

Rivelazione di oggetti in condizioni di scarsa luminosità

Pascarella Vincenzo

Reggio Raimondo

Riccio Emanuele

Progetto ESM - gruppo 3

(a.a. 2020-2021)

1 - Introduzione

In questo progetto ci siamo interessati al problema dell'object detection che consiste nell'individuare istanze di oggetti all'interno di un'immagine e nel classificarle come appartenenti ad una certa classe. In particolare abbiamo trattato la rilevazione di oggetti in condizioni di scarsa luminosità. Infatti attualmente i sistemi di visione artificiale sono utilizzati in applicazioni critiche come autonomous drive o videosorveglianza quindi nasce la necessità di garantirne robustezza e affidabilità in condizioni non ideali come la scarsa luminosità. Le immagini in condizioni di scarsa luminosità sono affette da rumore e mancanza di dettagli, ciò rende difficile l'estrazione di features map semanticamente significative. Per questo motivo gli attuali rivelatori di oggetti sono incapaci di affrontare adeguatamente queste situazioni e allo stesso modo un algoritmo di riduzione del rumore all'avanguardia è anche insufficiente per gestire il rumore che si verifica frequentemente insieme ai dati in condizioni di scarsa illuminazione. In altre parole la scarsa illuminazione "altera" le caratteristiche degli oggetti, cioè lo stesso oggetto in condizioni di luce intensa e scarsa produce caratteristiche ampiamente diverse. Pertanto, il rilevamento di oggetti in condizioni di scarsa illuminazione non deve essere affrontato con leggerezza, ma richiede invece un'attenta considerazione ed è necessario un set di dati dedicato per far avanzare i progressi.

2 - Approccio implementato

2.1 Rete utilizzata

La rete utilizzata in questo progetto è Retinanet [1], quest'ultima è una rete one-stage composta da una backbone (*ResNet*) e due sottoreti (figura 1). La prima effettua la classificazione, ovvero predice per ogni anchor la probabilità della presenza di ogni classe di oggetti. In parallelo la seconda esegue la regressione convoluzionale delle anchors predicendo l'offset tra quest'ultime e i ground-truth bounding boxes, ossia i box che delimitano correttamente gli oggetti presenti all'interno dell'immagine.

Retinanet si differenzia dalle reti "two-stage" in quanto in un solo stadio individua e classifica un insieme di posizioni candidate a contenere gli oggetti. Nonostante l'alta velocità e la loro semplicità, i riconoscitori "one-stage" hanno sempre avuto problemi nell'accuratezza delle predizioni. La rete utilizzata risolve tale problema introducendo una nuova loss function chiamata *focal loss* e la sua architettura è caratterizzata dalla cosiddetta Features Pyramid Network (**FPN**) diventata una delle componenti alla base di molti riconoscitori nella computer vision.

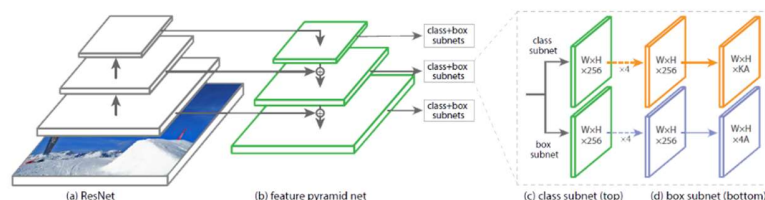
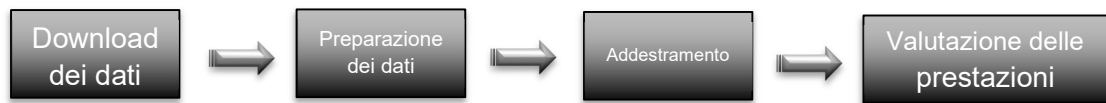


Figura 1

2.2 Passi Eseguiti



Download dei dati: Abbiamo scaricato dalla repository ufficiale il dataset ExDark composta da circa 7000 immagini.

Preparazione dei dati: Delle 7000 immagini ne sono state utilizzate 3000, suddivise in: 2000 per il training, 1000 per la validation e 1800 per il test. Come richiesto dalla rete abbiamo generato dei file CSV per ogni set di immagini attraverso uno script python [2]. Ogni file contiene, per ogni immagine: il path, le dimensioni dei box, e le classi presenti. Oltre a questi abbiamo realizzato un altro file CSV contenente il nome delle classi e un valore numerico associato ad ognuna di esse.

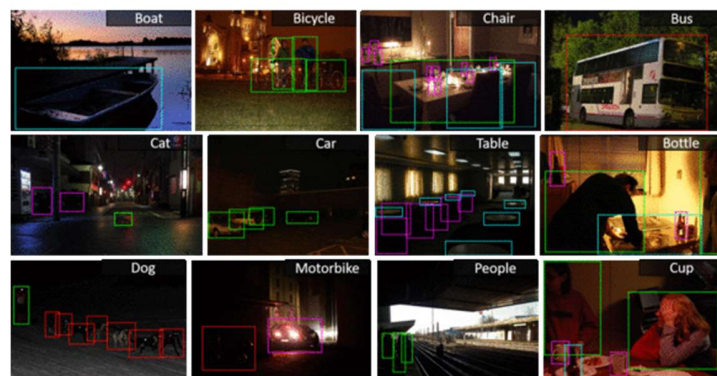
Addestramento: Utilizzando lo script *train.py* fornito dalla repository [3] abbiamo addestrato la rete valutando in maniera sperimentale i parametri di input (freeze-backbone, batch-size, epochs, learning rate, steps-for-epochs).

Valutazione delle prestazioni: Utilizzando lo script *evaluate.py* fornito dalla repository [1] abbiamo valutato le prestazioni ottenute. Al fine di testare la rete su una singola immagine abbiamo utilizzato lo script *resnet50_retinanet.py* [5] modificando alcune istruzioni.

3 - Dataset

Il dataset utilizzato è Exclusively Dark (*ExDARK*) [4]. Esso è una raccolta di 7.363 immagini in condizioni di scarsa illuminazione da ambienti molto poco illuminati al crepuscolo (cioè 10 condizioni diverse) con 12 classi di oggetti annotate sia a livello di classe di immagini che a riquadri di delimitazione degli oggetti locali.

Di seguito sono riportate 4 immagini che rappresentano rispettivamente: esempi di dataset, percentuale delle immagini per classe contenute nel dataset, esempi di immagini in condizioni di illuminazione diverse, istogramma del numero di immagini per le 10 condizioni di luminosità.



4 - Risultati sperimentali

Descrivete gli esperimenti che avete realizzato e i risultati ottenuti (inserite tabelle/grafici). Ricordate di definire le misure utilizzate. Cercate di giustificare i risultati quando fate un confronto (es. Le prestazioni si riducono all'aumentare della dimensione del blocco perché).

Sperimentalmente abbiamo riscontrato che i parametri per ottenere un mAP migliore sono: batch size 6, learning rate 10^{-4} , epochs 15 (*tabella 1*).

Per definire l'mAP dobbiamo prima far riferimento a due concetti fondamentali: *precision* e *recall*. La prima è il rapporto tra il numero di oggetti riconosciuti e giusti e il numero totale di oggetti riconosciuti. Invece la seconda è il rapporto tra il numero di oggetti riconosciuti e giusti e il numero totale di oggetti nell'immagine di quella classe. Si definisce AP, l'area sottesa dalla curva precision-recall (*figura 2*). Di conseguenza definiamo l'mAP come la media delle AP di ogni classe di oggetti. Riportiamo in *figura 3* un grafico che descrive l'andamento della loss function in funzione del learning rate, si può notare che il punto di minimo della curva ha ascissa pari a 10^{-4} .

Come si evince dai risultati ottenuti in fase di test, la rete presenta principalmente difficoltà nel riconoscimento dei tavoli, infatti tale classe è caratterizzata da un AP pari a 0,2670 (*figura 4*).

Batch size	Learning rate	Epochs	mAP
6	$1e^{-4}$	15	0.4344
4	$1e^{-5}$	10	0.1723
6	$1e^{-5}$	15	0.2742
6	$1e^{-3}$	15	0.1971
6	$1e^{-6}$	15	0.0739

Tabella 1

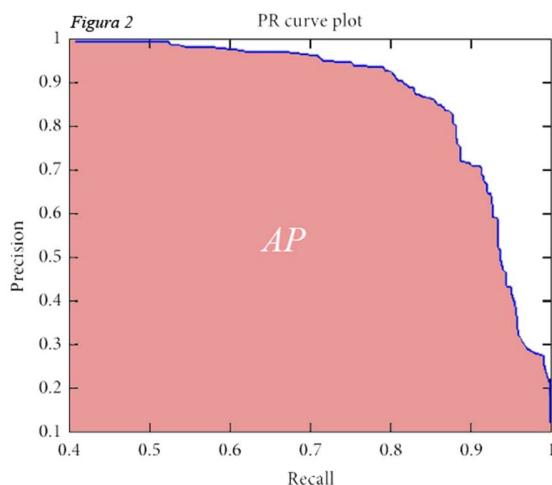
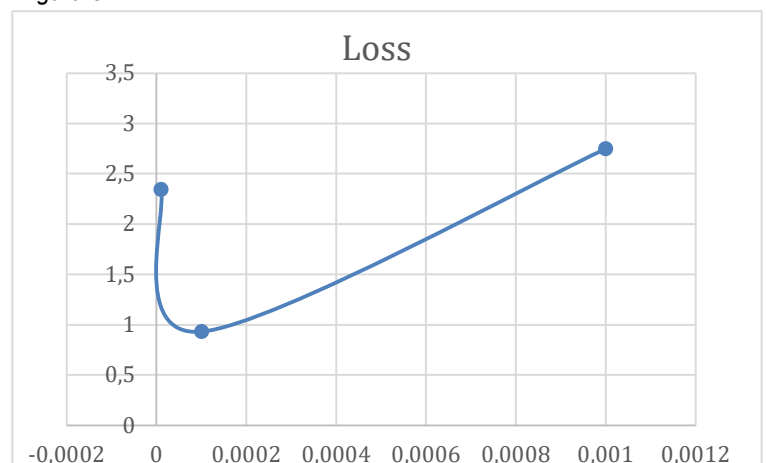


Figura 3



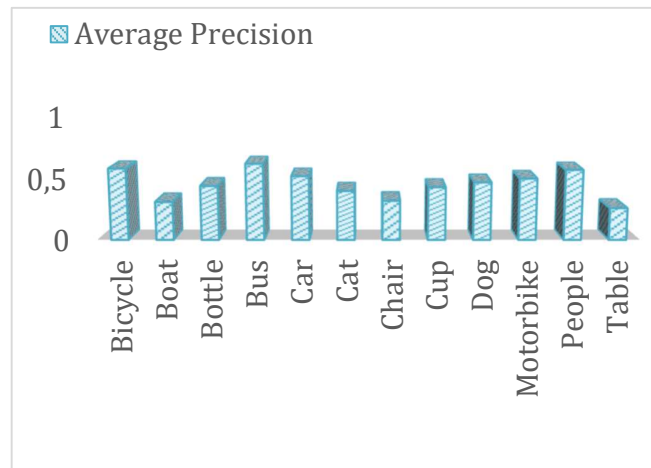


Figura 4

Riferimenti bibliografici

- [1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection" International Conference on Computer Vision (ICCV), 2017.
- [2] [CSV generator.py - Pastebin.com](#)
- [3] <https://github.com/fizyr/keras-retinanet>
- [4] Y. P. Loh, and C. S. Chan, "Getting to know low-light images with the Exclusively Dark dataset" Computer Vision and Image Understanding (CVIU), 2019. <https://github.com/cs-chan/Exclusively-Dark-Image-Dataset>
- [5] [resnet50_retinanet.py - Pastebin.com](#)