

深度學習

基於 YOLOv3 進行影像增強前後的 暗影像物件檢測評估

作者:

312510191 杜家漢

目錄

- 緒論
- 研究動機
- 研究方法和改進方式
- 研究結果
- 結論
- 參考資料

緒論:

這次深度學習期末專題，起初的期中發表我選擇的主題是 ViT，以及使用蒸餾法改善後的 DeiT 架構，但後來發現這些模型使用到的參數以及架構過於龐大，再加上教授上課也有提到這邊論文，再三考量後，我決定在期末時更換論文以及領域。首要目標是以效率為主，再來是論文本身的新奇程度去做選擇。於是我挑選了這篇“Getting to know low-light images with the Exclusively Dark dataset”，發表於 CVIU 的期刊論文。

研究動機:

選擇這篇論文的動機主要可以分為以下幾點

- 物件檢測在**自動駕駛汽車、自動車輛檢測和圖像分析**一直都是這幾年來非常熱門的議題，隨著時代的變遷，期望能在這方面再注入一些新的技術加以改革。
- 以往深度學習架構再針對解析度高或品質好的圖像效果都還不錯，但使用深度學習框架進行物件檢測在**弱光影像中的物件**時會面臨一些挑戰
- 弱光是我們生活環境中不可避免的因素，一天中根據不同的因素會有不同條件下的光源，**檢測這些光源照射下的物件成為一項挑戰**

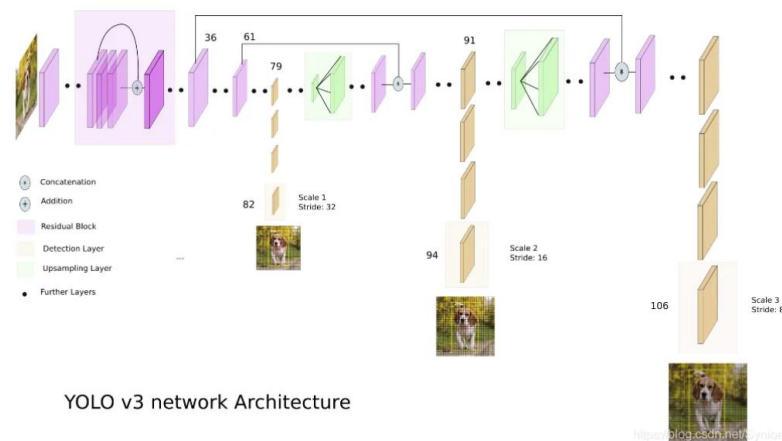
研究流程:



研究方法:

這次的專題我主要使用的 dataset 為 Exclusively Dark 來當作我的訓練和驗證資料集，此資料集應該算是網路上唯一有整理好的弱光影像資料集，各大相關論文都是以這個資料集來訓練，使用的模型架構為 YOLOv3(You Only Look Once)

- YOLOv3(You Only Look Once version 3)是一種用於物件檢測的深度學習模型，通過 Darknet-53 作為特徵提取器，可以捕捉不同層次的圖像細節與訊息，實現更有效率與精確度的演算(Redmon & Farhadi, 2018)



Yolov3 整理架構，可以發現他從backbone中取了三種尺度的feature，丟到dection model做物件偵測

圖 1 YOLO 架構

此次資料及總共有 12 個類別分別是 'Bicycle', 'Boat', 'Cat', 'Dog', 'Car', 'Bottle', 'Bus', 'Motorcycle', 'Cup', 'People', 'Chair', 'Table'. 然後此次資料集也包含了光強度的分類總共有 10 個類別，但此次沒有再針對光強度下去做分類。

Number of images per object class used for analyses.	
Dataset Class	Exclusively Dark Number of Image
Bicycle	652
Boat	679
Bottle	547
Bus	527
Car	638
Cat	735
Chair	648
Cup	519
Dog	801
Motorbike	503
People	609
Table	505
Total	7363

圖 2 資料集分布

透過這個架構訓練出來的結果再計算 **ap** 以及 **map** 來判斷模型對於物體偵測效果的好與壞。並將每個 **epoch** 結束後得到的 **weight** 儲存下來，最後在保留 **mAP** 最佳的那個 **pth** 檔案。

改進方式:

起初我想調整模型內部的架構，並做 **fine-tuning**，觀察是否可以得到更好的結果，但考量到模型本身已經是完整的架構，再做微調可能會造成效果更差，所以我選擇使用影像增強的方式，在同樣的模型架構中是否可以得到更好的結果。我使用的影像增強方式是把原本弱光的影像丟入 **Illumination-Adaptive-Transformer(IAT)**，這個模型架構中，生成出更高品質的影像。

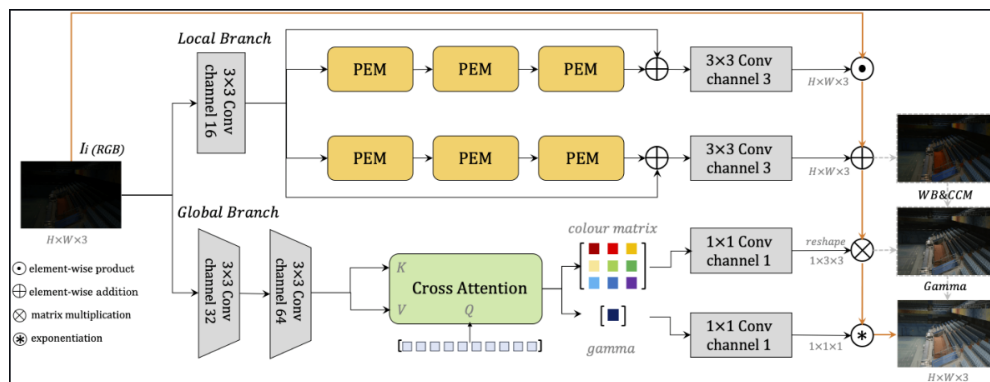


圖 3 IAT 架構

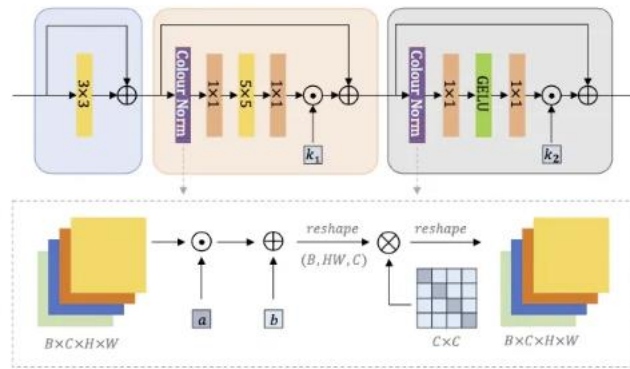


圖 4 PEM 架構

論文我是參考 “You Only Need 90K Parameters to Adapt Light: a Lightweight Transformer for Image Enhancement and Exposure Correction.” 2022 發表於英國的 BMVC。

IAT 模型由兩個獨立分支組成，其中本地分支(Local Branch)用於像素級調整，輸出兩個特徵圖以進行加法和乘法。全局分支(Global branch)用於全局調整，輸出色彩矩陣和 γ 值，全局分支受到 DETR 的啟發，該網絡通過動態查詢學習的方式更新色彩矩陣和 γ 值。簡單來說該模型先透過一個傳統的 CNN 架構大致還原出圖像的特徵，再透過另一個分支計算 Attention，得到色彩矩陣和 γ 值，對暗圖做 γ correction 和色彩的還原，而且效率也非常高。



圖 5 增強前



圖 6 增強後

此模型運算也有出現效果不佳的情況



圖 7 增強前



圖 8 增強後

撇除少數處理不好的情況，模型總體的表現是非常好的

研究結果：

- 未做影像增強的訓練結果

----- Training Model -----		
Training Epoch 27: 100% 348/348 [06:32<00:00, 1.13s/it]		
----- Saving checkpoint to: 'checkpoints/yolov3_ckpt_27.pth'		
----- Evaluating Model -----		
Validating: 100% 113/113 [01:37<00:00, 1.16it/s]		
Computing AP: 100% 12/12 [00:00<00:00, 719.07it/s]		
----- mAP 0.17375 -----		
----- Training Model -----		
Training Epoch 28: 100% 348/348 [06:35<00:00, 1.14s/it]		
----- Saving checkpoint to: 'checkpoints/yolov3_ckpt_28.pth'		

Index	Class	AP
0	bicycle	0.33563
1	boat	0.17091
2	bottle	0.21484
3	bus	0.40548
4	car	0.34458
5	cat	0.23040
6	chair	0.24309
7	cup	0.25995
8	dog	0.36645
9	motorbike	0.31478
10	people	0.35519
11	table	0.14299

----- mAP 0.28202 -----

圖 9 訓練中途的截圖

圖 10 訓練完畢後的驗證數據

- 影像增強的訓練結果

----- Training Model -----		
Training Epoch 27: 100% 348/348 [06:08<00:00, 1.06s/it]		
----- Saving checkpoint to: 'checkpoints/yolov3_ckpt_27.pth' -----		
----- Evaluating Model -----		
Validating: 100% 113/113 [01:33<00:00, 1.20it/s]		
Computing AP: 100% 12/12 [00:00<00:00, 475.99it/s]		
[0 1 2 3 4 5 6 7 8 9 10 11]		
----- mAP 0.20919 -----		
----- Training Model -----		
Training Epoch 28: 100% 348/348 [06:05<00:00, 1.05s/it]		
----- Saving checkpoint to: 'checkpoints/yolov3_ckpt_28.pth' -----		

Index	Class	AP
0	bicycle	0.47033
1	boat	0.20667
2	bottle	0.19733
3	bus	0.43480
4	car	0.38139
5	cat	0.30711
6	chair	0.18123
7	cup	0.24168
8	dog	0.42005
9	motorbike	0.38615
10	people	0.38704
11	table	0.16120

----- mAP 0.31458 -----

圖 11 訓練中途的截圖

圖 12 訓練完畢後的驗證數據

可以觀察到雖然只訓練了 30 個 EPOCH 但整體的效能上升非常明顯 Map 從 0.28 → 0.314，說明對於增強後的圖像，模型是可以比較好的去辨別圖像裡面的內容。值得注意的是，在訓練結果中有些 LABEL 的結果不升反降像是 Chair 和 Bottle，推測是因為訓練 epoch 太少或是資料集本身不夠大和多樣才會出現這種

狀況。

- 隨機影像輸出結果

我從驗證資料集中隨機挑選幾張照片，觀察有增強後和沒有增強的表現之間的差異。以下為結果圖

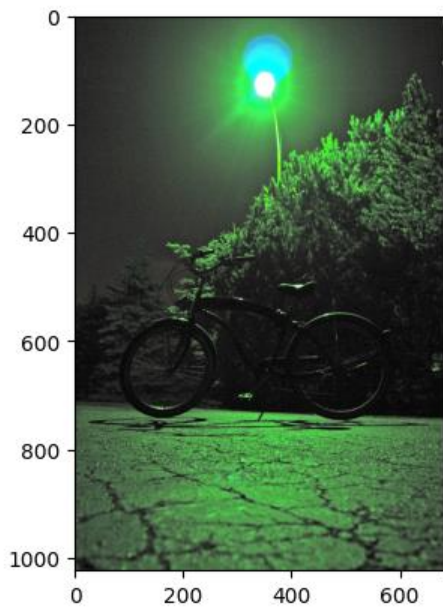


圖 12 增強前

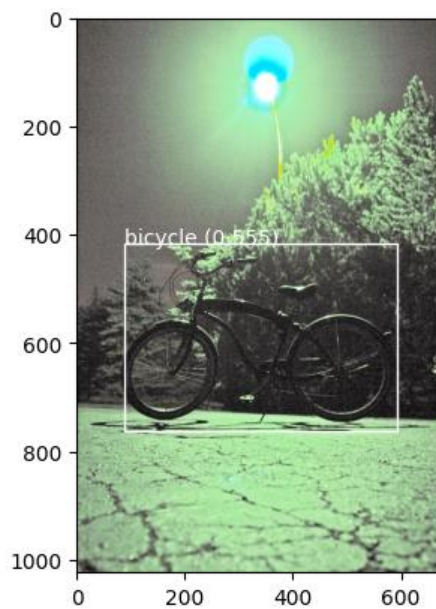


圖 13 增強後



圖 14 增強前

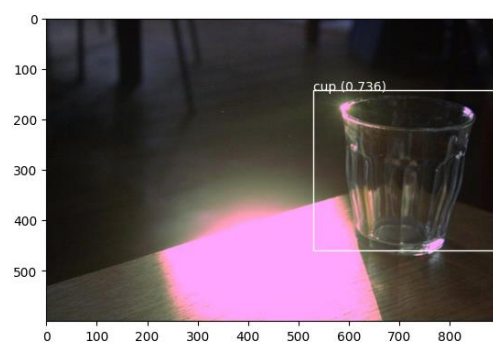


圖 15 增強後



圖 16 增強前



圖 17 增強後

- 實際照片

再來我自己拍攝幾張暗的影像

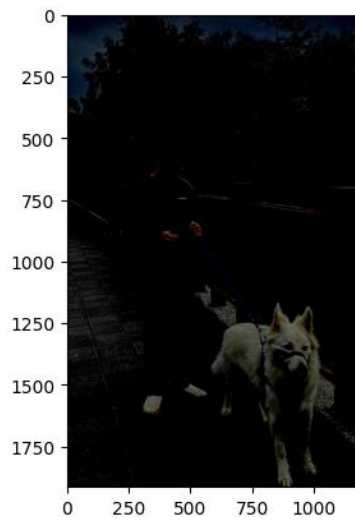


圖 18 增強前

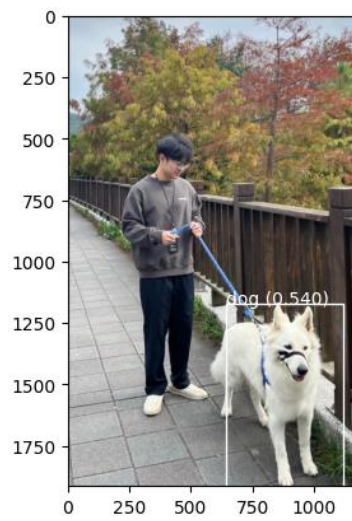


圖 19 增強後



圖 20 增強前

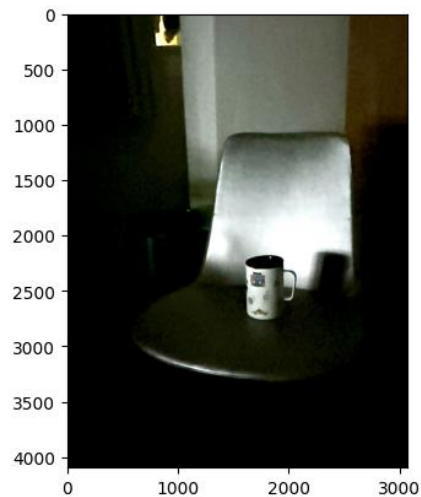


圖 21 增強後

結論：

這次實驗結果，我覺得兩個效果都還不錯，IAT 模型對於圖形的強化是第一個，YOLOv3 對於物件的偵測對於暗光強化後的影像，效果也有更顯著的提升，觀察結果我發現，兩個 **weights**(增強前和增強後訓練的權重)對於某些特定的物件還是沒有辦法很有效的偵測，像是我自行拍攝的素材，除了狗以外，其他像是人、椅子和杯子都沒有顯示 **bounding box**，我認為有兩個原因，第一個是訓練時間不夠長，因為我使用 Google Colab 訓練，隨時會有斷線的風險，所以只訓練了 30 個 **epoch**，第二個是因為資料集只有 7000 多張，算是非常小樣本的數據，如果是要捕捉真實世界拍下的物件，可能還需要更多資料及讓模型可以學會辨別對應物件的特徵。

總結來說，這次的期末報告，我認為我學到很多，包括像是 **data** 的前處理，模型裡面的架構和應用以及為何要這樣設計，都有更深入的了解。也學到如何將現有的模型在跟其他領域的概念做整合，進而得到更好的結果，這次的專題雖然只是一個小樣本的訓練，還不足以應用到真實世界去使用，但我會繼續鑽研這方面的知識，學習更新的模型架構，期望自己能在這塊領域繼續前進。最後謝謝老師和助教們這學期的指導，你們辛苦了。

相關資料:

- <https://github.com/eriklindernoren/PyTorch-YOLOv3>
- <https://www.appliedaicourse.com/>
- <http://cs-chan.com/doc/cviu.pdf>
- <https://towardsdatascience.com/enhancing-low-light-images-using-mirnet-a149d07528a0>
- <https://github.com/cuiziteng/Illumination-Adaptive-Transformer>
- <https://arxiv.org/abs/2205.14871>