Relazione progetto 1

Stefano Ruggiero mat.364989

L’obiettivo di questo primo progetto è stato quello di testare, valutare ed imparare l’utilizzo dei modelli ritenuti lo stato dell’arte per i task di object detection e segmentation. Le librerie utilizzate per il progetto sono state detectron2 (RCNN) e ultralytics (YOLO).

Detectron2 è il successore di Detectron e maskrcnn-benchmark, supporta numerosi progetti di ricerca sulla computer vision e applicazioni di produzione in Facebook. Mette, inoltre a disposizione ampia serie di baseline e modelli pre-addestrati, cosa molto utile per l’hardware limitato a disposizione.

Il primo codice prodotto è progetto\_1.ipynb: in questo primo file abbiamo testato le potenzialità delle RCNN facendo un primo test sul dataset con in modello pre-addestrato sul dataset COCO. Ovviamente i risultati sono stati ottimi. Ma è stato molto utile per capire meglio il funzionamento di detectron2. Infatti, questa libreria ha un tipo specifico di input: dataset in formato COCO (file e associato un file .json con le labels) e contiene molti strumenti per la configurazione personalizzata del modello che si desidera utilizzare.

A tal proposito, nel codice progetto\_1\_2.ipynb, l’obiettivo è stato proprio quello di utilizzare un dataset non in formato COCO e soprattutto l’apprendimento di una classe non presente nei modelli pre-addestrati (contiene 80 classi in totale). Il task è stato quello di segmentation. La classe appresa dalla rete RCNN è stata quella dei palloncini, anche con dei buoni risultati (validation):

Bounding box:

'bbox', {'AP': 73.99201924874454, 'AP50': 87.61313821351739, 'AP75': 83.59329701967158, 'APs': 0.0, 'APm': 59.74862344023673, 'APl': 87.75364502442075}

Segment:

'AP': 79.67371020843741, 'AP50': 85.50598828879848, 'AP75': 85.50598828879848, 'APs': 0.0, 'APm': 59.2493551876196, 'APl': 96.77404096684973

Note:

* AP (precisione media): È la metrica principale utilizzata per misurare l'accuratezza dei modelli di rilevamento degli oggetti. Rappresenta l'area sotto la curva precisione-recall. Tutte le varianri di AP:
* AP: Precisione media complessiva su tutte le soglie IoU (Intersection over Union) e le dimensioni degli oggetti.
* AP50: precisione media con una soglia IoU di 0,5.
* AP75: Precisione media con una soglia IoU di 0,75.
* APs, APm, APl: Precisione media per oggetti piccoli, medi e grandi, rispettivamente.
* AR (Average Recall): Questa metrica misura la capacità del modello di recuperare tutti gli oggetti rilevanti. Analogamente ad AP, si vedranno variazioni come AR1, AR10, AR100, ARs, ARm, APl che rappresentano il richiamo a un numero diverso di rilevamenti e di dimensioni degli oggetti.

Nel codice di progetto\_1\_3.ipynb, abbiamo effettualo la conversione del file label .odgt a .json del dataset crowd human. E’ un dataset in cui l’unica classe da saper riconoscere è quella della persona, la sfida in questo caso e di riconoscere tutte le persone in una folla di persone o in scenari con prospettive molto difficili. In questo caso il task è stato di object detection, e la rete utilizzata è FASTER-RCNN addestrata per 300 iterazioni ed i risultati non sono stati in questi casi buoni (validation):

'AP': 16.36097941488944, 'AP50': 31.709529334270055, 'AP75': 15.033665037347008, 'APs': 3.171532649694159, 'APm': 14.563460096689202, 'APl': 24.127557824273573})])

Nell’ultimo codice progetto\_1\_4.ipynb, invece, la libreria utilizzata è stata quella Ultralytics YOLO11. Per comparare FASTER RCNN all’ultima versione di YOLO, è stato utilizzato sempre il dataset crowd human, ovviamente sono state prima convertite le etichette per poter essere accettate dal modello (YOLO accetta file .txt e bisogna creare file .yaml). I risultati con 4 epoche di addestramento e image size pari a 32 (il minimo) non sono buoni, con un AP di solo 0.0217. Purtroppo per limitazioni tecniche e di tempistiche non si è potuto aumentare il numero di epoche, probabilmente con un numero superiore di epoche i risultati sarebbero stati migliori.

Si tiene a precisare che i tempi di attesa per la valutazione sul validation set, sono stati decisamente più lunghi su detecrtro2+Faster RCNN rispetto alla rete YOLO11 (3 minuti vs 126 minuti).

Analizzando i risultati, le possibili ipotesi per questi risultati deludenti possono essere dovuti ad un training non adeguato, oppure la challenge del dataset risulta molto complessa per entrambi i modelli.