可以透過手機查看房屋的門窗目前的開關狀態。而 M. Udin Harun Al Rasyid[30]等學者建立一個以 arduino 作為節點和使用樹莓派為控制器的智慧家庭系統，該系統提供完整的 Web 介面，屋主可以從 Web 介面打開和關閉每個節點的電源，監控屋內的溫度和濕度，控制門窗，屋主可以在 Web 介面輸入自己的信箱和密碼，當屋內的運動感測器偵測到區域中的運動時，將會發送通知至屋主的手機。目前智能家居逐漸普及化，現在市面上也有許多公司提供智慧家庭平台服務，像是 HomeKit[31]是 Apple 的智慧家庭平台，HomeKit 可以輕鬆辨識和操作各種配件，支援達 20 種類的智慧裝置，包含插座、風扇、門鈴、保全等等，安全方面也是相當重視的一環。雖然市面上已經提供許多智慧家庭平台服務，但是許多平台侷限於使用特定的配件，以至於消費者為了配合平台服務，購買許多配件，而 IFTTT[32]則不會有這樣的顧慮，IFTTT 是 If this, then that 的縮寫，可以將不同的 APP、連網裝置和軟體服務整合在一起，IFTTT 可以使兩個服務(APP、連網裝置)連結上，提供內建的 Applet，可以輕鬆地達到智慧家庭自動化，IFTTT 支援的裝置非常眾多，像是家電類、智能窗簾、室內環境控制、智能照明、智能安全、智能中樞和橋接器等，IFTTT 能建立的 Applet 自由度相當高且能夠很好的透過時間、地點和感測器去觸發另一個智能裝置，所以在此研究中，利用 IFTTT 將樹莓派連接上手機和電腦，當偵測到貓咪有異常行為將立即發送通知至飼主的手機。

第三章 研究方法

3.1 研究架構

本研究提出的架構如同摘要所述，我們使用 YOLOv3-Tiny 模型對蒐集好的貓咪圖片進行測試，在文獻概述中的 YOLO 介紹中可以知道 YOLO 在影像辨識上具有較好的檢測精度。系統架構如圖 4，依序由編號介紹。

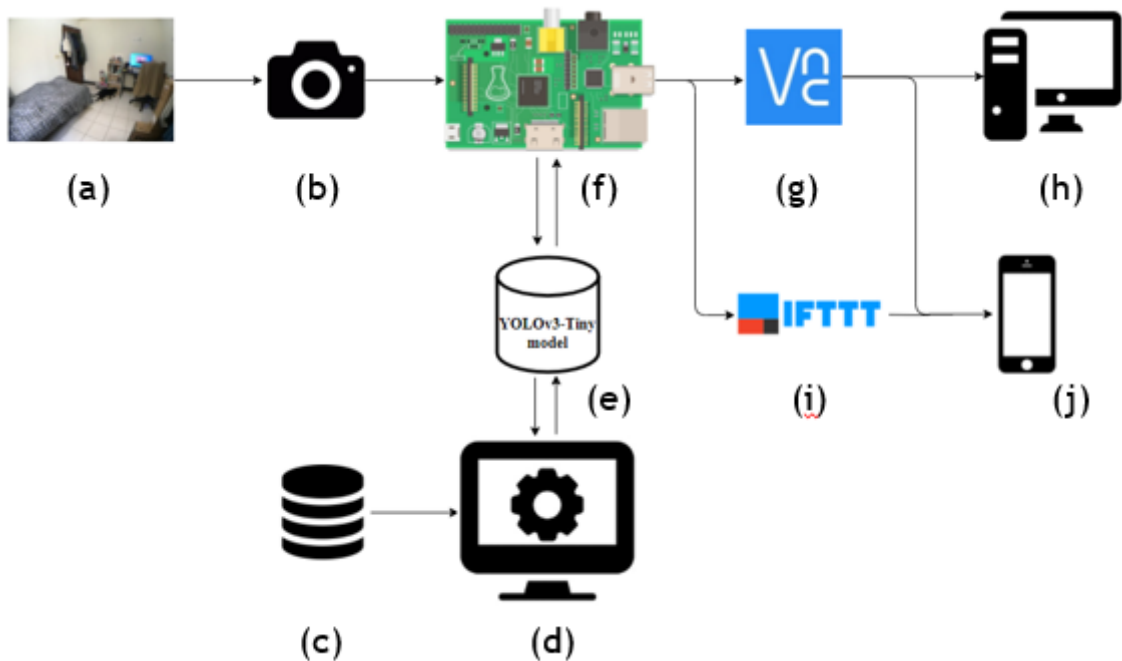


圖 4 研究架構

(a)為使用樹莓派攝像頭進行拍攝的畫面，為了能拍攝整個房間畫面，我們將樹莓派攝像頭架設在高處，藉此能夠拍攝到較完整的房間畫面；(b)為樹莓派攝像頭，考量到設備相容性問題，我們採用的是 Raspberry Pi Camera Module V2.1 做為輸入設備，其像素為 800 萬像素；(c)為此實驗所收集的數據集，我們總共收集了 1410 張照片，六種類別，分別為行走、吃飯、坐下、睡覺、上廁所和亂翻垃圾桶；(d)為電腦訓練模型，我們考量到在樹莓派上訓練模型會對硬體造成太大的負擔，因此我們先利用電腦訓練好模型在放入樹莓派中，在放入電腦訓練的過程中，我們先將資料做預先處理，

將所有照片都裁切成同樣大小，使用工具標示圖片座標和類別標籤，並且我們在 YOLOv3-Tiny 的參數上也有做修改，如圖 5 所示，batch 設為 64，batch 所代表的意思為每次學習採用的樣本資料，subdivisions 設為 4，subdivisions 代表一個 batch 在學習的過程中，拆成多個 mini-batch，可根據電腦的性能，增加 batch 的數量來減少訓練所花費的時間。Width 和 height 統一設為 416，angle 設為 20，在訓練的過程中會隨機旋轉 -20 度~20 度的圖片角度。Saturation 設為 1.5，在訓練的過程中會隨機改變 1 倍~1.5 倍的圖像飽和度。Exposure 設為 1.5，在訓練的過程中會隨機改變 1 倍~1.5 倍的圖像曝光度。Max_batches 是模型最大訓練次數，也是影響模型準確率最大的參數，我們多次嘗試調整 Max_batches 參數，在實驗結果也可以看到我們測試的實驗數據，最終訓練結果 Max_batches 參數設為 12000 有最良好的準確率。Step 為模型訓練到達設定次數時，學習率會產生變化，我們將 Step 參數設定 Max_batches 達 80%與 90%的時候，學習率產生變化，因為這樣的參數會有最好的準確率。在設定模型參數時，要特別注意修改 YOLO 層前一層的捲基層輸出通道數，此輸出通道數是依據模型訓練的類別數量做調整，因此我們在 filters 設定為 33。在模型的 data 參數，由於我們的類別有行走、吃飯、坐下、睡覺、上廁所和亂翻垃圾桶等六種類別，因此我們的 classes 設為 6，訓練的資料在前面的過程也先做過裁切並使用工具標示圖片座標和類別標籤，處理完成並放入指定路徑，如圖 6 所示。

```

6 batch=64
7 subdivisions=16
8 width=416
9 height=416
10 channels=3
11 momentum=0.9
12 decay=0.0005
13 angle=20
14 saturation = 1.5
15 exposure = 1.5
16 hue=.1
17
18 learning_rate=0.001
19 burn_in=1000
20 max_batches = 12000
21 policy=steps
22 steps=9600,10800
23 scales=.1,.1
24
105 #####
106
107 [convolutional]
108 batch_normalize=1
109 filters=256
110 size=1
111 stride=1
112 pad=1
113 activation=leaky
114
115 [convolutional]
116 batch_normalize=1
117 filters=512
118 size=3
119 stride=1
120 pad=1
121 activation=leaky
122
123 [convolutional]
124 size=1
125 stride=1
126 pad=1
127 filters=33
128 activation=linear
129
130

```

圖 5 YOLOv3-Tiny-cfg 參數

```

1 classes= 6
2 train = data/shiantsau-tiny/train.txt
3 valid = data/shiantsau-tiny/test.txt
4 names = cfg/Shiantsau-tiny.names
5 backup = Shiantsau-tiny/

```

圖 6 YOLOv3-Tiny-data 參數

(e)為最終訓練完成的 YOLOv3-Tiny 模型；(f)為樹莓派，規格採用 Raspberry Pi 4 Model B V1.2 版，相比樹莓派三代，在四代的性能上，各方面都有明顯的提升，我們將訓練好的模型放入樹莓派，利用樹莓派和攝像頭來偵測貓咪的行為動作，搭配撰寫好的行為偵測程式碼，如圖 7 和圖 8 所示，該程式碼是將樹莓派偵測到的資料傳送至 IFTTT 並將訊息發送至使用者手機，其中定義了三個函數，第一個函數為 `postMessageToLoine`，該函數將偵測的貓咪的名字、當下的行為、準確率和照片記錄下來，透過網址的方式傳送至 IFTTT，再藉由 IFTTT 連結 LINE Notify 將訊息發送至手機，如圖

9 所示，如此一來，當貓咪上廁所超過 30 秒或是亂翻垃圾桶時，我們就能從 LINE 得知即時訊息。第二個函數 `getAction` 為利用抓取樹莓派偵測到的資料。第三個函數為 `upload_imgur`，該函數用來儲存圖片至圖庫使用的；(g) 為 VNC(Virtual Network Computing[33])遠端監控影像，VNC 是一套免費且開放原始碼的遠端遙控工具，VNC 可分為客戶端(VNC Viewer)和伺服器端(VNC Server)，將樹莓派上安裝 VNC Server 設定 IP 和密碼，並在電腦上和手機安裝 VNC Viewer 作為監控端，如圖 10 所示；(i)的部分為 IFTTT(If this, then that) 網路服務平台，將偵測到的行為影像傳送到手機，IFTTT 對於需要設計自動化智能家庭的人來說是很棒的一個平台，同樣的也可以應用在寵物辨識上，可以很好的將 APP、連網裝置和軟體整合在一起，本研究利用 IFTTT，當貓咪上廁所超過 30 秒或是亂翻垃圾桶，系統將及時發送訊息至飼主的手機。

```
count = 0
while True:
    try:
        select.select([yolo_proc.stdout], [], [])
        #讀取yolo_proc資料路徑
        stdout = yolo_proc.stdout.read()

        if len(stdout.strip()) > 0:
            current_action, action_correct = getAction(stdout)
            count += 1

            if current_action != last_action:
                last_action = current_action
                count = 0

            elif count >= 0.1 * VIDEO_FPS:
                myScreenshot = pyautogui.screenshot()
                myScreenshot.save(r'./{}.png'.format(count))
                image_link = upload_imgur('./{}.png'.format(count))
                pos(MessageToLine("{}{}%", {}).format(CAT_NAME, action_correct, ACTION_DICT[current_action], current_action), image_link)
                print("Screenshoot")
                count = 0

    except Exception as e:
        print("Error:", e)
```

圖 7 訊息發送程式碼-1

```

def postMessageToLine(message, image_url):
    post_url = "https://maker.ifttt.com/trigger/%E4%BB%99%E8%BD%89%E5%9D%90%E4%B8%B/with/key/b6cmY6nuhpKrij7dvlambDf"
    requests.post(post_url, data={"value1": message, "value2": image_url})

def getAction(stdout):
    regex = r'([A-Za-z]*): ([0-9]*)%'

    #找尋所有stdout回傳的文字
    action = re.findall(regex, stdout)
    if action[0] is not None:
        cat_action = action[0][0]
        action_correct = action[0][1]

    #關鍵字
    if cat_action in VALID_ACTION:
        return cat_action, action_correct

def upload_imgur(image_name):
    imgur_upload_url = "https://api.imgur.com/3/image"
    payload = {'image': open(image_name, 'rb')}
    headers = {
        'Authorization': 'Client-ID {}'.format(IMGUR_CLIENT_ID)
    }
    response = requests.request('POST', imgur_upload_url, headers=headers, files=payload)
    try:
        link = response.json()
        return link["data"]["link"]
    except KeyError:
        link = "Failed"
        return link

```

圖 8 訊息發送程式碼-2

The screenshot displays the IFTTT app interface. On the left, a workflow is shown with two steps: "If" (Receive a web request) and "Then" (Send message to LINE). A "Continue" button is at the bottom. On the right, the configuration for the "Send message" action is shown. The "Event Name" is set to "貓咪動作". The "Recipient" is set to "透過1對1聊天接收LINE Notify的通知". The "Message" field contains "Value1". The "Photo URL (optional)" field contains "Value2".

圖 9 IFTTT 連結 Notify

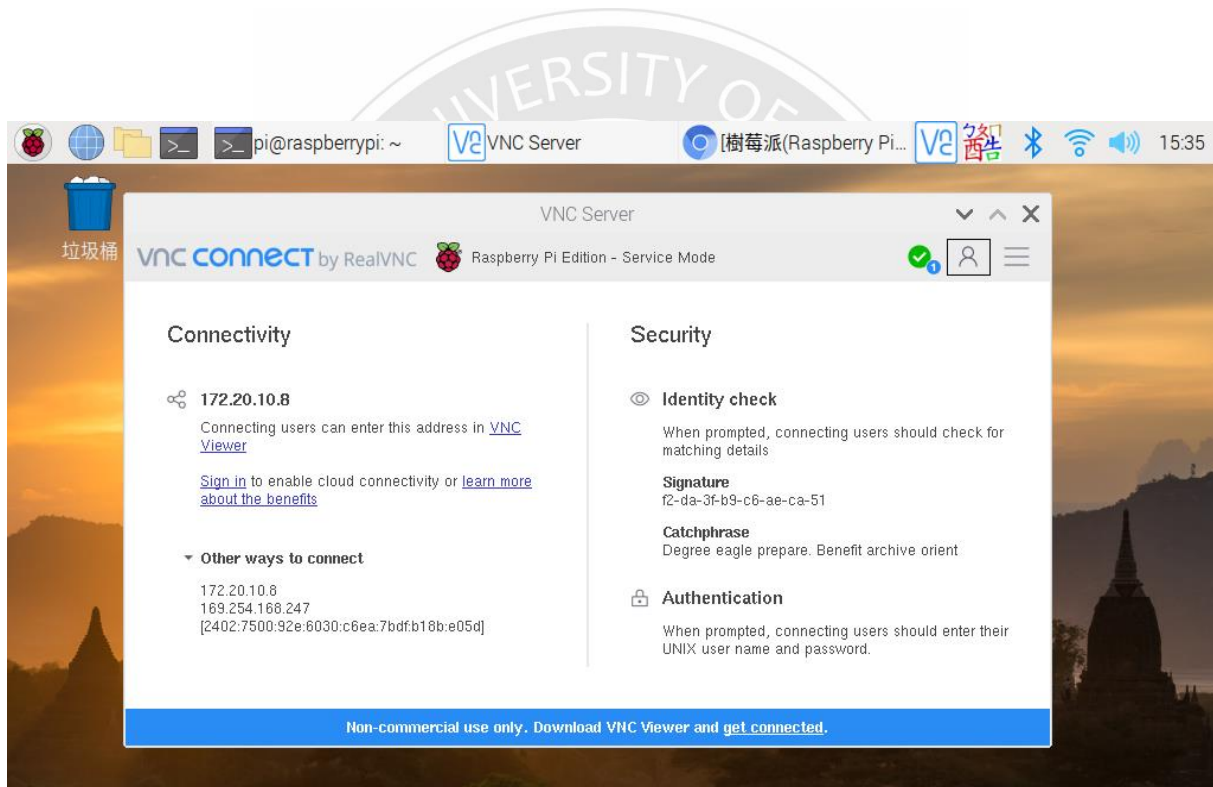


圖 10 樹莓派安裝 VNC Server

3.2 行為偵測

在大部分的寵物行為偵測研究中，往往只有單純偵測寵物行為或是寵物品種而沒有進一步的研究，所以在此研究中，希望當寵物發生特定行為時，飼主能在第一時間知道並做出相對應的處理，於是我們建立一套能夠偵測寵物行為的系統，並且當寵物發生上廁所過久或是亂翻垃圾桶等行為時，將即時傳送訊息和圖像至飼主手機，讓飼主能夠即時了解寵物的情況，以預防寵物發生狀況。

在此篇研究中，考量到在樹莓派上訓練 YOLOv3-Tiny 模型會造成負擔且花費過久的訓練時間，所以我們在電腦上先訓練好模型再將模型放入樹莓派中進行測試。寵物行為偵測流程圖如圖 11，首先開啟樹莓派並使用命令提示字元執行編寫完成的程式碼，當樹莓派偵測到行為，根據行為做出相對應的動作，若偵測到吃飯、行走、坐下或睡覺不會發出通知，但是當偵測到上廁所超過 30 秒或是翻垃圾桶等動作時，系統將會將影像儲存下來，並即時將圖像和行為發送至飼主手機，當然飼主也能透過手機和電腦串流樹

莓派查看當前的偵測影像。

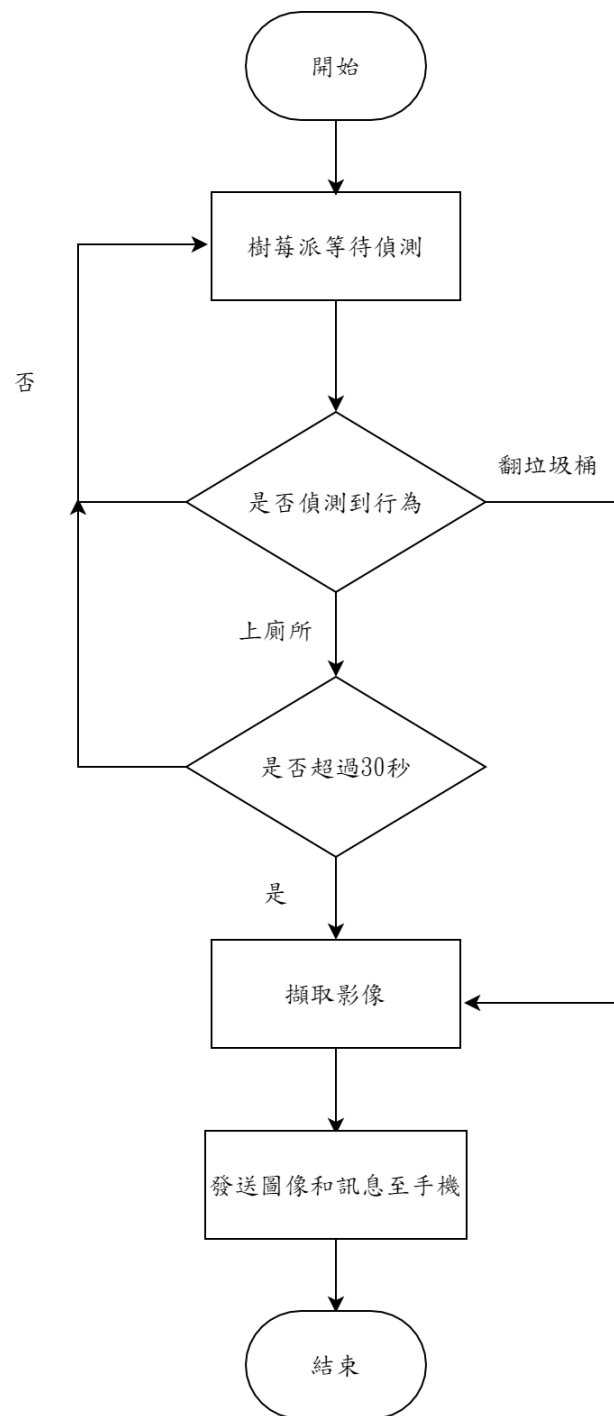


圖 11 寵物異常行為偵測流程圖

3.3 YOLOv3-Tiny

本研究因考量樹莓派硬體上的配置，採用 YOLOv3-Tiny 作為物體偵測的模型，YOLOv3-Tiny 簡單來說就是 YOLOv3 的簡化版本，雖然在檢測速度和檢測精度上不如 YOLOv3，但是在此研究最終檢測結果上，還是有很好的表現。相比於 YOLOv3-Tiny 版本將網路壓縮許多，省去了一些特徵層且沒有使用殘差層，只使用了 13×13 和 26×26 兩個不同尺度的 YOLO 輸出層，也因為 YOLOv3-Tiny 的主幹網路比較淺，所以無法提取出更高層次的語義特徵，但 YOLOv3-Tiny 的優勢在於網路簡單且計算量負擔較小，方便使用在像樹莓派等可移動裝置上運行，YOLOv3-Tiny 網路如圖 12。

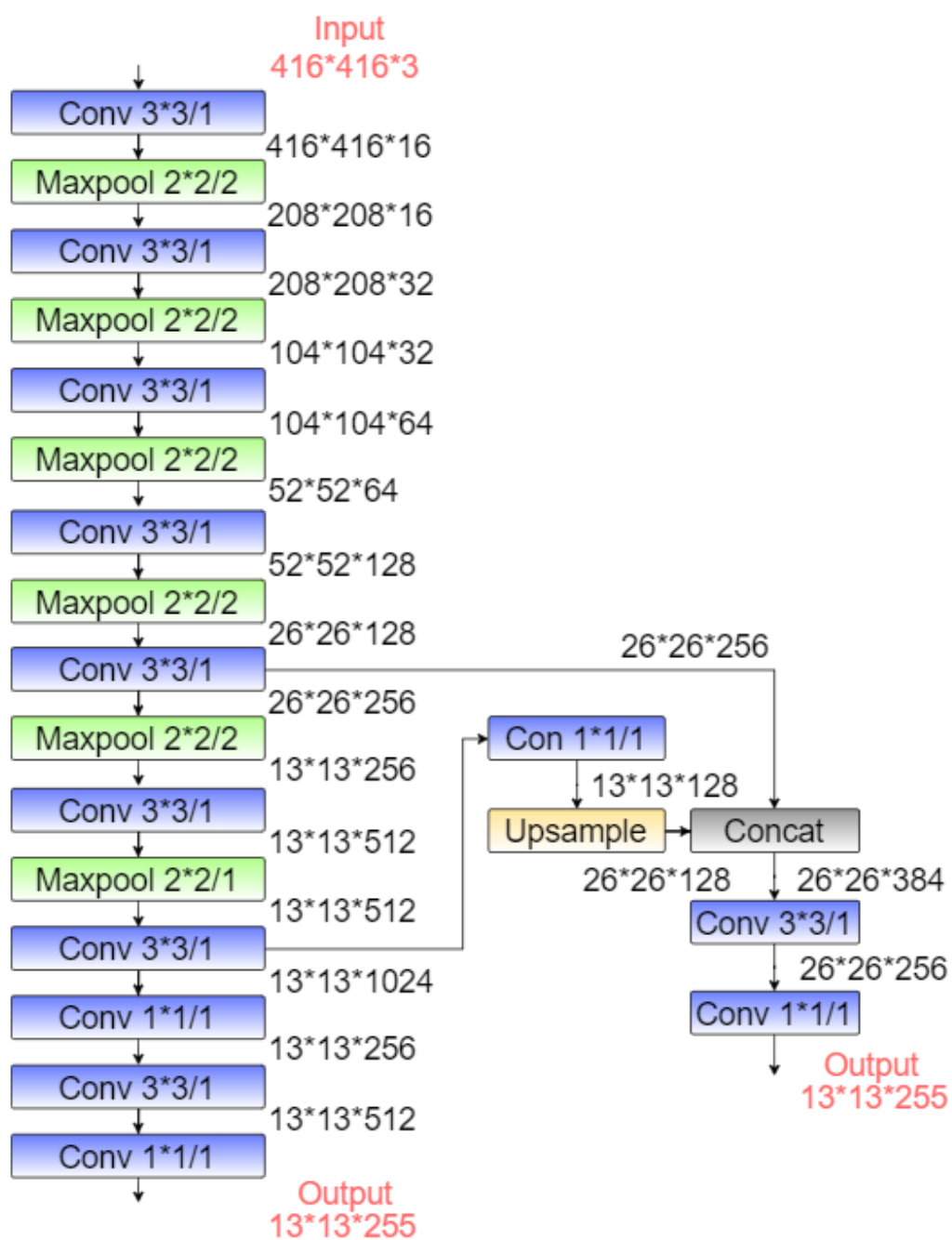


圖 12 YOLOv3-Tiny 主幹網路

第四章 實驗結果與分析

4.1 實驗環境

實驗數據來自於宿舍內的客廳和房間的手機拍攝，蒐集到的圖片統一裁剪成 416x416 像素。為了使數據集更準確，將圖片角度隨機旋轉正負 20 度內，以及在不同時段拍攝，使數據集在訓練上更具有代表性。

本研究硬體設備使用 Raspberry Pi 4 Model B V1.2，記憶體為 2G LPDDR4-2400 SDRAM，儲存空間採用 microSD 64G，攝影機使用 Raspberry Pi Camera Module V2.1，其規格為 800 萬像素。

4.2 數據集

最後總共蒐集了 1410 張照片，在訓練過程中，我們使用 bounding box labeling tool(BBox Label Tool[34]),手動標示檢測目標的座標位置，該工具將返回座標的四個點及類別標籤，如圖 13 所示。數據集由 6 個類別組成，行走、進食、睡覺、坐下、上廁所和亂翻垃圾桶。每個類別的訓練圖像及測試圖像為 7 比 3。



圖 13 BBox Label Tool 手動標示圖片

4.3 實驗結果

在此實驗中，為了能夠訓練出最好的模型，我們在 YOLOv3-Tiny 模型參數上進行了多次的調整和訓練，分別將 max_batches 設為 9000 次、12000 次和 15000 次訓練了 10 次，以確保實驗的準確性，損失值是損失函數的一部分，數量越少越好，而 mAP 則是越高越好。圖 14 至圖 23 是 max_batches 設為 9000 次的損失函數和 mAP，在使用 max_batches 設為 9000 次的參數設定中，可以發現平均損失較高，平均十次的損失函數為 0.1234，而 mAP 則為 96.71%。圖 24 至圖 33 是 max_batches 設為 12000 次的損失函數和 mAP，平均十次的損失函數為 0.1029，mAP 則為 97.84%，可以發現 max_batches 設為 12000 次相較於 max_batches 設為 9000 次的損失函數和 mAP 都是較為良好的。圖 34 至圖 43 是 max_batches 設為 15000 次的損失函數和 mAP，平均十次的損失函數為 0.0954，而 mAP 為 97.26%，可以發現 max_batches 設為 15000 次的 mAP 是低於 max_batches 設為 12000 次的，也證明了模型的培訓步驟並不是越多越好，考量的訓練模型的時間，我們認為 max_batches 設為 12000 次是最良好的參數設定，而表 2 至表 4 為 max_batches 設為 9000 次、12000 次和 15000 次的各項數據結果。

為了進一步驗證 YOLOv3-Tiny 模型的性能，我們同時訓練 YOLOv3 模型進行比較，使用相同的訓練集和測試集。圖 44 為 YOLOv3 的平均損失和 mAP，我們同樣將 max_batches 設為 12000 次培訓步驟來更好地分析培訓過程，損失值為 0.0367，而 mAP 為 98.3%，mAP 與 YOLOv3-Tiny 相當接近，損失值相差較多，雖然 YOLOv3-Tiny 主幹網路較淺，但是其性能還是相當良好。而圖 45 和圖 46 為比較兩個模型的檢測速度，可以發現 YOLOv3 模型的平均幀數高於 YOLOv3-Tiny 模型相當多。圖 47 為行走類別被偵測為翻垃圾桶類別，從混淆矩陣中可以看出這張圖像被歸類為 FP，我們認為可以加強圖片的數據集藉此讓準確率提升也能降低預測錯誤的情況，

YOLOv3-Tiny 模型測試結果如圖 48 至圖 53 所示。當上廁所超過 30 秒以及貓咪翻垃圾桶時，將會發送訊息至飼主的手機，如圖 54 和圖 55 所示。飼主可以透過手機或電腦遠端查看貓咪目前行為，如圖 56 和圖 57 所示。

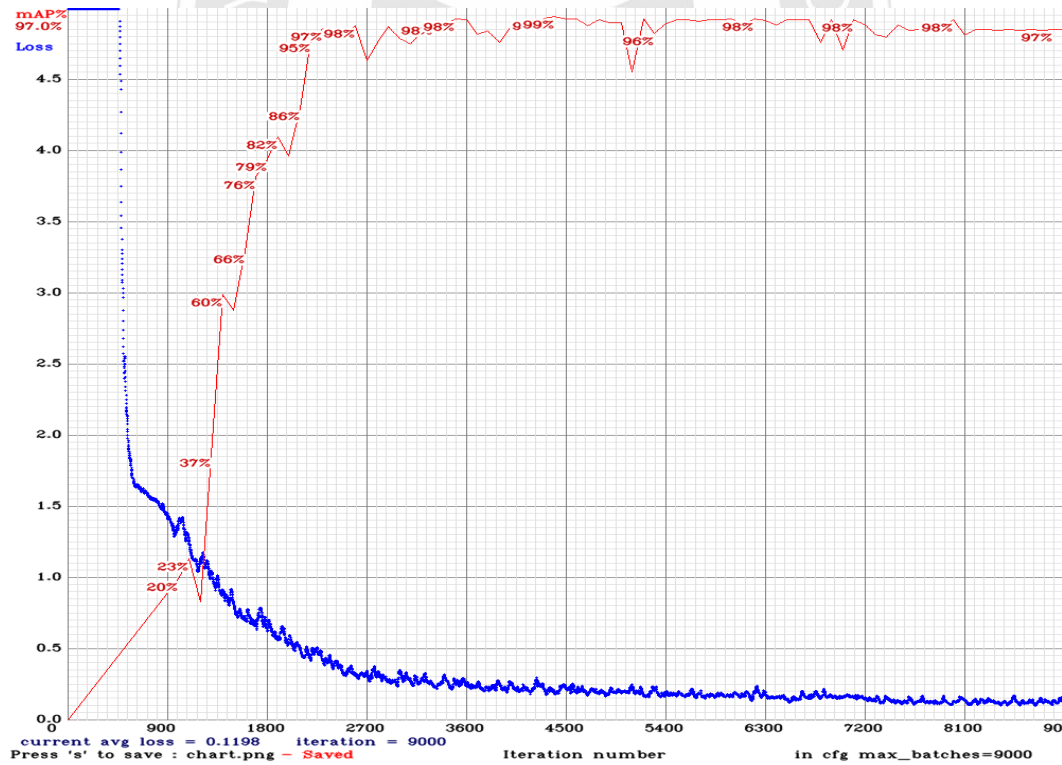


圖 14 YOLOv3-Tiny 9000 次培訓步驟-1

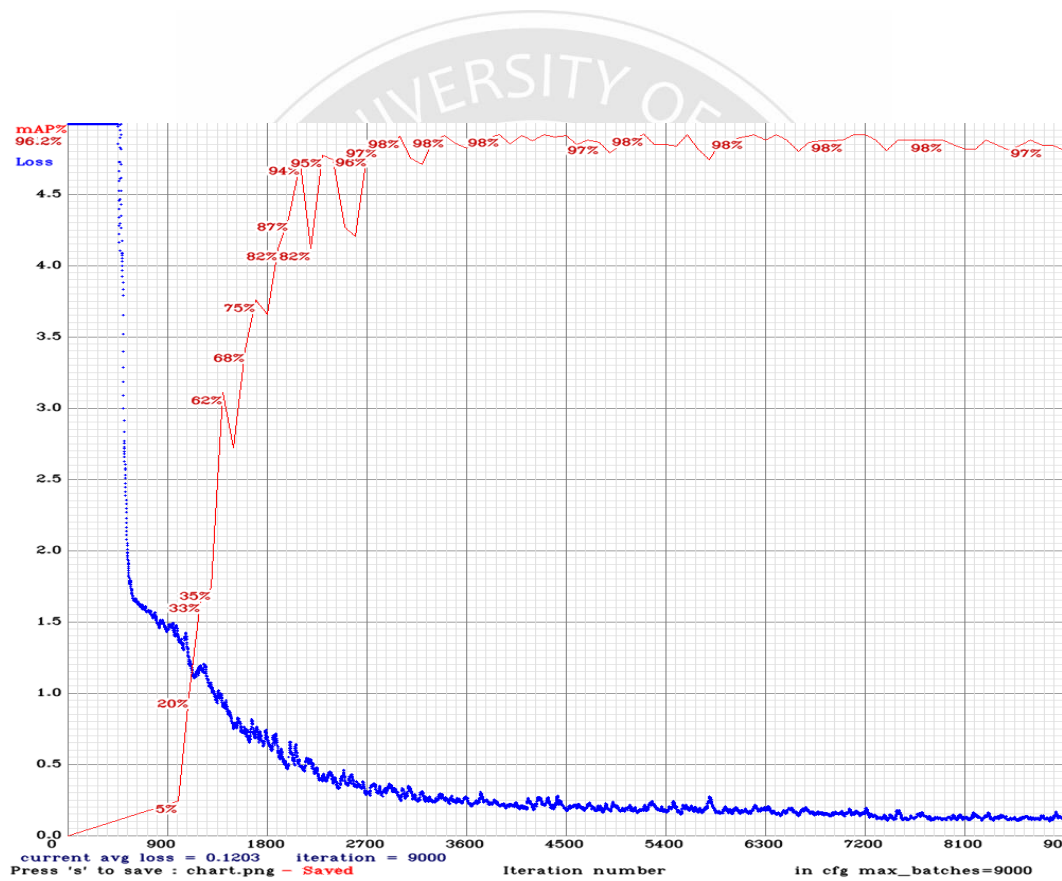


圖 15 YOLOv3-Tiny 9000 次培訓步驟-2

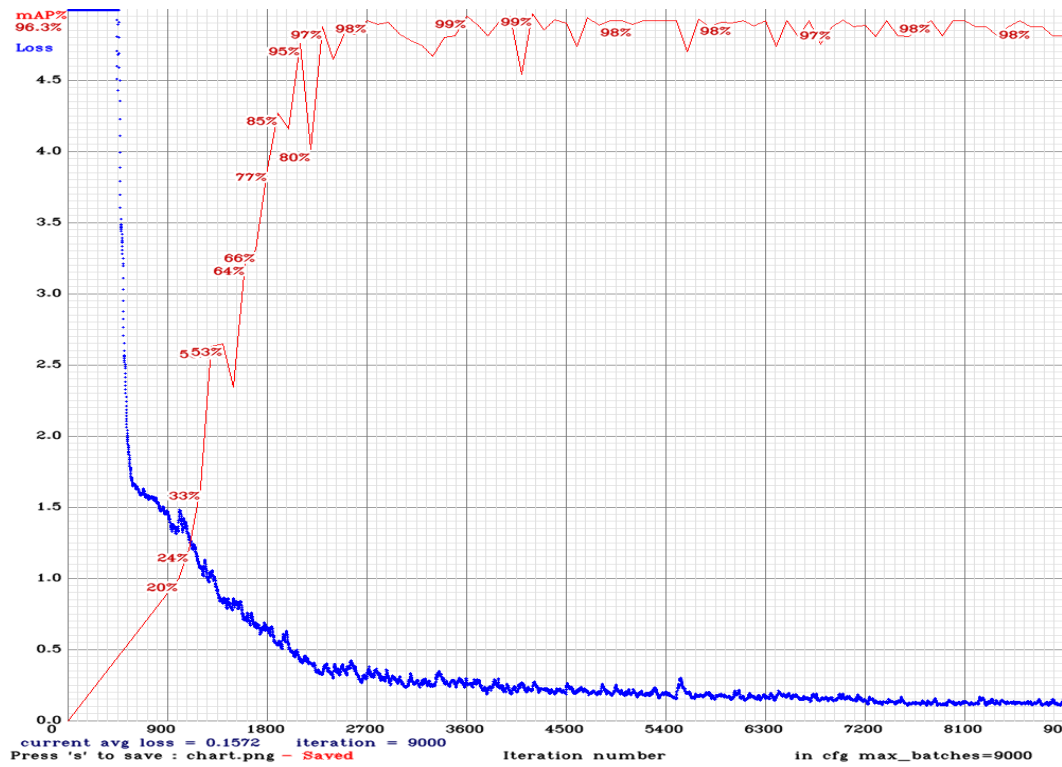


圖 16 YOLOv3-Tiny 9000 次培訓步驟-3

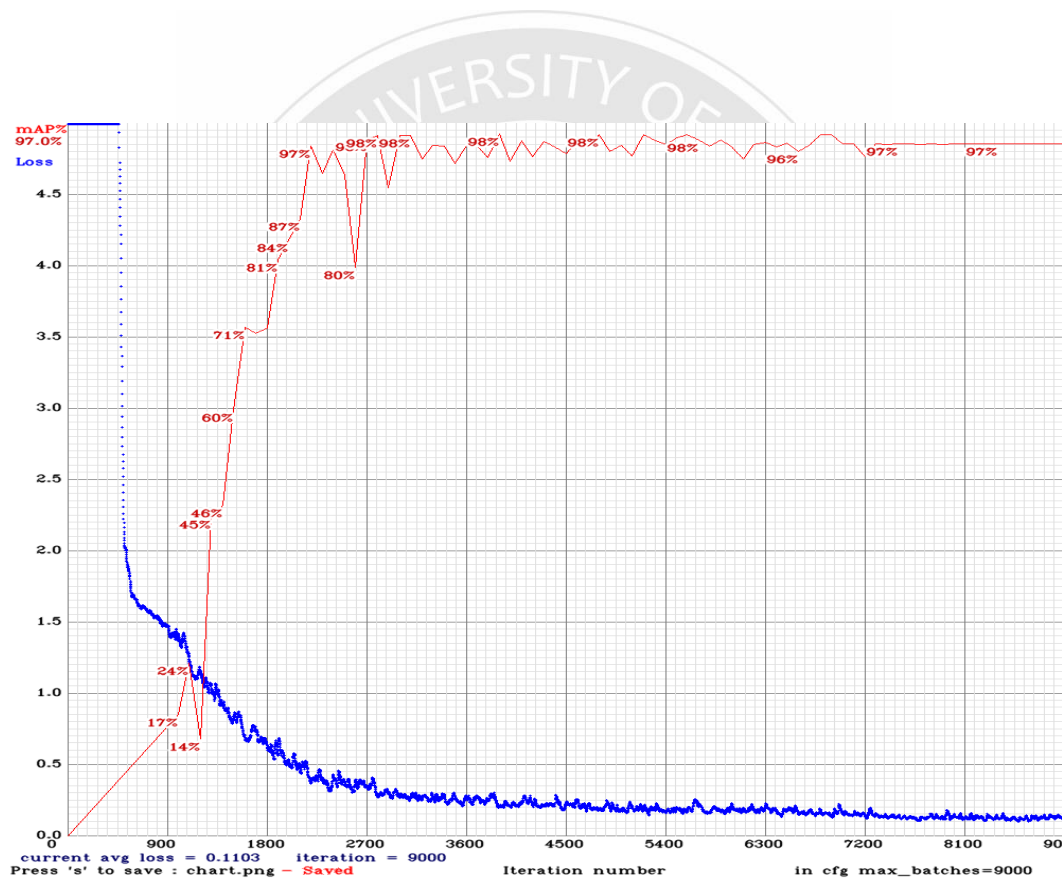


圖 17 YOLOv3-Tiny 9000 次培訓步驟-4

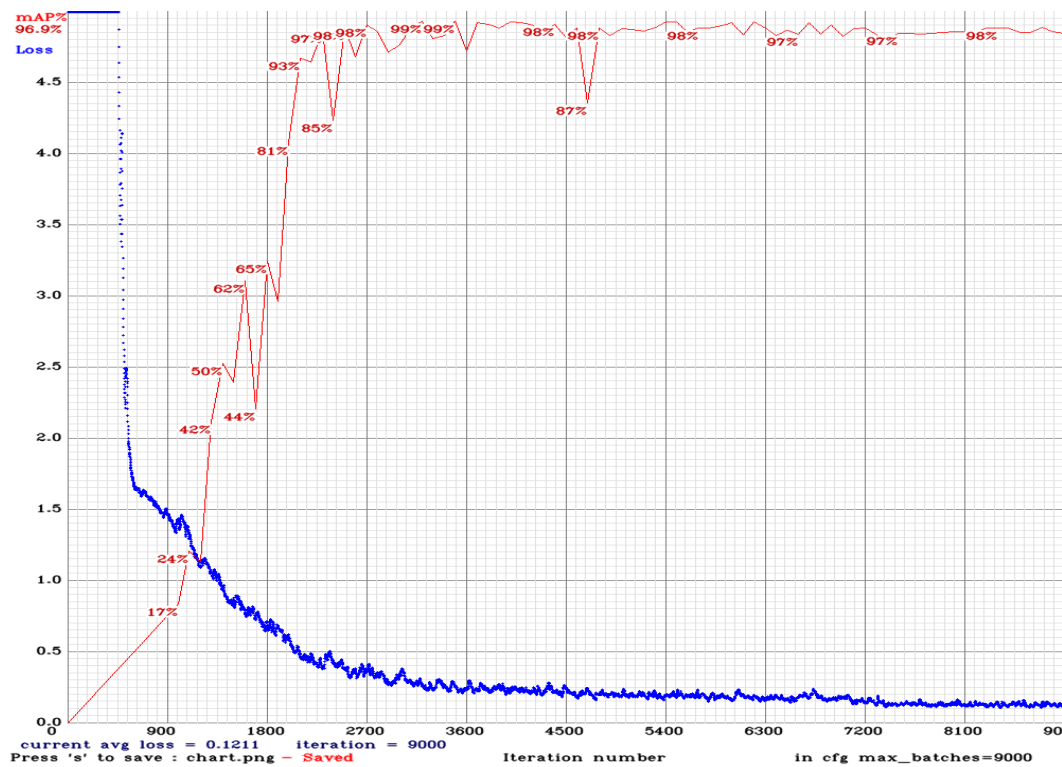


圖 18 YOLOv3-Tiny 9000 次培訓步驟-5

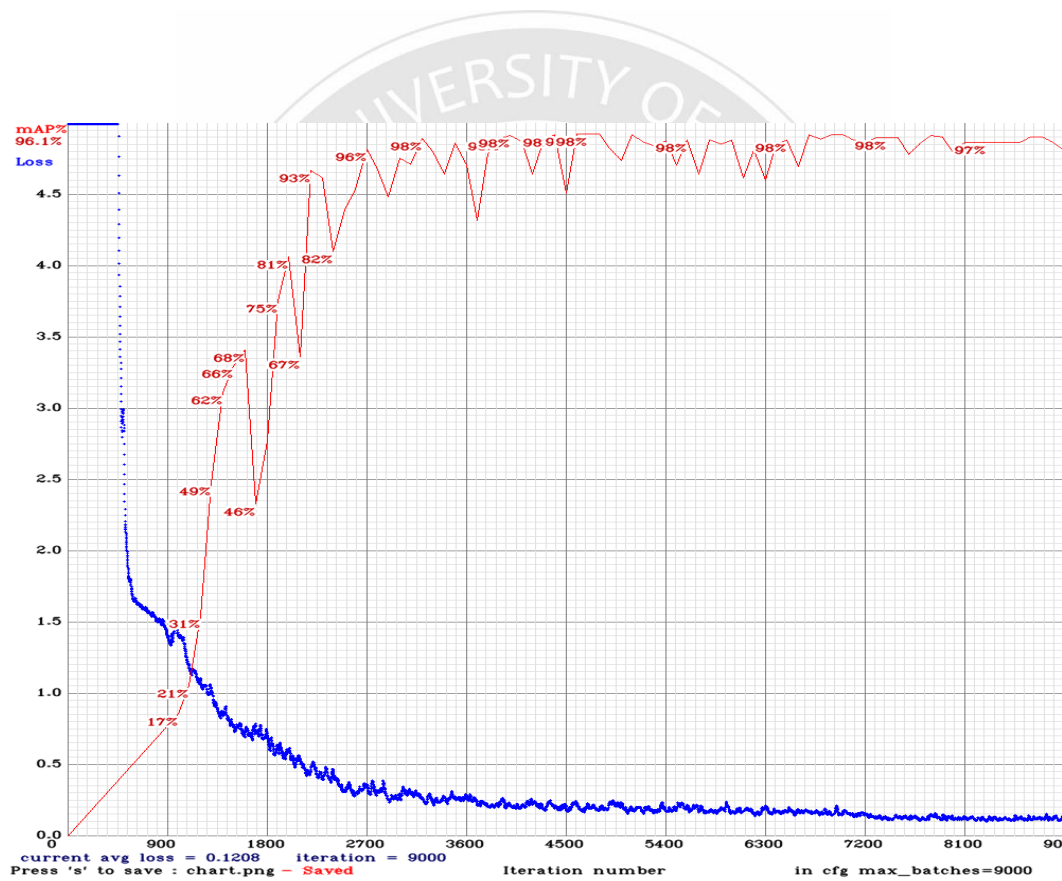


圖 19 YOLOv3-Tiny 9000 次培訓步驟-6

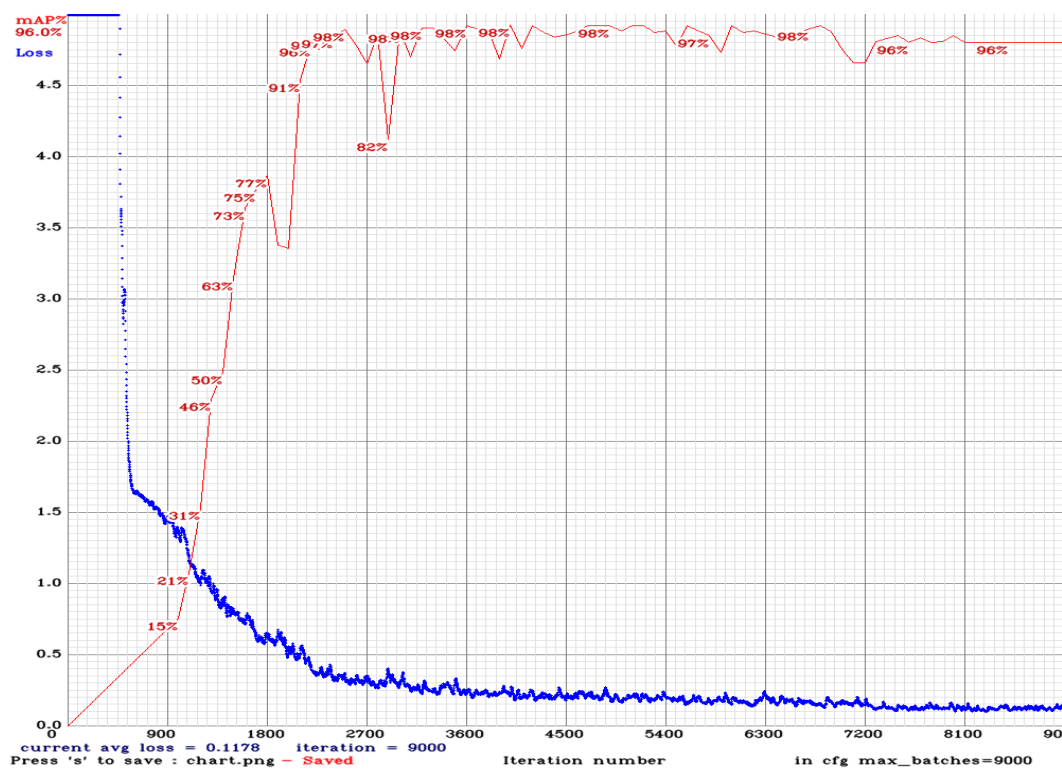


圖 20 YOLOv3-Tiny 9000 次培訓步驟-7

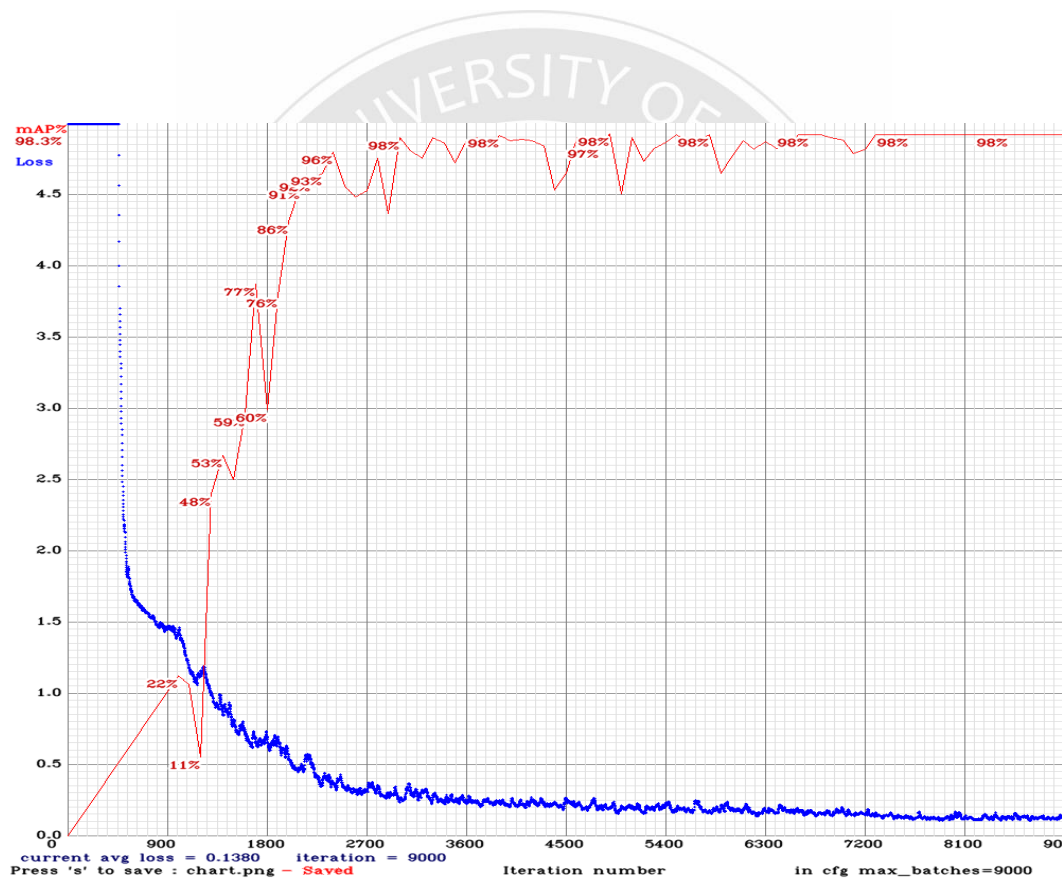
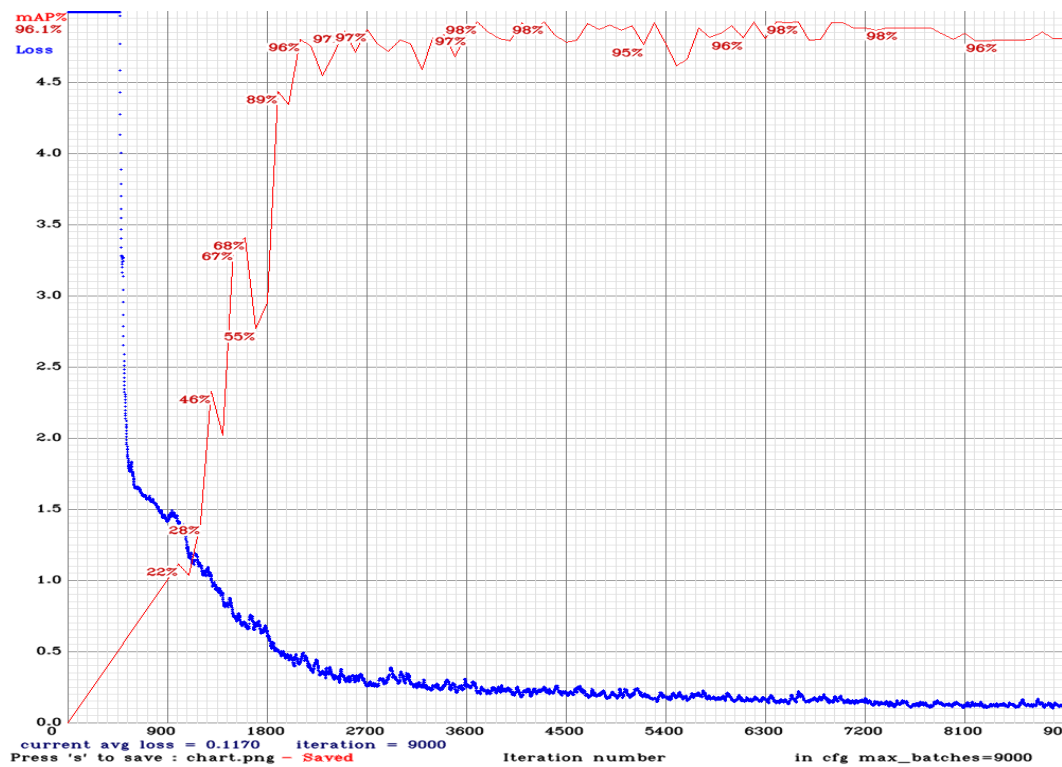


圖 21 YOLOv3-Tiny 9000 次培訓步驟-8



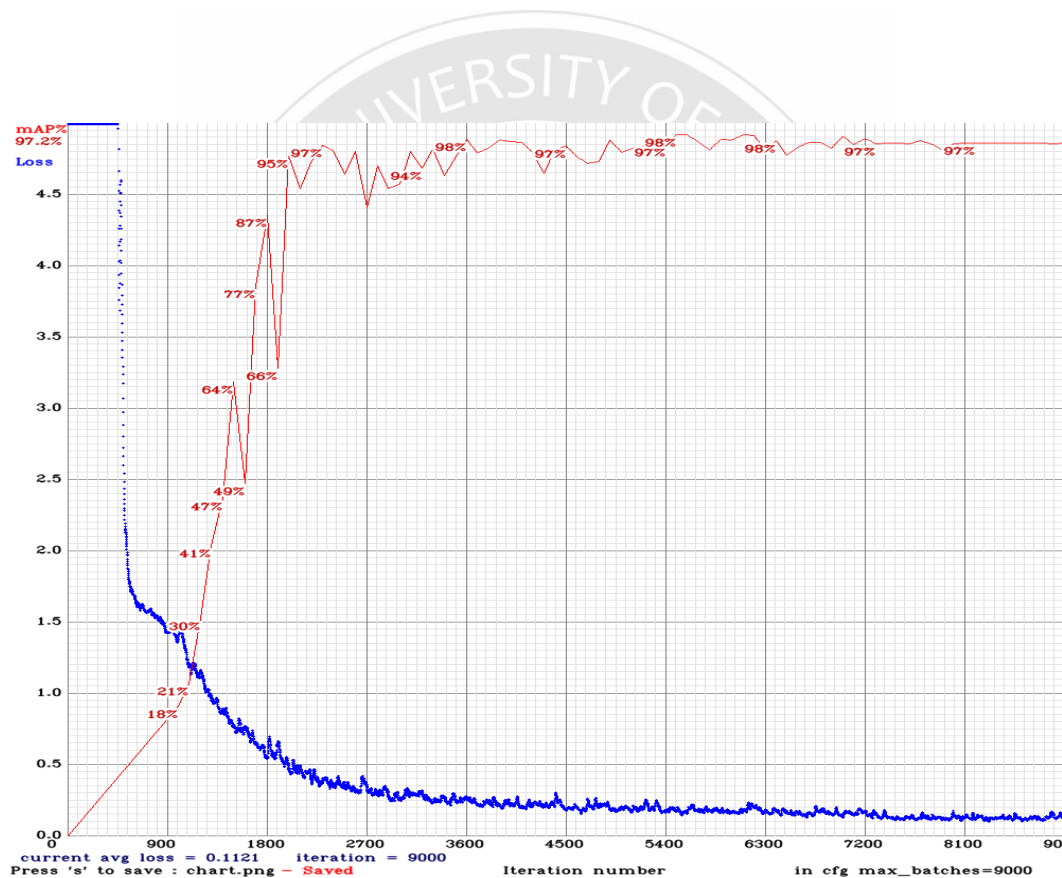


圖 23 YOLOv3-Tiny 9000 次培訓步驟-10

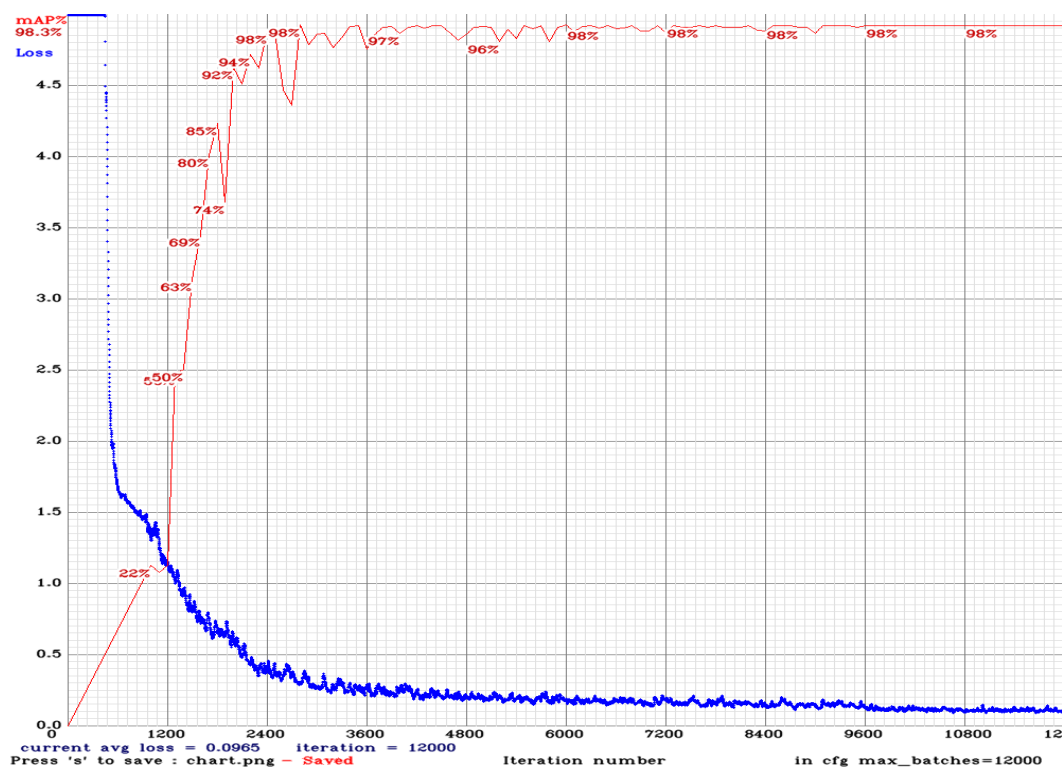


圖 24 YOLOv3-Tiny 12000 次培訓步驟-1

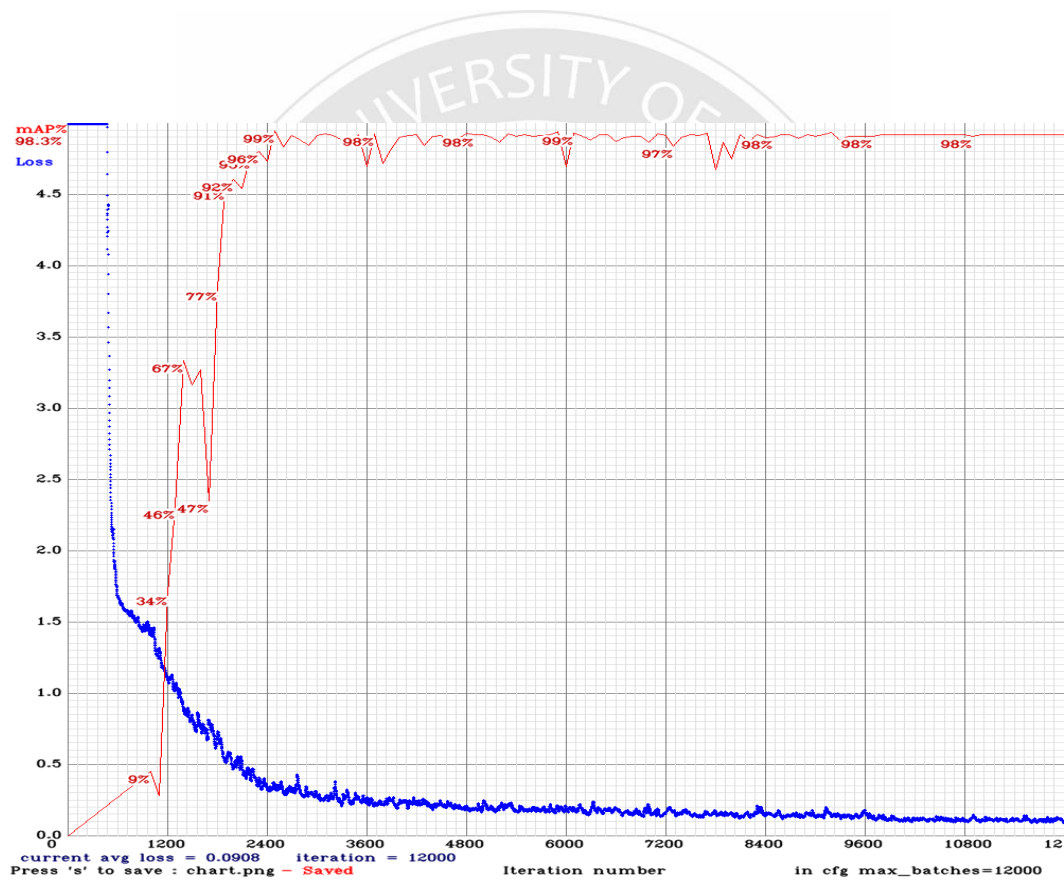


圖 25 YOLOv3-Tiny 12000 次培訓步驟-2

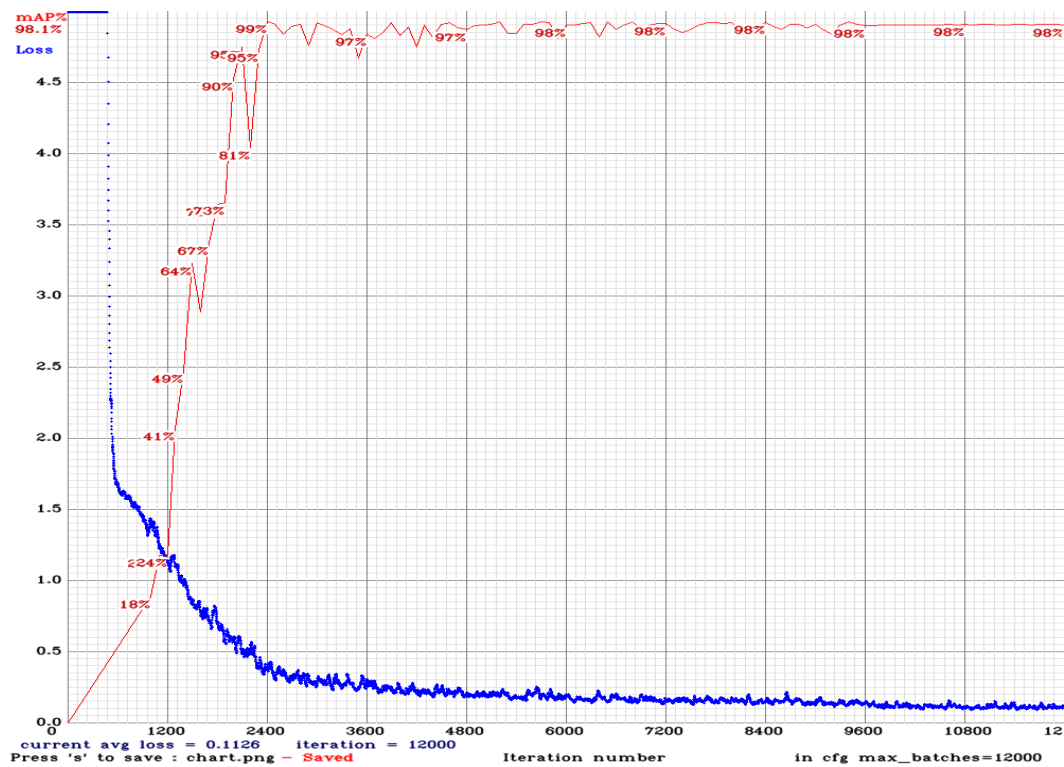


圖 26 YOLOv3-Tiny 12000 次培訓步驟-3

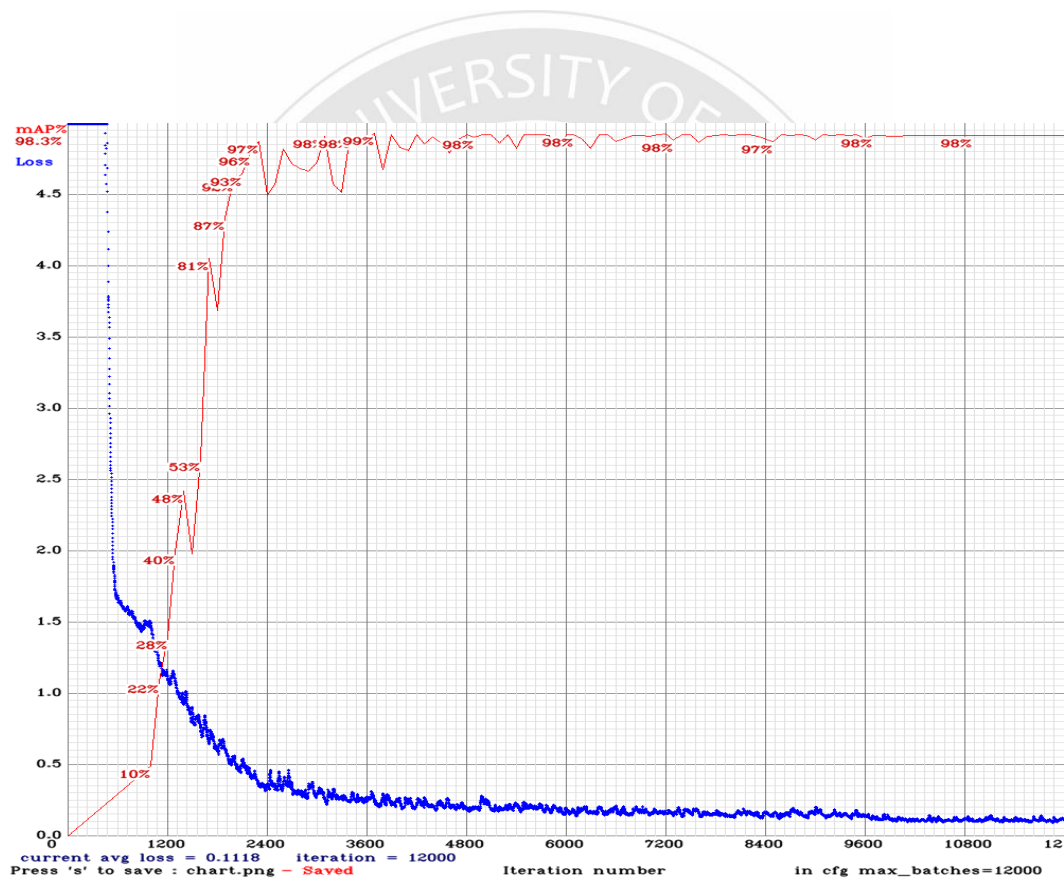


圖 27 YOLOv3-Tiny 12000 次培訓步驟-4

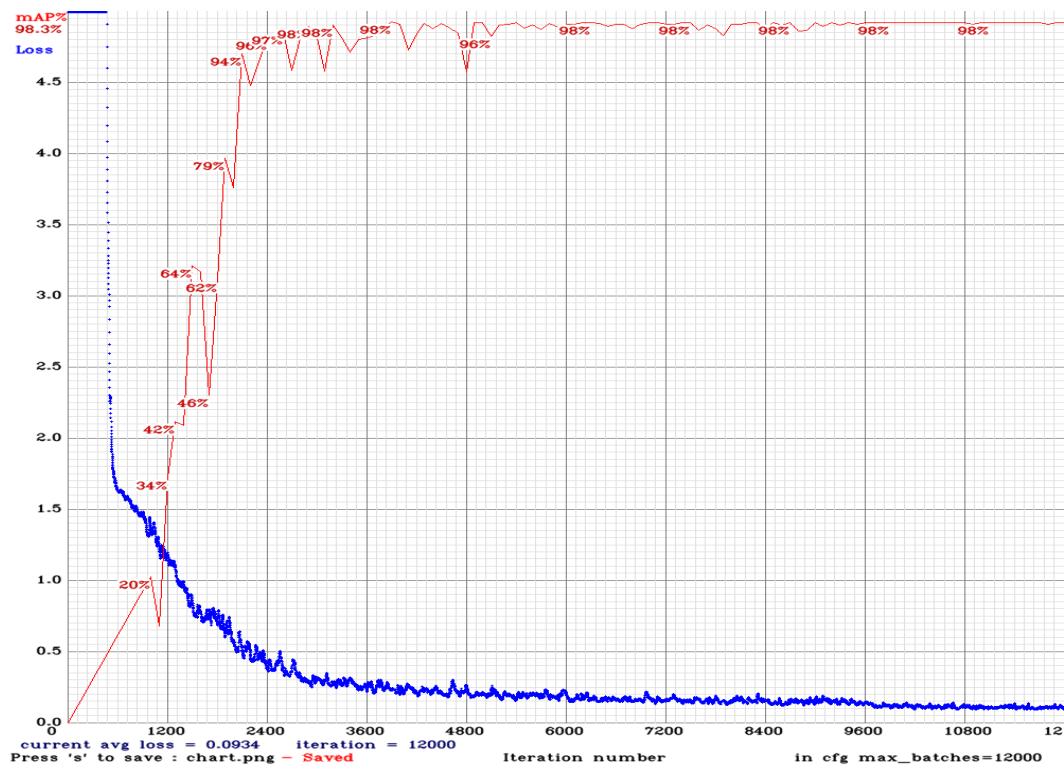


圖 28 YOLOv3-Tiny 12000 次培訓步驟-5

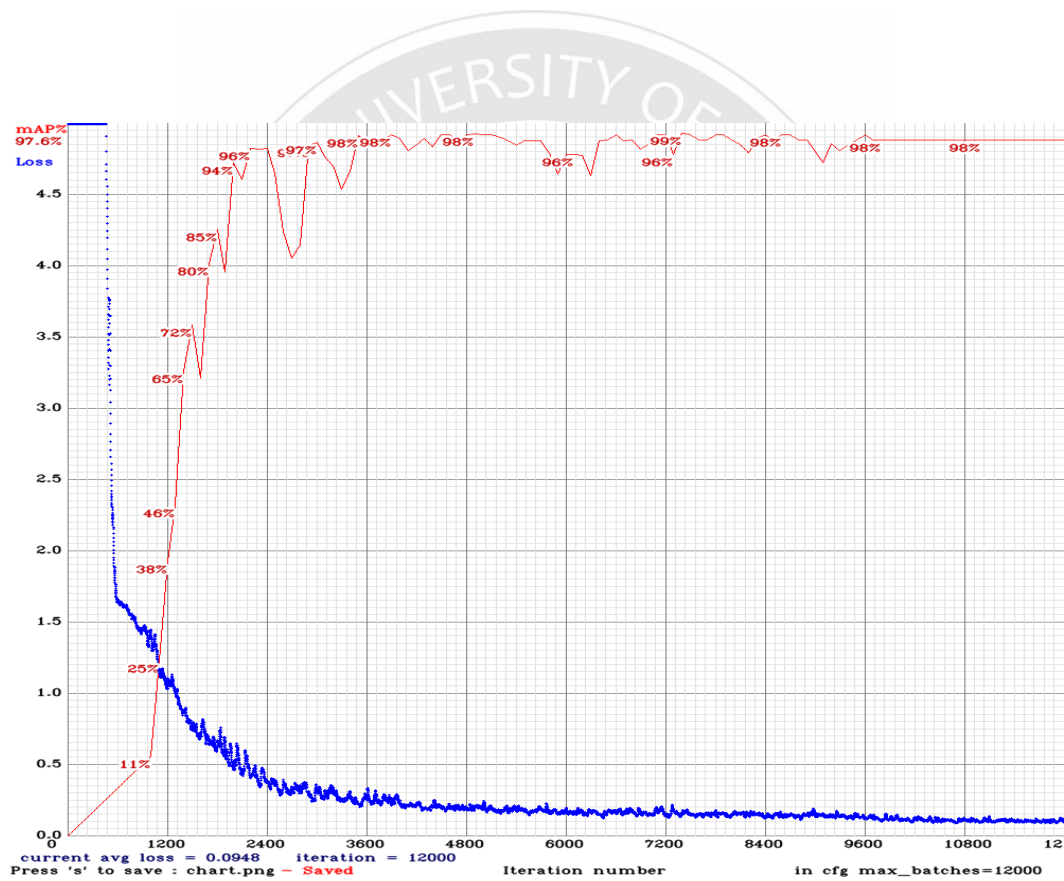


圖 29 YOLOv3-Tiny 12000 次培訓步驟-6

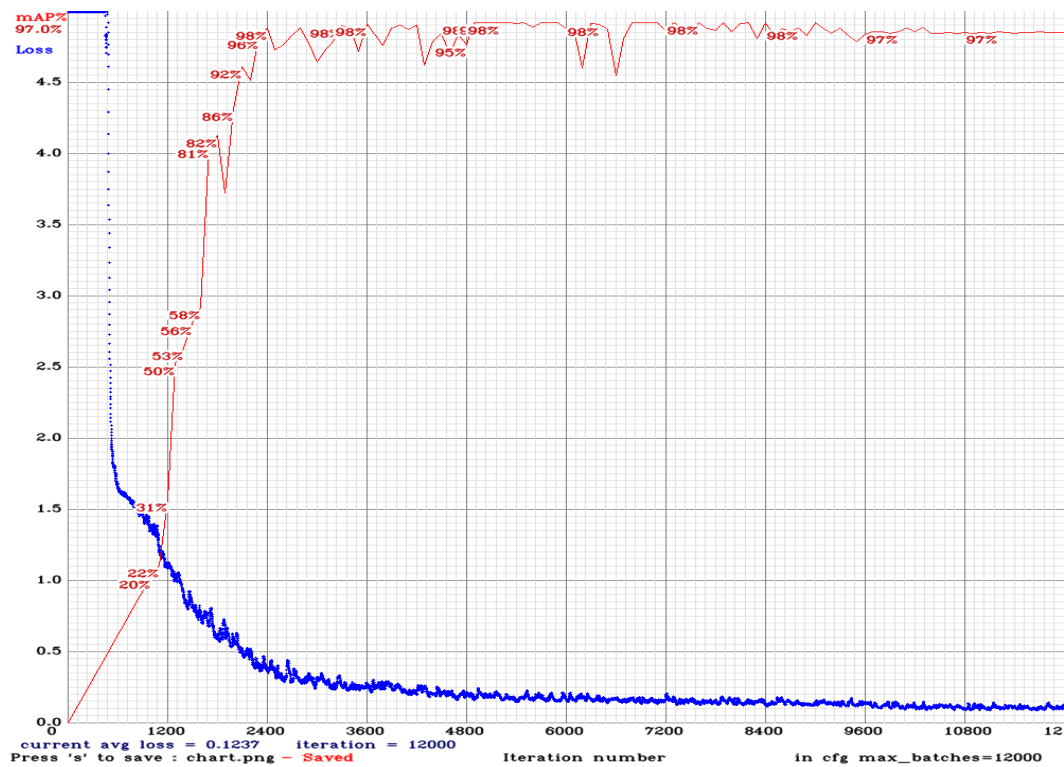


圖 30 YOLOv3-Tiny 12000 次培訓步驟-7

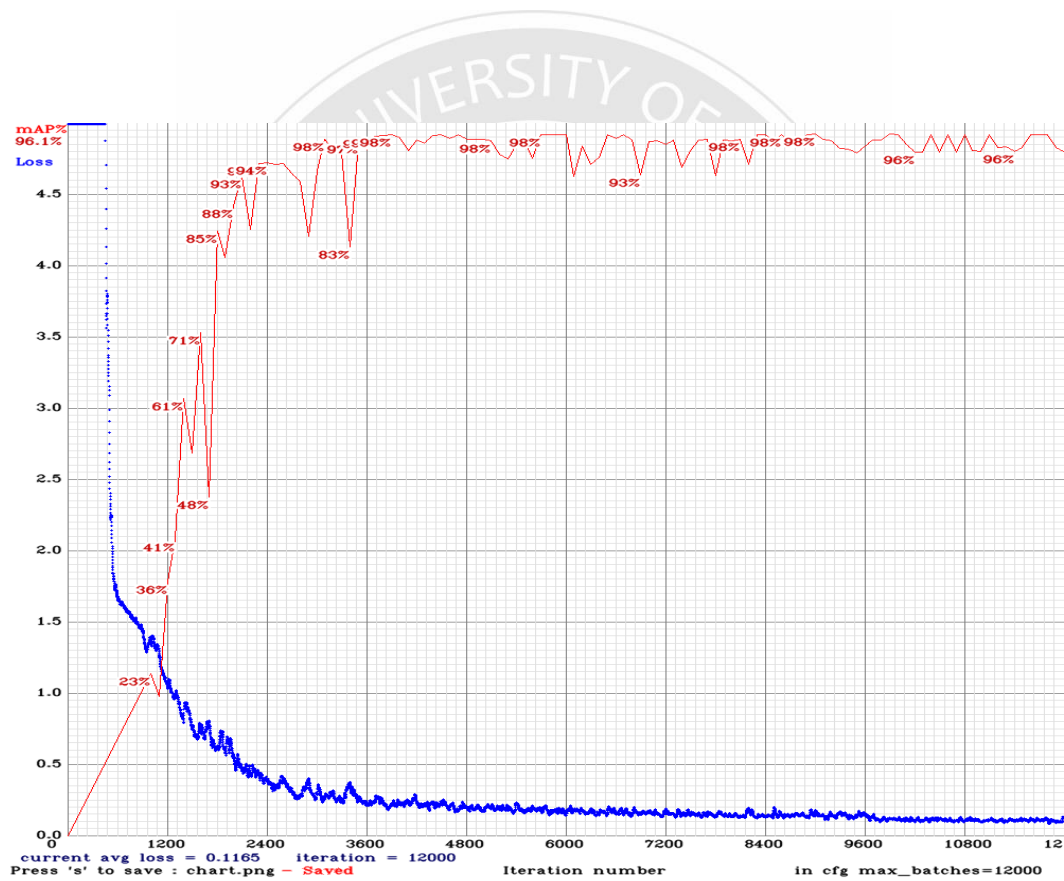


圖 31 YOLOv3-Tiny 12000 次培訓步驟-8

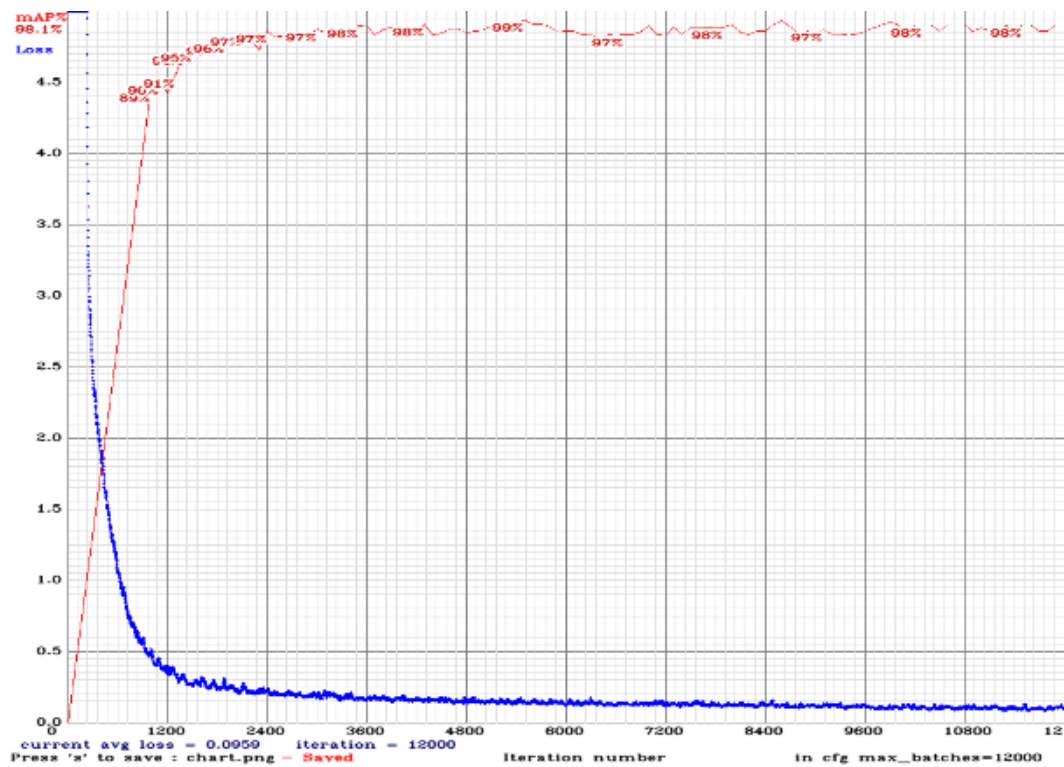


圖 32 YOLOv3-Tiny 12000 次培訓步驟-9

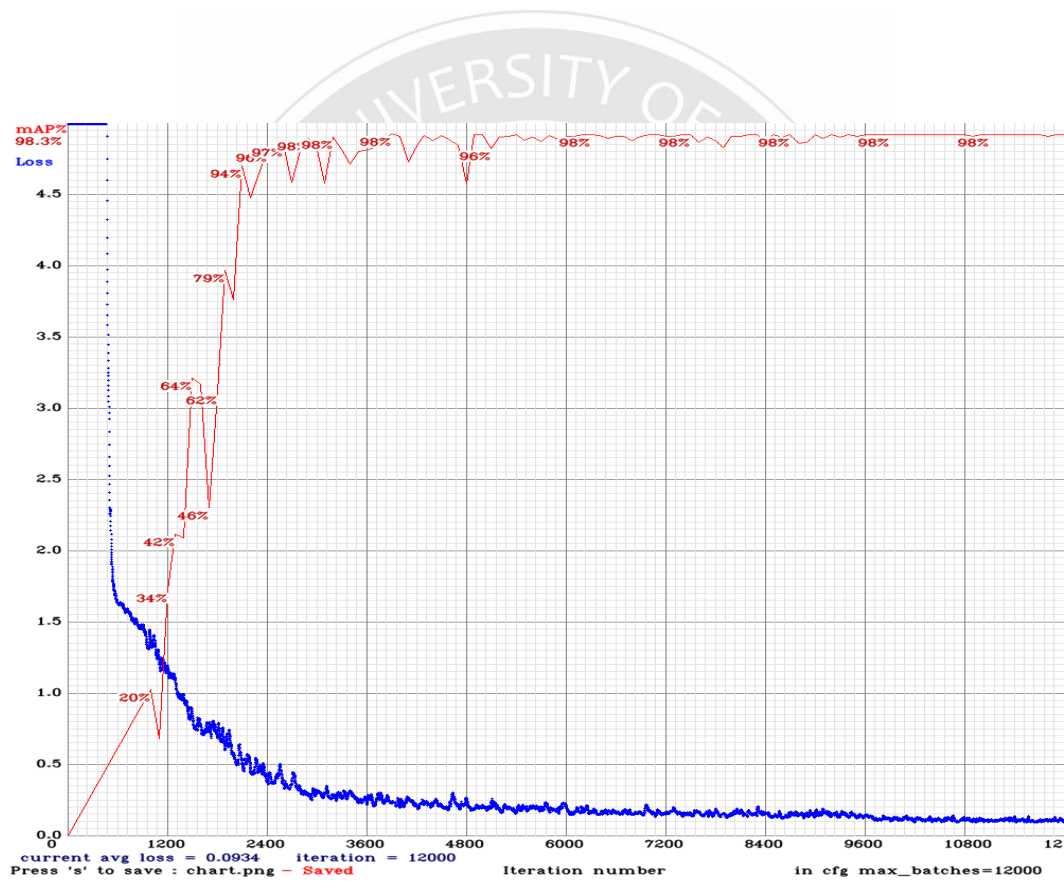


圖 33 YOLOv3-Tiny 12000 次培訓步驟-10

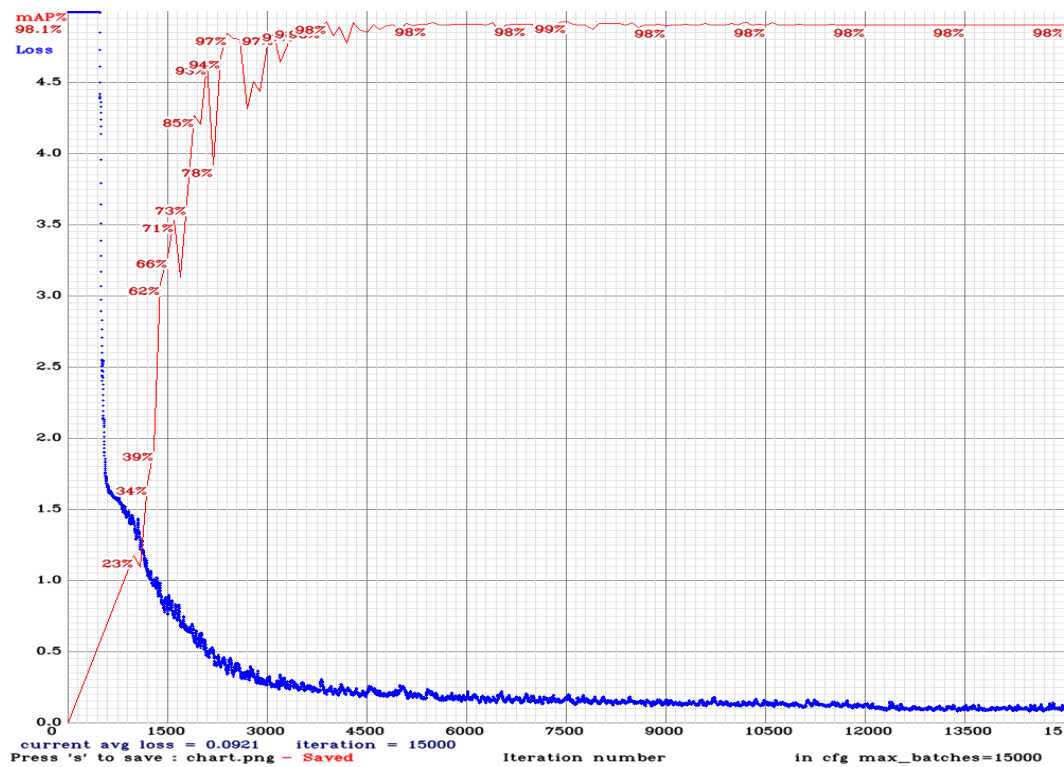


圖 34 YOLOv3-Tiny 15000 次培訓步驟-1

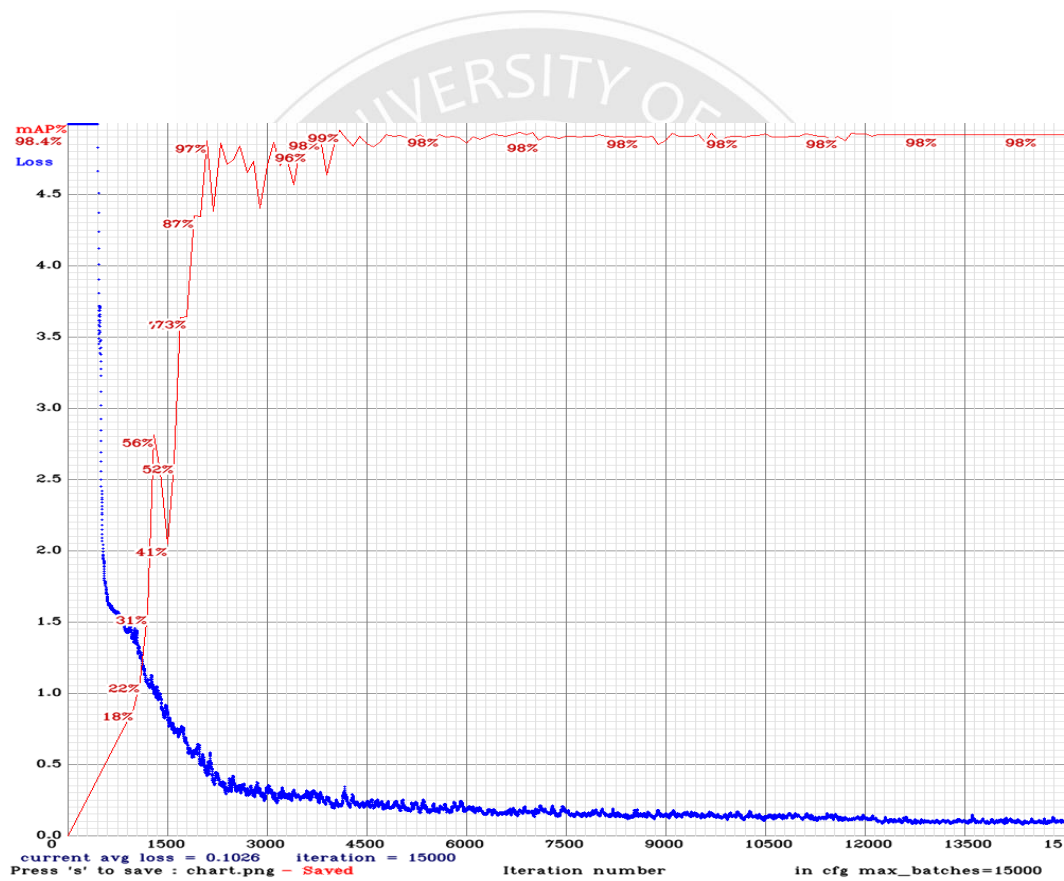


圖 35 YOLOv3-Tiny 15000 次培訓步驟-2

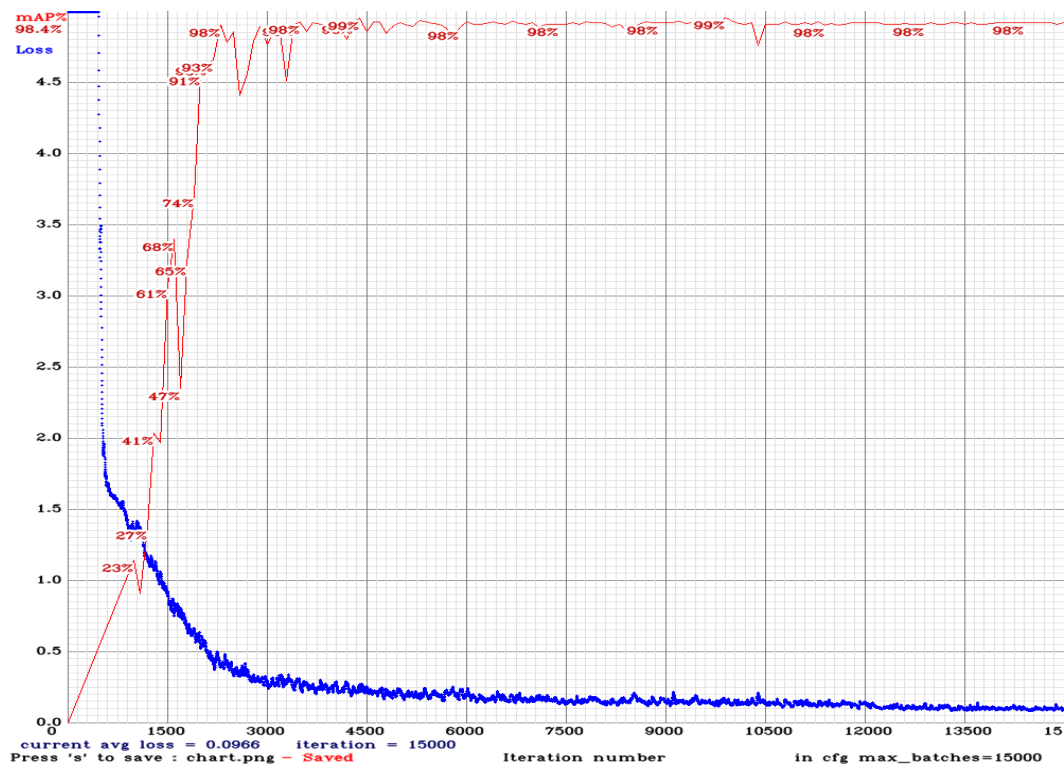
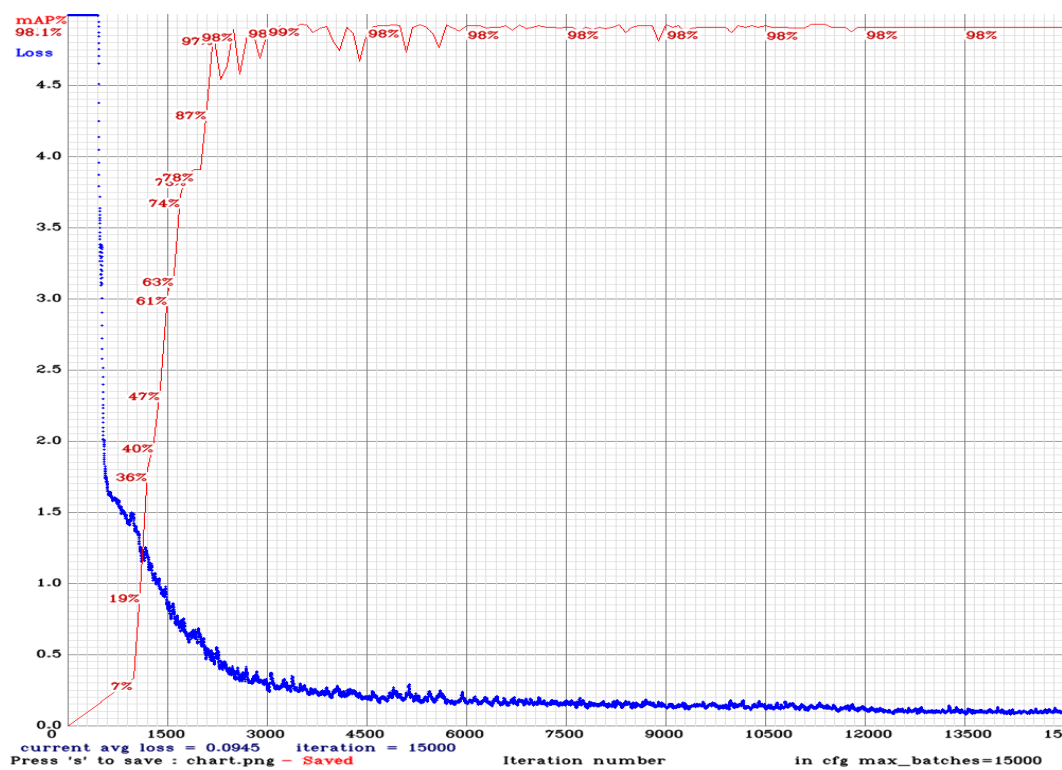
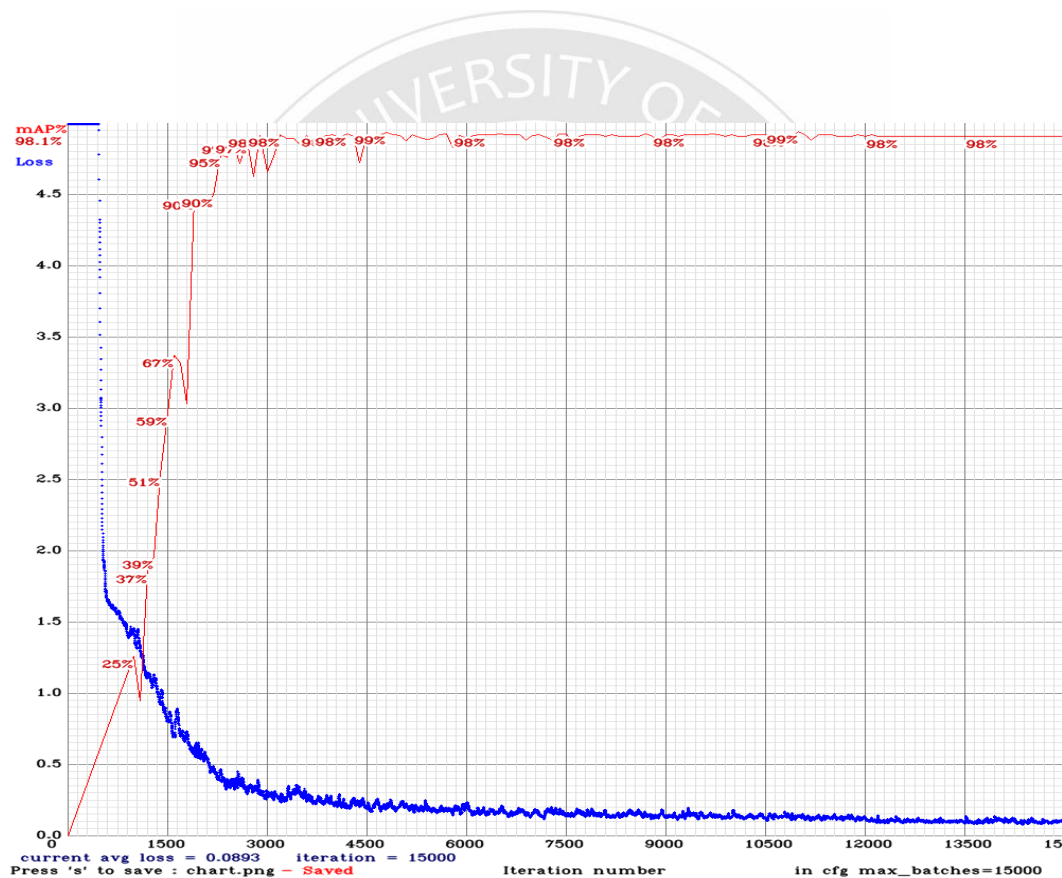


圖 36 YOLOv3-Tiny 15000 次培訓步驟-3



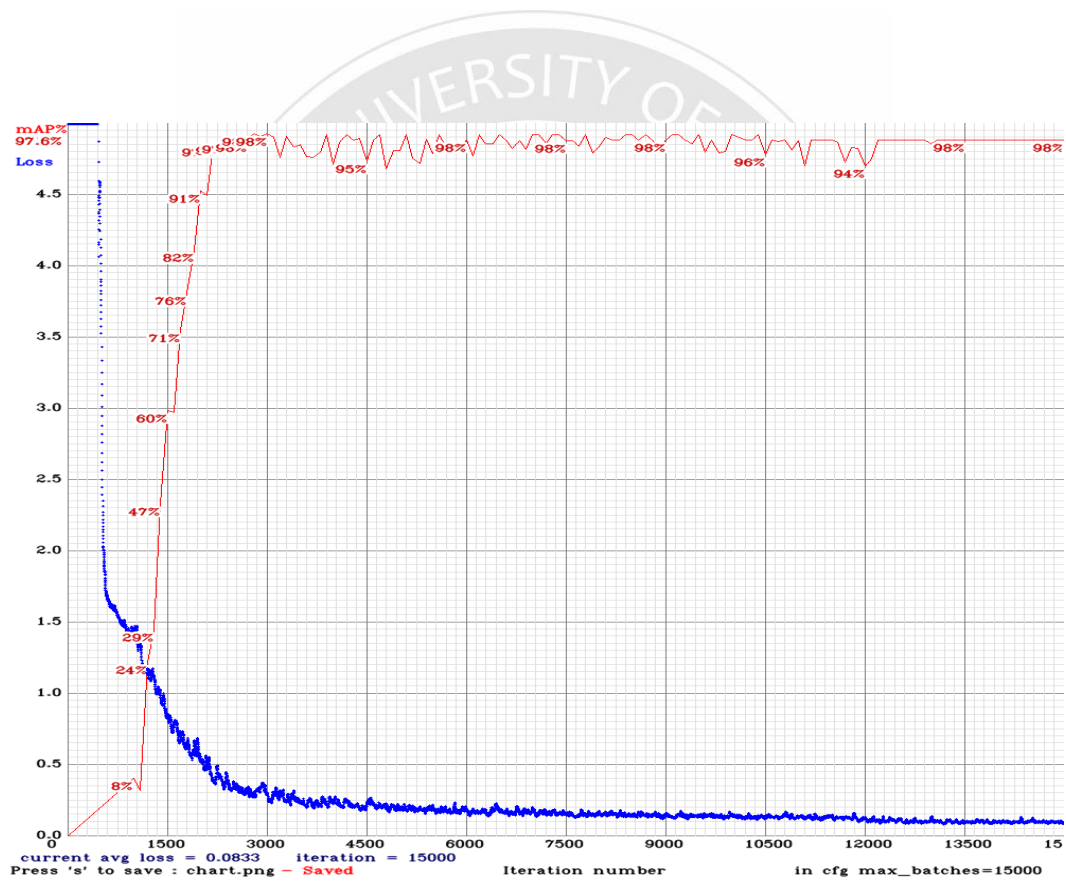


圖 39 YOLOv3-Tiny 15000 次培訓步驟-6

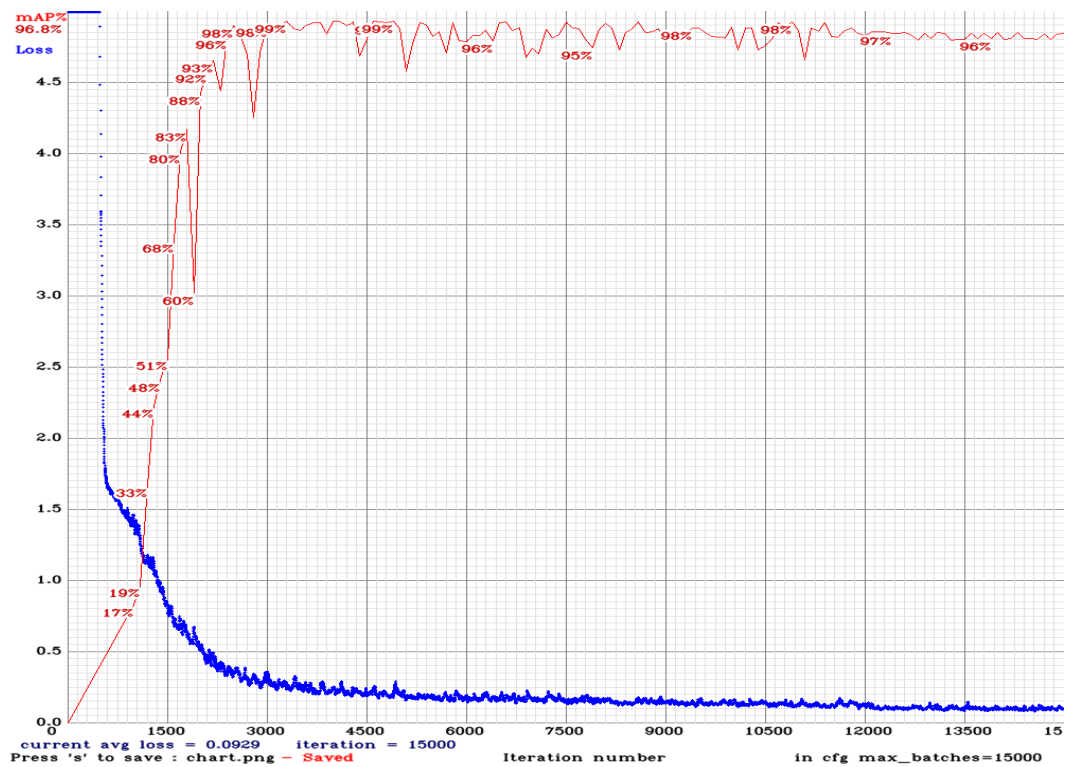


圖 40 YOLOv3-Tiny 15000 次培訓步驟-7

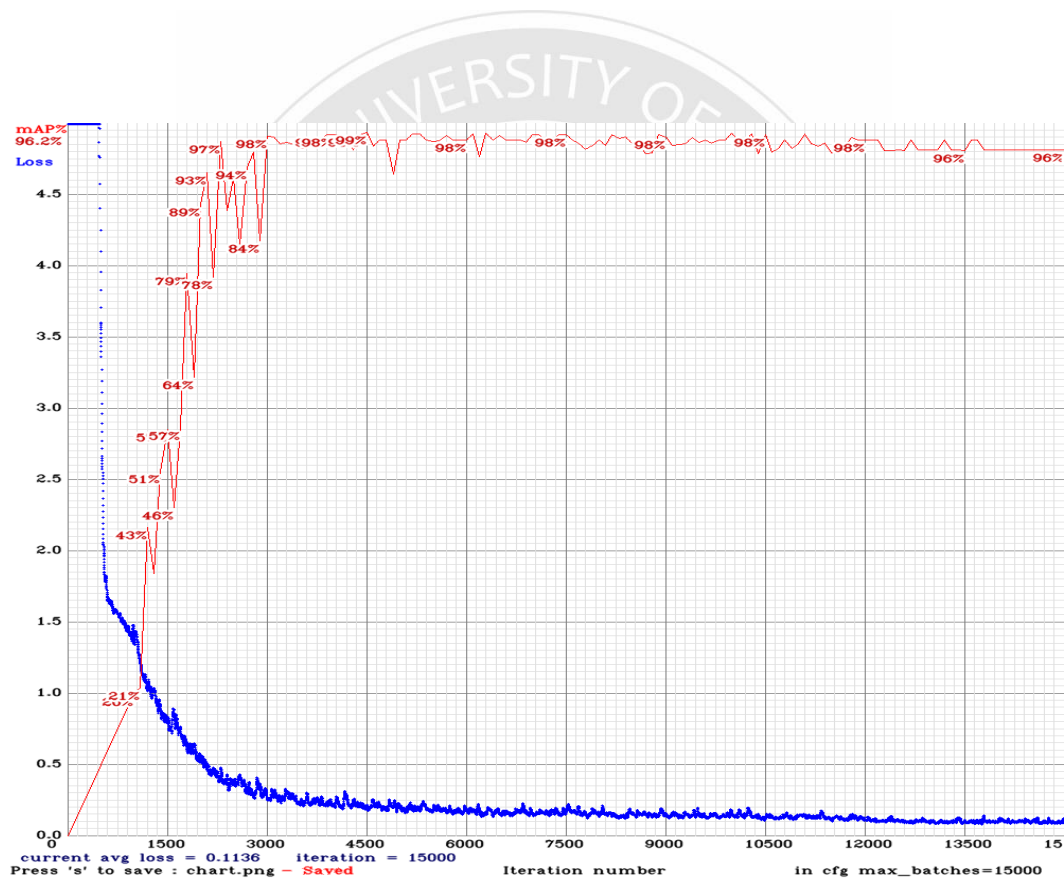


圖 41 YOLOv3-Tiny 15000 次培訓步驟-8

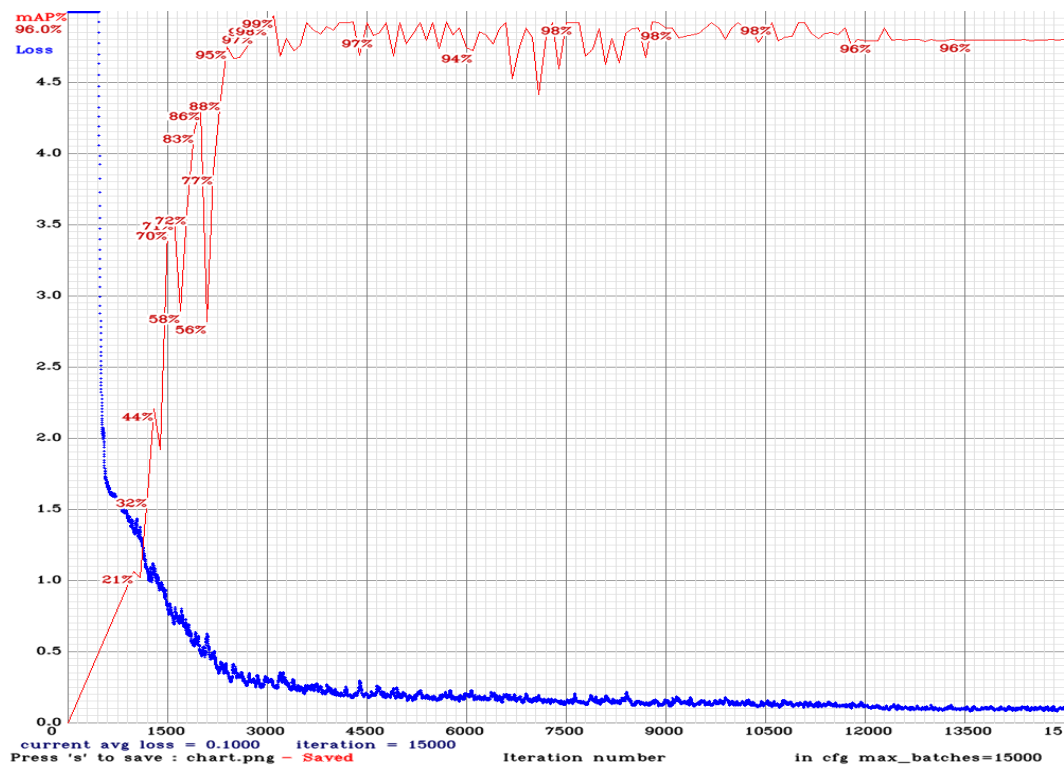


圖 42 YOLOv3-Tiny 15000 次培訓步驟-9

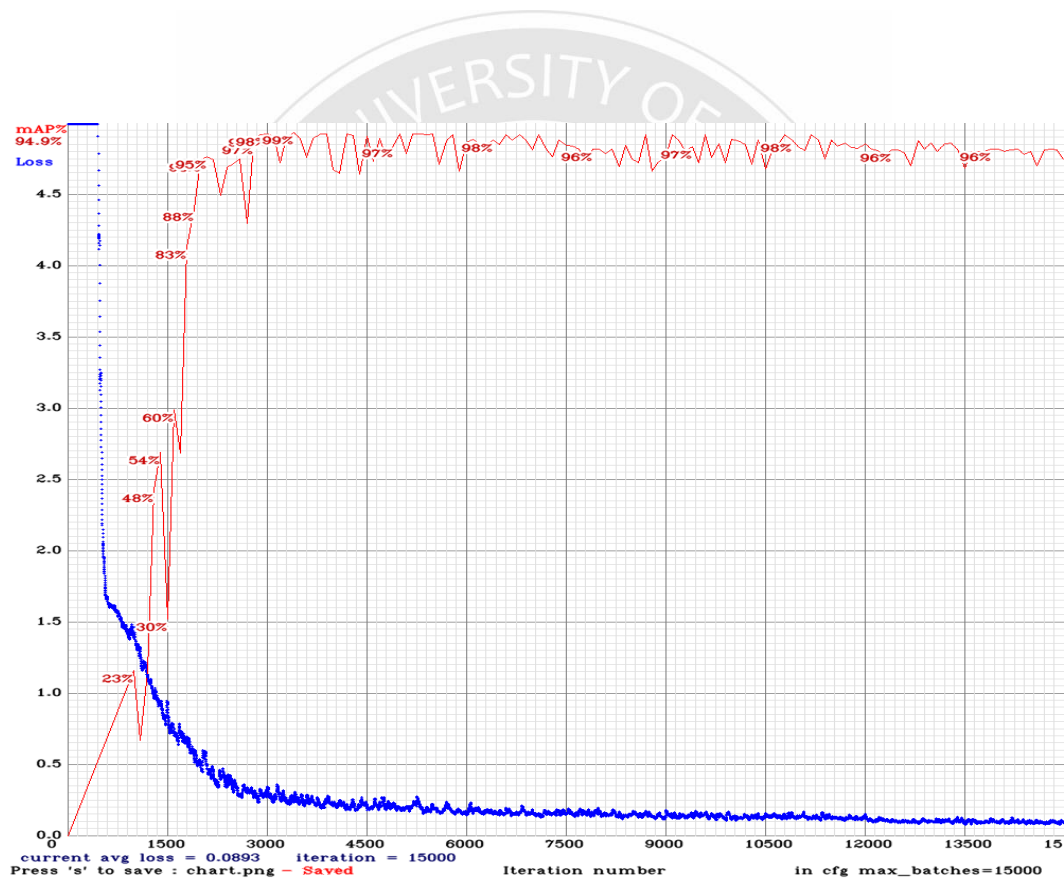


圖 43 YOLOv3-Tiny 15000 次培訓步驟-10

表 2 YOLOv3-Tiny batches9000 各項指標結果

No.	TP	FP	FN	Precision	Recall	F1-score	IoU	mAP	Avg loss
1	349	2	2	0.99	0.99	0.99	81.52	97.0	0.1198
2	349	2	2	0.99	0.99	0.99	80.78	96.2	0.1203
3	349	3	2	0.99	0.99	0.99	80.69	96.3	0.1572
4	349	2	2	0.99	0.99	0.99	81.42	97.0	0.1103
5	349	3	2	0.99	0.99	0.99	81.40	96.9	0.1211
6	349	3	2	0.99	0.99	0.99	80.98	96.1	0.1208
7	349	2	2	0.99	0.99	0.99	81.23	96.0	0.1178
8	349	2	2	0.99	0.99	0.99	81.78	98.3	0.1380
9	349	3	2	0.99	0.99	0.99	80.89	96.1	0.1170
10	349	2	2	0.99	0.99	0.99	81.67	97.2	0.1121

表 3 YOLOv3-Tiny batches12000 各項指標結果

No.	TP	FP	FN	Precision	Recall	F1-score	IoU	mAP	Avg loss
1	349	2	2	0.99	0.99	0.99	81.81	98.3	0.0965
2	349	1	2	1.00	0.99	1.00	81.73	98.3	0.0908
3	349	2	2	0.99	0.99	0.99	81.36	98.1	0.1126
4	349	2	2	0.99	0.99	0.99	81.47	98.3	0.1118
5	349	1	2	1.00	0.99	1.00	81.88	98.3	0.0934
6	349	2	2	0.99	0.99	0.99	81.73	97.6	0.0948
7	349	2	2	0.99	0.99	0.99	81.56	97	0.1237
8	349	2	2	0.99	0.99	0.99	81.13	96.1	0.1165
9	349	1	2	1.00	0.99	1.00	81.65	98.1	0.0959
10	349	2	2	0.99	0.99	0.99	81.84	98.3	0.0934

表 4 YOLOv3-Tiny batches15000 各項指標結果

No.	TP	FP	FN	Precision	Recall	F1-score	IoU	mAP	Avg loss
1	349	2	2	0.99	0.99	0.99	82.17	98.1	0.0921
2	349	3	2	0.99	0.99	0.99	81.76	98.4	0.1026
3	349	1	2	1.00	0.99	1.00	82.19	98.4	0.0966
4	349	2	2	0.99	0.99	0.99	82.28	98.1	0.0893
5	349	1	2	1.00	0.99	1.00	82.19	98.1	0.0945
6	349	2	2	0.99	0.99	0.99	81.56	97.6	0.0833
7	349	2	2	0.99	0.99	0.99	81.42	96.8	0.0929
8	349	3	2	0.99	0.99	0.99	82.53	96.2	0.1136
9	349	2	2	0.99	0.99	0.99	81.05	96	0.1000
10	349	3	2	0.99	0.99	0.99	81.43	94.9	0.0893

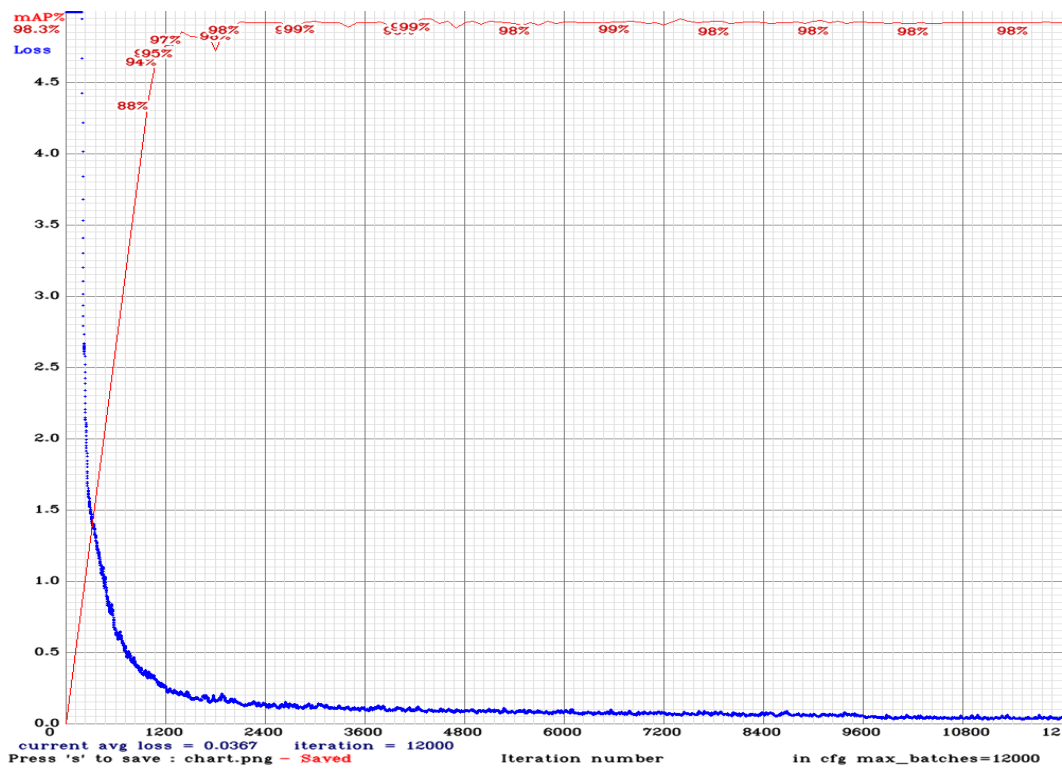
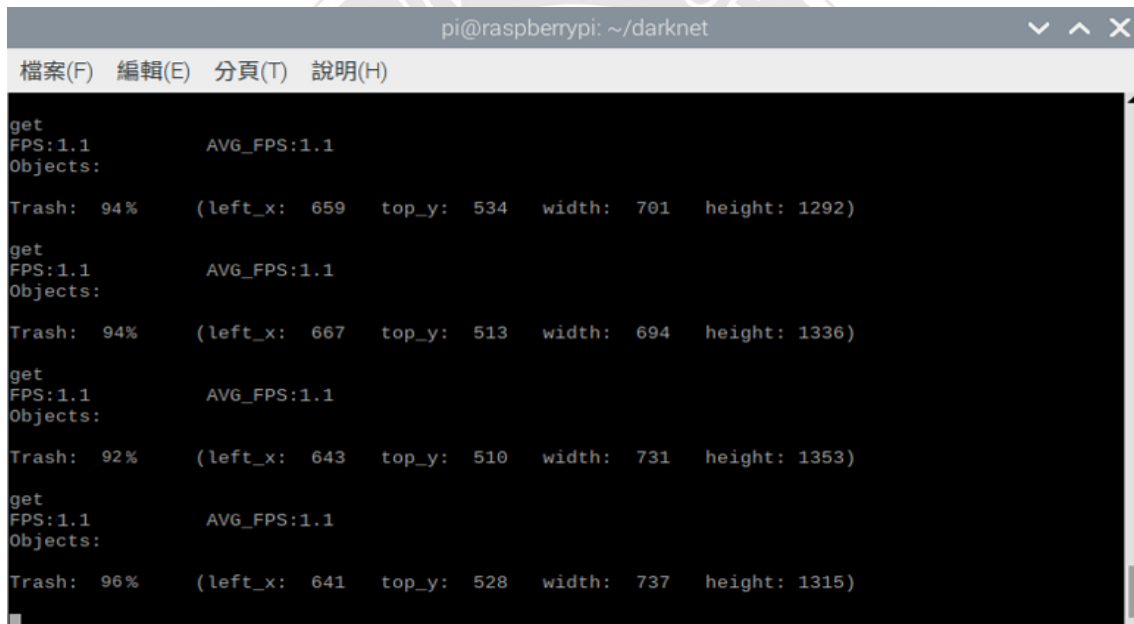


圖 44 YOLOv3 模型的平均損失和 mAP



```
pi@raspberrypi: ~/darknet
檔案(F) 編輯(E) 分頁(T) 說明(H)

get
FPS:1.1      AVG_FPS:1.1
Objects:

Trash: 94%   (left_x: 659  top_y: 534  width: 701  height: 1292)

get
FPS:1.1      AVG_FPS:1.1
Objects:

Trash: 94%   (left_x: 667  top_y: 513  width: 694  height: 1336)

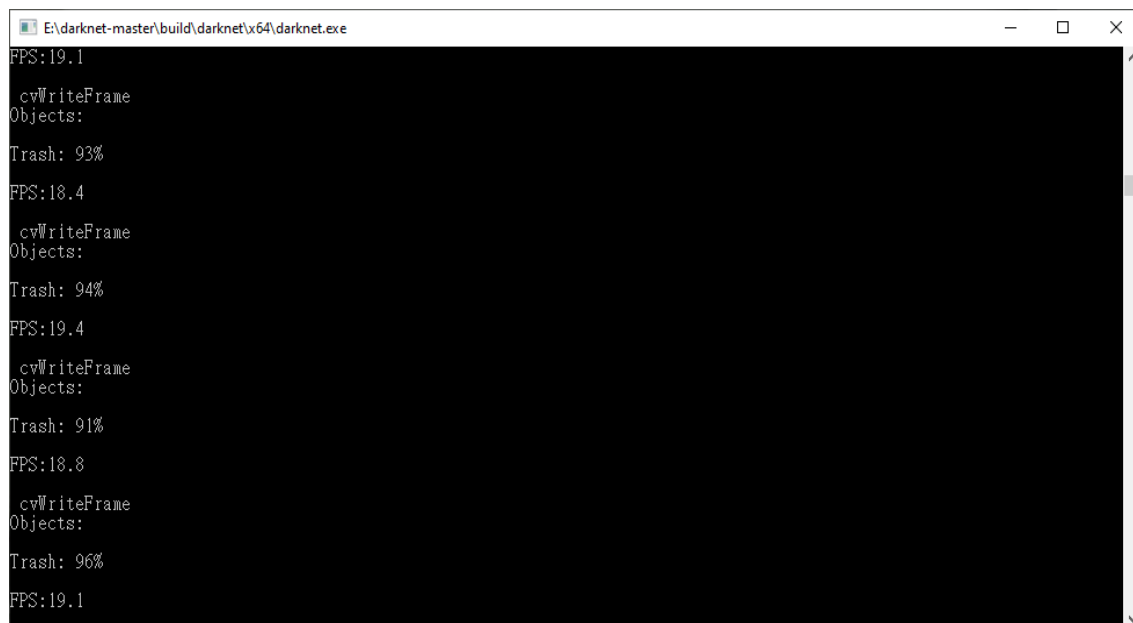
get
FPS:1.1      AVG_FPS:1.1
Objects:

Trash: 92%   (left_x: 643  top_y: 510  width: 731  height: 1353)

get
FPS:1.1      AVG_FPS:1.1
Objects:

Trash: 96%   (left_x: 641  top_y: 528  width: 737  height: 1315)
```

圖 45 YOLOv3-Tiny 模型檢測速度



```
E:\darknet-master\build\darknet\x64\darknet.exe
FPS:19.1
cvWriteFrame
Objects:

Trash: 93%

FPS:18.4
cvWriteFrame
Objects:

Trash: 94%

FPS:19.4
cvWriteFrame
Objects:

Trash: 91%

FPS:18.8
cvWriteFrame
Objects:

Trash: 96%

FPS:19.1
```

圖 46 YOLOv3 模型檢測速度

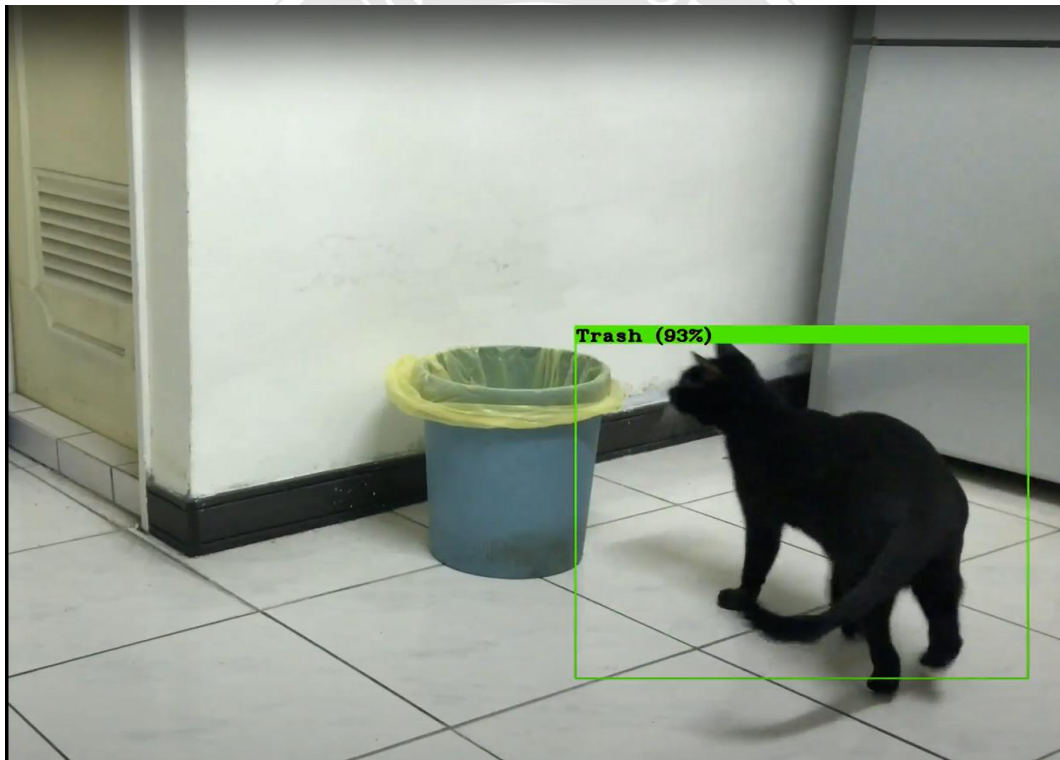


圖 47 行走類別被偵測為翻垃圾桶類別

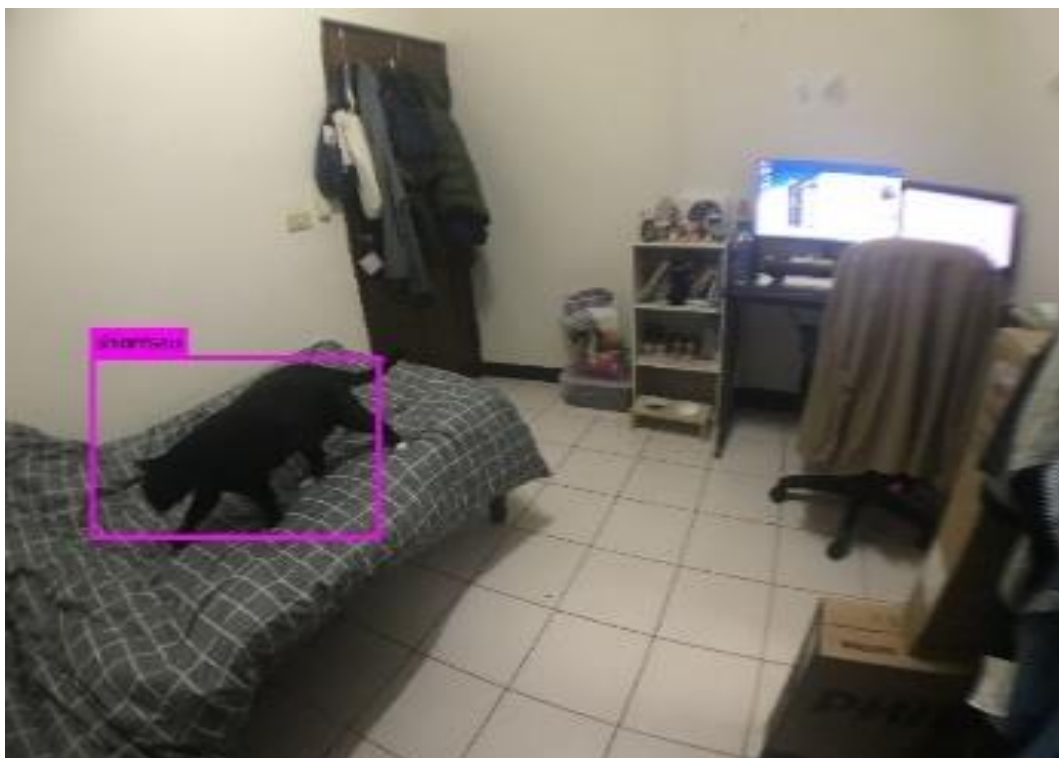


圖 48 行走檢測結果



圖 49 進食檢測結果

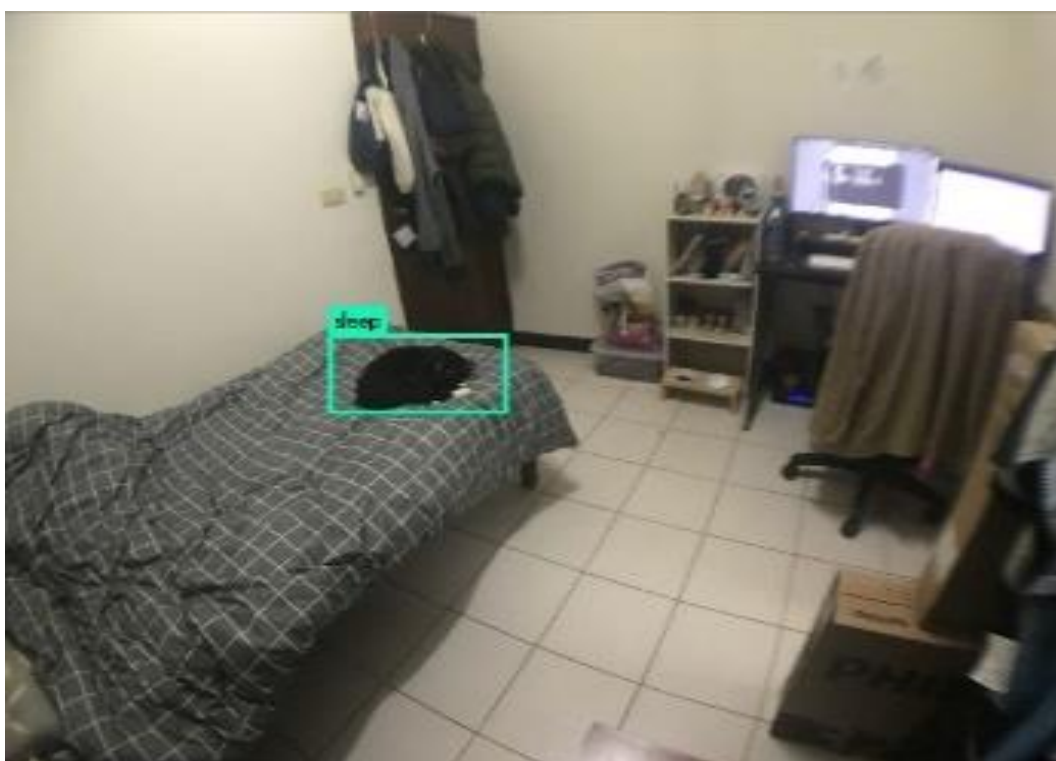


圖 50 睡覺檢測結果

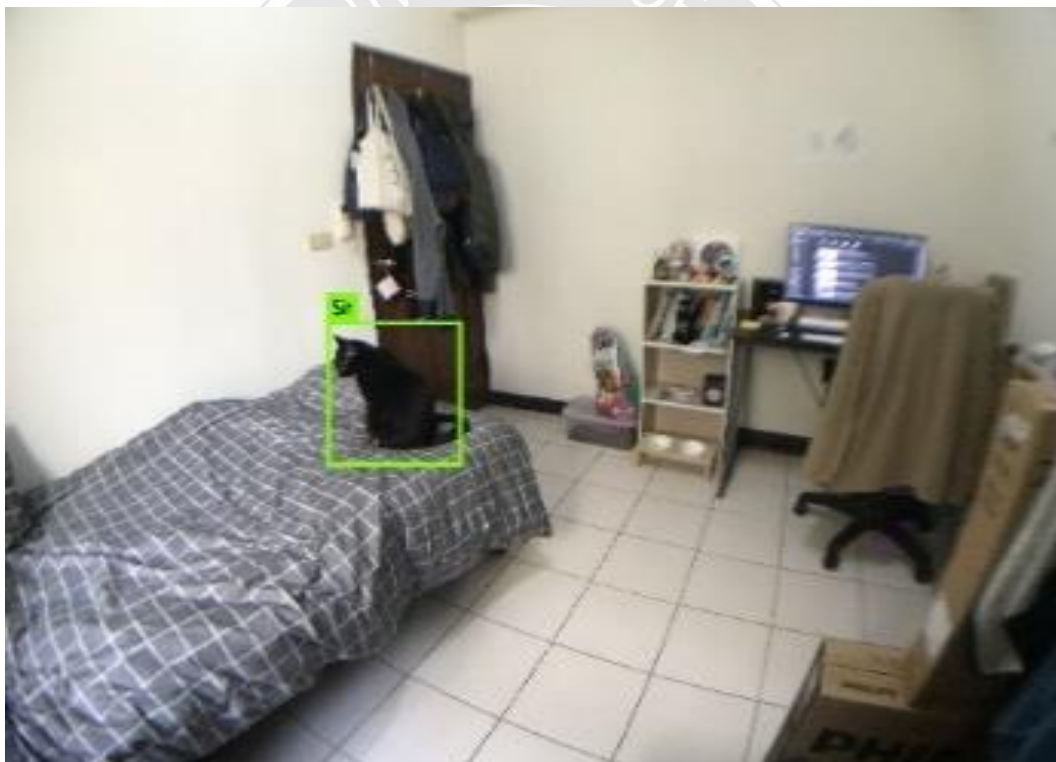


圖 51 坐下檢測結果

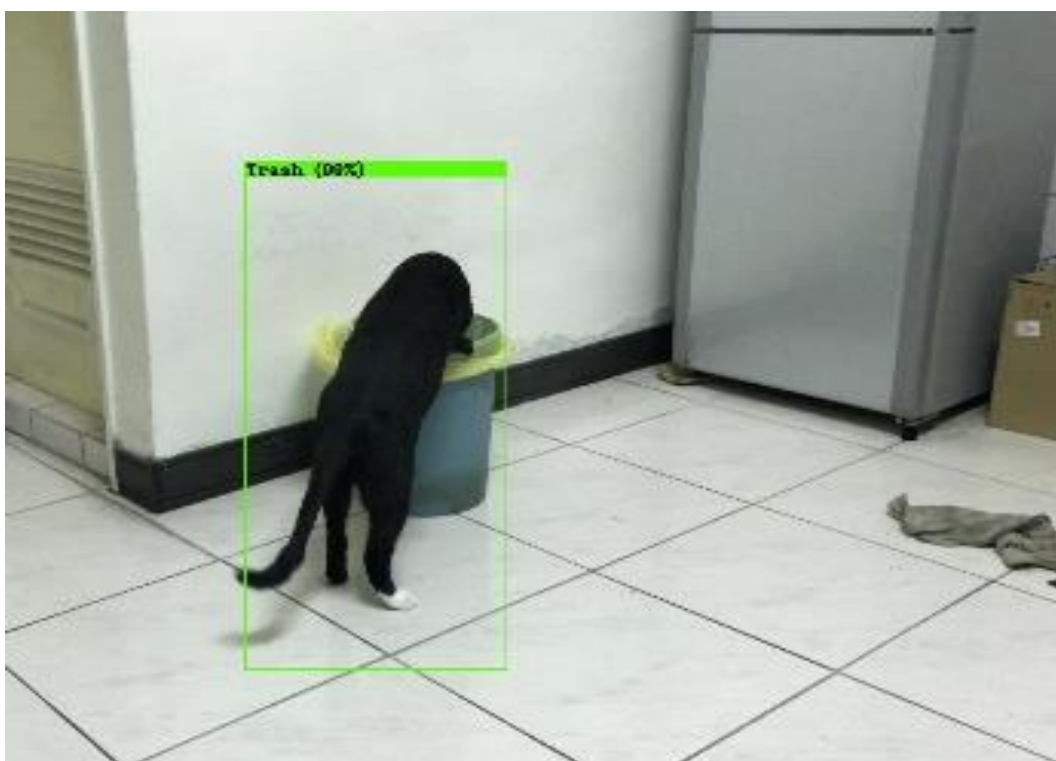


圖 52 翻垃圾桶檢測結果



圖 53 上廁所檢測結果

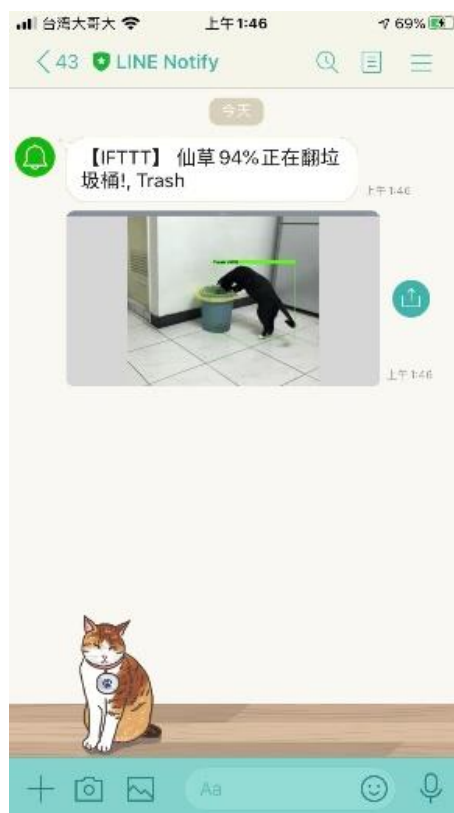


圖 54 翻垃圾桶回傳至手機結果



圖 55 上廁所超過 30 秒回傳至手機結果

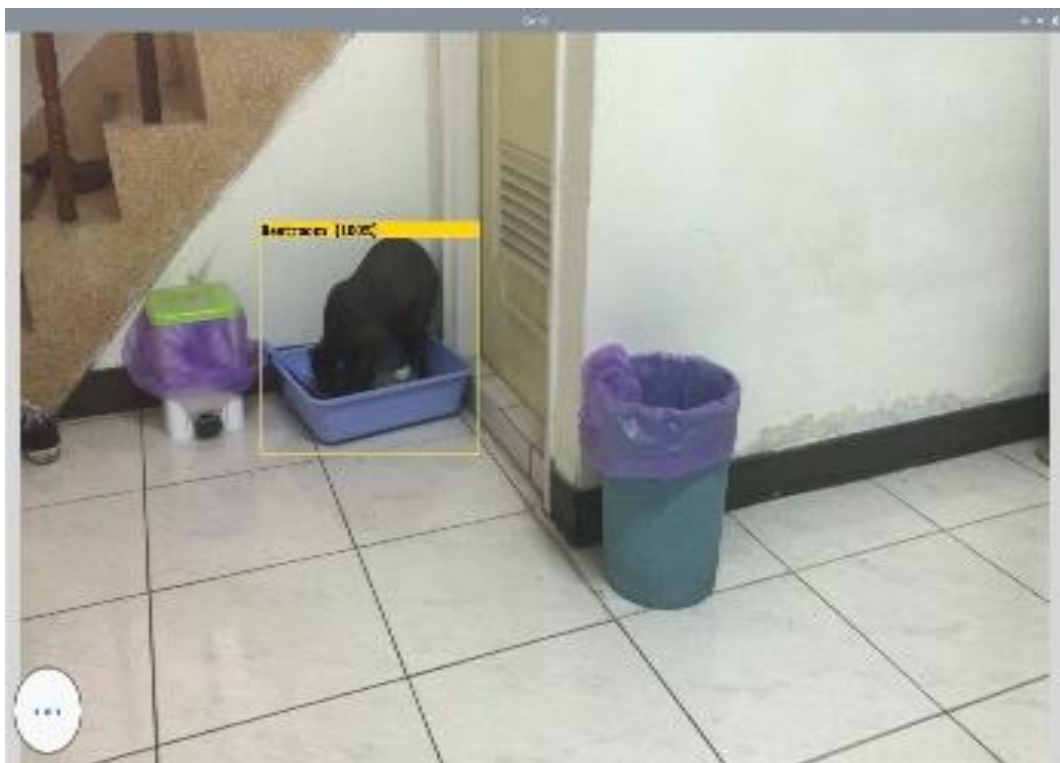


圖 56 透過手機使用 VNC 即時察看結果

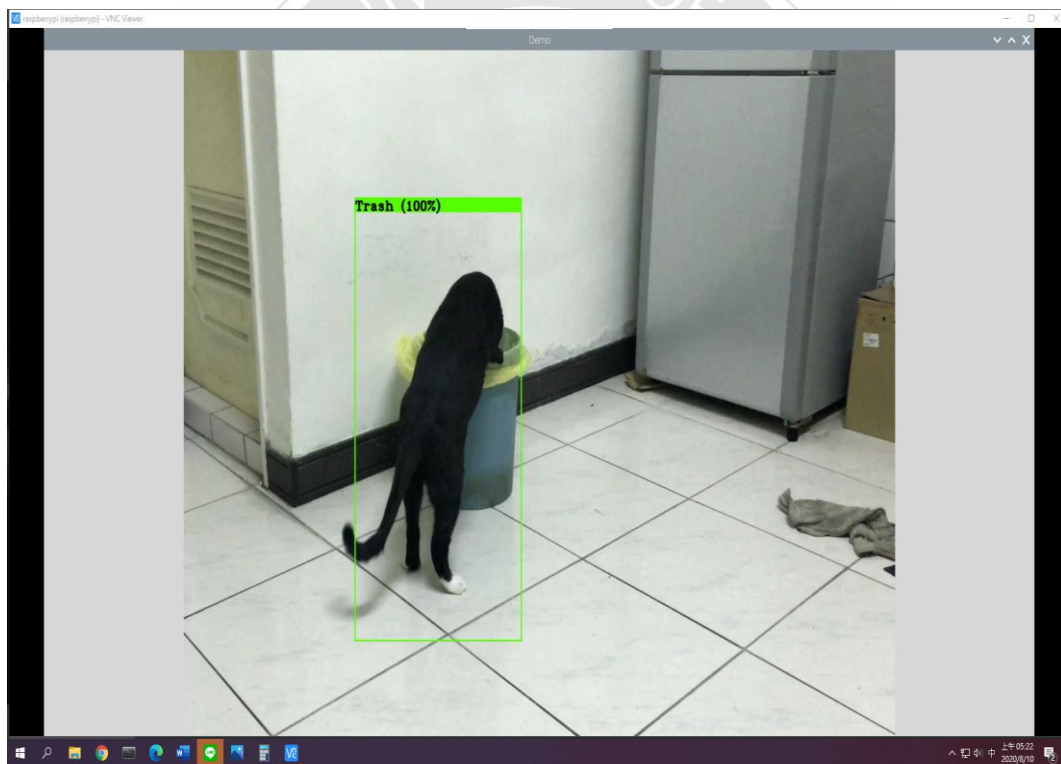


圖 57 透過電腦使用 VNC 即時察看結果

第五章 結論與未來展望

5.1 結論

現今，雖然市面上已經有許多寵物攝影機可以監控寵物們的一舉一動，但是飼者並不能時時刻刻地看著視頻監控著寵物，當飼主離開屏幕前，寵物很有可能在這期間亂翻垃圾桶或是亂咬東西而造成生命危險。因此本研究藉由自己飼養的貓咪所遇到的問題，設計一個基於 YOLO 模型的寵物貓咪行為偵測系統，當貓咪有行走、坐下、睡覺、吃飯、上廁所或亂翻垃圾桶等行為時，系統會偵測出正確的行為類別，而其中當寵物上廁所超過 30 秒或是亂翻垃圾桶時，系統透過 IFTTT 將偵測到的行為和照片，即時傳送訊息至手機，系統也使用 VNC 遠端監控，讓飼主也可以透過手機或電腦即時遠端監控貓咪當下的行為。在實驗中，我們比較了 YOLOv3-Tiny 和 YOLOv3，雖然 YOLOv3-Tiny 在檢測速度上並不算太優秀，但是在準確率和 YOLOv3

相比還是相當高。

5.2 未來展望

在未來，飼養寵物的家庭會只會越來越多，為了能夠將系統能夠套用在每一隻寵物上，我們希望能收集更多寵物的種類和行為的類別，例如：舔毛、玩耍、抓紙板、尾巴甩動等等，藉此豐富此數據集，以確保實驗的可用性和實用性，而在檢測速度上，由於樹莓派硬體的限制，很難將效能在更進一步的提高，所以在未來若有更適合的硬體能套用此系統，就能夠有更良好的檢測性能。

參考文獻

- [1] LifeSmart 智慧家庭系統，<http://www.intop.com.tw/>，2020/6/15。
- [2] Pawbo 波寶寵物智能家族，<https://www.pawbopet.com/tw/>，2020/6/15。
- [3] 解析貓咪異食癖，<https://www.pettalk.tw/blog/>，2020/6/16。
- [4] 貓咪的泌尿，<https://www.hills.com.tw/health-conditions/cat/cat-urinary>，2020/6/16。
- [5] World Population Review，<https://worldpopulationreview.com/>，2020/6/18
- [6] 2020 黃金交叉，台灣毛小孩首次超過兒童數，<http://www.trendsightinc.com/insight/>，2020/6/18。
- [7] 財政部統計通報 - 財政部，<https://www.mof.gov.tw/singlehtml/384fb3077bb349ea973e7fc6f13b6974?cntId=statdot86090>，2020/6/18。
- [8] 陪心糧寵，<https://www.nu4pet.com/>，2020/6/20。
- [9] 百威犬舍 - 百威愛犬訓練中心，<http://www.dogtrain.com.tw/plain/charges.htm>，2020/6/20。
- [10] 我們好好-寵物美容，<https://www.we.net.tw/>，2020/6/21。

- [11]安家|行動寵醫，<https://drfurkids.com/>，2020/6/21。
- [12]臺北市動物保護處，<https://www.tcapo.gov.taipei/>，2020/6/22。
- [13]劉家名(2018)，利用樹莓派實作基於影像及手勢控制板之手勢辨識，碩士論文，朝陽科技大學資訊管理系。
- [14]R. A Nadafa, S.M. Hatturea, Vasudha M Bonalam Susen P Naikb(2020), “Home Security against Human Intrusion using Raspberry Pi,” *Procedia Computer Science*, Vol 167, pp. 1811-1820.
- [15]M. F. Hansen, M. L. Smith, L. N. Smith, M. G. Salter, E. M. Baxter, M. Farish and B. Grieve(2018), “Towards on-farm pig face recognition using convolutional neural networks” *Computers in Industry*, Vol 98, pp. 145-152.
- [16]H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, E. G. Ritchie and D. Phung(2017), “Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring,” *2017 IEEE International Conference on Data Science and Advanced Analytics(DSAA)*, pp. 40-49.
- [17]X. Shu-shu, L. Lan and Z. Cai-bao(2019), ”Detection and classification of pet dogs based on SSD,” *Information Technology and Informatization*.
- [18]R. O. Sinnott, F. Wu and W. Chen(2018), “A Mobile Application for Dog Breed Detection and Recognition Based on Deep Learning,” *2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT)*, pp. 87-96.
- [19]P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn and W. Kusakunniran(2019), “Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning,” *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 1-5.
- [20]J. Redmon, S. Divvala, R. Girshick and A. Farhadi(2016), “You Only Look Once: Unified, Real-Time Object Detection,” *2016 IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR), pp 779-788.
- [21]J. Redmon and A. Farhadi(2017), “YOLO9000: Better, Faster, Stronger,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525.
- [22]J. Redmon and A. Farhadi(2018), “YOLOv3: An Incremental Improvement,” 2018 Computer Vision and Pattern Recognition, arXiv.1804.
- [23]K. He, X. Zhang, S. Ren and J. Sun(2015), “Deep Residual Learning for Image Recognition,” Computer Vision and Pattern Recognition, arXiv.1512.
- [24]T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie(2016), “Feature Pyramid Networks for Object Detection,” Computer Vision and Pattern Recognition, arXiv.1612.
- [25]C. Liang, J. Xiong, Z. Zheng, Z. Zhong, Z. Li, S. Chen and Z. Yang(2020), “A visual detection method for nighttime litchi fruits and fruiting stems,” Computers and Electronics in Agriculture, Vol 169.
- [26]F. Wu, G. Jin, M. Gao, Z. HE and Y. Yang, “Helmet Detection Based On Improved YOLO V3 Deep Model,” 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), pp. 363-368.
- [27]G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger(2017), "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261-2269.
- [28]Y. Hongwen, L. Zhenyu, C. Qingliang, H. Zhiwei and L. Y. wen(2019), “Detection of facial gestures of group pigs based on improved Tiny-YOLO,” Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), pp.169-179.
- [29]M. Botticelli, L. Ciabattini, F. Ferracuti, A. Monteriù, S. Pizzuti and S. Romano(2018), “A Smart Home Services Demonstration: Monitoring, Control and Security Services Offered to the User,” 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin),

pp. 1-4.

- [30]M. U. H. A. Rasyid, F. A. Saputra and A. Prasetyo(2018), “I-ON Smart Controller: Portable Smart Home Solution Based on Arduino And Raspberry Pi,” 2018 International Conference on Applied Science and Technology (iCAST), pp. 161-164.
- [31]HomeKit, <https://www.apple.com/tw/shop/accessories/all-accessories/homekit>, 2020/7/5.
- [32]IFTTT helps every thing work better together, <https://ifttt.com/home>, 2020/7/5.
- [33]Download VNC Viewer | VNC® Connect – RealVNC, <https://www.realvnc.com/en/connect/download/viewer/>, 2020/7/6.
- [34]Bbox label tool, <https://github.com/puzzledqs/BBox-Label-Tool>, 2020/7/15.