1) STUDIOMAPP

Studiomapp è una startup con sede a Ravenna ed è specializzata in location intelligence ovvero applicazioni di intelligenza artificiale applicata alla computer vision. In particolare si occupa di analisi del territorio tramite l’analisi di immagini satellitari in alta risoluzione.

L’anno scorso è stata premiata dal pentagono nella sfida xView arrivando al 4 posto, la sfida consisteva nell’usare l'intelligenza artificiale per analizzare immagini satellitari in alta risoluzione, rilevare disastri naturali e quantificarne l’impatto in modo da fornire informazioni ai soccorritori per organizzare al meglio le operazioni di soccorso.

Analizzare immagini satellitari comporta difficoltà e sfide aggiuntive rispetto all’analisi di normali immagini a causa della riduzione della risoluzione minima di rilevamento in quanto essendo che i satelliti orbitano a migliaia di kilometri dalla superficie terrestre, oggetti anche di grandi dimensioni apparirebbero molto piccoli e difficili da distinguere nelle immagini catturate.

2) SCOPO DELLO STAGE

Durante lo stage sono stati portati avanti due progetti principali.

Il primo consisteva nello sviluppare un algoritmo per migliorare le metriche relative alle detection effettuate nella xView challenge.

Il secondo progetto consisteva nello sviluppare un sistema di tracking di oggetti su video, i video utilizzati per testare la qualità del sistema sono stati presi da un dataset messo a disposizione da un’ altra sfida chiamata DroneChallenge, i cui video sono stati ripresi da droni invece che da satellite.

Prendere i video da questa sfida è stato comodo in quanto gli organizzatori mettevano a disposizione uno script per calcolare le metriche per valutare la qualità del tracking calcolate confrontando gli oggetti tracciati dal sistema con oggetti tracciati manualmente e quindi senza errori ai fini di misurare l’accuratezza del tracking.

L’algoritmo sviluppato può comunque essere applicato a qualsiasi tipo di video indipendentemente dal suo contenuto o dalle modalità di ripresa.

3) OBJECT DETECTION

L’object detection ha come obiettivo quello di interpretare il contenuto di un’immagine riconoscendo specifici oggetti presenti in essa. Riconoscere un oggetto in un’immagine significa individuare la sua posizione e classificarlo in una categoria di appartenenza. Ovviamente possono venire riconosciuti solo gli oggetti per cui il modello è stato allenato a riconoscere.

Ogni elemento individuato verrà poi evidenziato da una bounding box che è un rettangolo al cui interno è rappresentato l’ oggetto, vicino alla bounding box viene inoltre mostrata la sua categoria di appartenenza ed un valore di probabilità detto score che indica la probabilità che la categoria predetta sia effettivamente quella corretta.

Infine per classificare un’immagine viene tipicamente utilizzata una CNN (convolutional neural network)

Allo stato dell’arte per individuare e classificare più oggetti in un’immagine viene utilizzato un modello più avanzato chiamato Faster R-CNN, che è formato da due reti neurali profonde, delle quali, una serve per individuare le probabili bounding boxes nell’immagine all’interno delle quali potrebbero trovarsi degli oggetti ed una CNN per provare a classificare gli oggetti presenti all’interno delle bounding boxes ipotizzate.

4) TRE MOTIVI PER PREFERIRE LAVORARE SU IMMAGINI IN BASSA RISOLUZIONE

Per i modelli esistenti lavorare direttamente su immagini in alta risoluzione risulta complicato per i seguenti motivi:

Il primo motivo è che i modelli sviluppati con i frameworks più comuni come tensorflow tendono a ridimensionare automaticamente le immagini ricevute in input in modo tale da adattarle alla loro dimensione massima. Questo processo comporta inevitabilmente una perdita di risoluzione che potrebbe non permettere di individuare alcuni elementi oppure rendere più complicata la loro individuazione.

Inoltre, processare immagini in bassa risoluzione è più efficiente in quanto devono essere elaborati un minor numero di pixels.

Il terzo motivo è che i datasets utilizzati per allenare i modelli sono composti da milioni di immagini in bassa risoluzione, come ad esempio ImageNet che ne contiene 15 milioni. E’ per questo motivo che i modelli vengono allenati con immagini a bassa risoluzione in quanto più facili da reperire ed in maggiore quantità

5) OBJECT DETECTION CON FRAMMENTAZIONE

L’idea per aggirare i problemi appena discussi consiste nel suddividere un’ immagine in alta risoluzione in diverse sotto-immagini dette regioni di dimensione uguale in modo tale che ogni regione risulti di dimensioni gestibili da un modello senza che l’immagine venga ridimensionata

Successivamente si effettua una detection su ogni regione trattandola come se fosse una singola immagine e poi si rimettono insieme tutte le bounding boxes individuate nelle regioni sull’immagine originale

Infine viene applicato un algoritmo di non-max suppression che serve per eliminare le bounding boxes molto sovrapposte in quanto è molto probabile che esse rappresentino lo stesso elemento.

Il problema è che con questo metodo gli elementi che si trovano in mezzo tra due o più regioni verrebbero individuati una volta per ogni regione ed ogni volta verrebbero conteggiati come se fossero oggetti distinti

(Per esempio questa immagine è virtualmente divisa in 4 sotto-immagini ed essendo che l’edificio raffigurato cade proprio in mezzo al confine tra le 4 regioni vengono quindi individuati 4 edifici distinti invece che 1)

Per far fronte a questo problema è stato quindi ideato un algoritmo di post-processing per ricomporre correttamente gli oggetti che sono stati frammentati

6) ALGORITMO PER RICOMPORRE LE DETECTIONS FRAMMENTATE

L’algoritmo consiste nel raggruppare tutte le boxes che potrebbero appartenere allo stesso elemento ed unirle per formare una singola box, questo compito viene eseguito tramite metodi euristici, ovvero vengono raggruppate insieme le boxes che rispettano dei criteri di correlazione come essere della stessa categoria, essere vicine o intersecate, essere state individuate in regioni diverse ed avere dimensioni compatibili.

Nella figura a sinistra sono presenti 6 boxes, applicando l’algoritmo le boxes contrassegnate col numero 1 vengono messe nello stesso gruppo in quanto queste boxes soddisfano i criteri di correlazione e verranno quindi sostituite da una nuova box che le racchiude tutte in modo da costruire la box raffigurante l’edificio per intero. Le boxes contrassegnate col numero 2 vengono raggruppate in un gruppo diverso e la loro unione raffigura un secondo edificio differente dal primo

EXTRA) RICOMPOSIZIONE DETECTION

Nella figura si può vedere un esempio di un’immagine sulla quale è stato applicato l’algoritmo. L’immagine è stata divisa in 4 regioni con sovrapposizione, avere sovrapposizione tra le regioni è utile perchè rende più efficace la detection.

Nella versione a sinistra si sono presenti le boxes individuate con la detection senza aver applicato alcun algoritmo di post-processing, vengono erroneamente individuate sette persone invece che tre dando quindi una errata interpretazione dell’immagine

Nella versione a destra viene nella stessa immagine applicato l’algoritmo per ricomporre correttamente le boxes in modo che vengano individuate 3 persone in modo corretto. In questo caso l’immagine viene interpretata correttamente come farebbe qualsiasi essere umano.

7) RISULTATI DETECTION

Il dataset xView utilizzato per calcolare le metriche finali è composto da 60 categorie di oggetti diverse, alcune molto simili tra loro e quindi molto difficili da distinguere, inoltre, a causa della distanza dalla quale sono state scattate le foto gli oggetti più piccoli risultano molto sfocati, difficili da riconoscere perfino per un essere umano.

Le metriche utilizzate per misurare la qualità della detection sono l’mAP e l’F1, queste metriche sono tipicamente utilizzate per misurare la qualità degli algoritmi di machine learning. Esse vengono calcolate a partire dalla precision e dalla recall delle detections che sono metriche di base più immediate da calcolare. Di conseguenza un aumento di entrambe le metriche comportano anche un aumento delle metriche F1 e mAP.

L’mAP e F1 risultano avere un valore di circa 25% che se ottenuto su un normale dataset non sarebbe un grande risultato ma considerata la difficoltà nell’individuare gli oggetti presenti è un risultato più che accettabile.

L’mAP migliora fino al 1.7% mentre l’F1 subisce un notevole miglioramento arrivando ad un aumento di quasi del 17%. L’algoritmo è parametrizzabile e questi risultati sono stati ottenuti con la migliore combinazione di parametri.

8) OBJECT TRACKING

L’object detection può essere estesa per eseguire l’object tracking. Il tracking consiste nel tracciare degli specifici elementi attraverso una sequenza di frames di un video, per tracciare un oggetto e riconoscere che è sempre lo stesso viene ad esso assegnato un ID univoco in modo da indentificarlo attraverso tutti i frames nei quali esso appare.

Tuttavia non è semplice tracciare un oggetto se pensiamo che durante un video qualsiasi oggetto in movimento può essere soggetto a problemi di occlusione, sfocatura, cambio di background, variazione di luminosità, deformazione o accelerazione.

9) ALGORITMO DI TRACKING

L’algoritmo di tracking sviluppato consiste in una pipeline formata da vari steps

Inizialmente, viene effettuata una detection su ogni nuovo frames in modo da individuare tutti gli elementi presenti in esso e le loro boxes. (Nella figura in alto, i rettangoli in verde corrispondono alle boxes degli elementi individuati)

Invece per tracciare gli elementi presenti nel video vengono sfruttati dei tracker, un tracker per ogni oggetto, questi trackers dispongono di un filtro, detto filtro di kalman, che serve per predirre la posizione della nuova bounding box dell’elemento tracciato rispetto alle sue posizioni predenti, la sua velocità e la sua traiettoria e migliorando la sua accuratezza tramite le individuazioni nei frames successivi. (Nella figura affianco, i rettangoli in rosso corrispondono alle boxes predette dai tracker con il rispettivo ID)

Il passo successivo consiste nell’assegnare ogni oggetto individuato dalla detection al tracker corretto, ovvero al tracker che lo tracciava nei frames precedenti confrontando le loro boxes in modo da assegnare l’ID corretto ad ogni elemento tracciato.

10) ASSEGNAZIONE TRACKER-DETECTION

Per effettuare l’assegnazione bisogna calcolare un valore di similarità tra ogni coppia tracker-detection, in modo tale che tanto più questo valore è maggiore, tanto è più probabile che un certo oggetto appartenga ad un determinato tracker

Questo valore è basato principalmente sulla percentuale di intersection over union tra le aree delle bounding boxes individuate e le aree delle bounding boxes predette dai trackers, in aggiunta, questo valore è influenzato anche dalla differenza tra le dimensioni delle due aree e la differenza tra il rapporto dei loro lati. Questo ragionamento è basato sull’assunzione che tra un frame ed uno successivo un oggetto non cambierà di molto le sue dimensioni e la sua posizione.

Dopo aver calcolato questo valore per ogni coppia tracker-detection viene utilizzato un algoritmo di ottimizzazione per effettuare le assegnazioni in modo ottimale massimizzando il valore di similarità totale che viene assegnato.

Nella figura a sinistra il tracker 1 ha un valore di similarità molto alto con la detection B in quanto le loro boxes sono molto sovrapposte e le loro dimensioni quasi identiche quindi il tracker 1 nel frame corrente traccierà l’oggetto B, analogamente per il tracker 2 e la detection C. Il tracker 3 non ha nessuna detection nelle vicinanze ed ha quindi un valore di similarità molto basso con tutte le altre detections, il suo oggetto tracciato è probabilmente scomparso dal video e quindi il tracker verrà ignorato, se dopo un certo numero di frames consecutivi il tracker non è ancora stato assegnato allora verrà cancellato. La detection A non ha nessun tracker nelle vicinanze, si tratta probabilmente di un oggetto che è appena apparso nel video e quindi verrà costruito un nuovo tracker, il numero 4 per iniziare a tracciarlo.

11) RISULTATI TRACKING

Il dataset utilizzato per misurare la qualità del tracking, al contrario del più complesso dataset xView, richiedeva solamente di tracciare oggetti appartenenti a 5 categorie diverse che comprendevano persone, biciclette, e tre tipi di veicoli.

I risultati sono stati in generale molto positivi, circa il 70% degli oggetti presenti sono stati tracciati per almeno l’80% della loro durata.

Altre due metriche molto importanti sono IDF1 e MOTA che misurano l’accuratezza del sistema di tracking in generale. Entrambi i valori si attestano intorno al 50% che è un valore accettabile comparabile con altri algoritmi di tracking già in circolazione.

Essendo che alla fine anche il tracking è basato sulla detection è stato provato che aumentare la qualità della detection aumenta anche tutte le metriche del tracking, in particolare simulando di avere detection perfette si è riusciti ad ottenere qualità di tracking quasi ottimale.