**OBJECT DETECTION**

**Descrizione**:

Il nostro progetto si focalizza sul rilevamento di oggetti, in particolare automobili, utilizzando il modello pre-addestrato YOLOv5.

L'obiettivo principale è valutare l'efficacia di YOLOv5 su più datasets di immagini, calcolare le metriche di precisione e visualizzare i risultati annotati.

**Obiettivi**:

1. Testare l'efficacia di YOLOv5 su più dataset specifici.
2. Calcolare le metriche di precisione, in specifico la classe riferita alle automobili nei 2 dataset più grandi, mentre nel dataset creato da noi valutare anche le classi riferite a camion, biciclette, persone e moto.
3. Visualizzare e salvare i risultati annotati per una valutazione visiva.

**Strumenti e Tecnologie**:

* Ambiente di sviluppo creato con Miniconda configurato durante le lezioni.
* Diverse librerie Python, con le loro versioni, elencate nel file requirements.txt.

**Processo**: Per ogni immagine nel dataset, il progetto esegue i seguenti passaggi:

1. **Caricamento dell'immagine**: L'immagine viene caricata dal percorso specificato.
2. **Rilevamento degli oggetti**: Il modello YOLOv5 viene applicato all'immagine per rilevare gli oggetti presenti.
3. **Calcolo delle metriche di precisione**: Vengono calcolate le metriche di precisione, confrontando i risultati del modello con le annotazioni di ground truth.
4. **Salvataggio e visualizzazione dei risultati annotati**: I risultati annotati vengono salvati e visualizzati per una revisione visiva.

**Statistiche Finali**:

Al termine dell'elaborazione di tutte le immagini nel dataset, il progetto calcola e stampa le seguenti statistiche:

* **Numero totale di immagini processate**.
* **Tempo medio di elaborazione per immagine**.
* **Precisione media per le classi prese in esame**.

**Descrizione dei datasets:**

Il primo e il secondo sono datasets molto grandi che raccolgono fotogrammi e video aerei che catturano vari tipi di veicoli mentre attraversano una strada, mentre il terzo si tratta di un piccolo dataset creato ad hoc da noi, e annotato tramite il tool “LabelImg”.

Ogni fotogramma è annotato con le coordinate dei bounding box che delimitano ciascun veicolo, essenziali per l'addestramento e la valutazione di modelli di rilevamento degli oggetti.

**Definizioni:**

Per ogni immagine, il modello YOLOv5 esegue il rilevamento delle automobili e restituisce una lista di bounding box predetti con coordinate (x1, y1, x2, y2), un punteggio di confidenza e una classe di oggetto.

Le annotazioni di verità a terra sono caricate da un file CSV e contengono un bounding box con coordinate (xmin, ymin, xmax, ymax), eccetto il CSV relativo al nostro dataset creato ad hoc, nel quale abbiamo inserito un attributo per identificare la classe.

Essendo il nostro un dataset piccolo abbiamo provato ad identificare altri oggetti come persone, camion, biciclette e moto.

Per ogni bounding box rilevato, il codice confronta questo box con i bounding box annotati di verità a terra per quella specifica immagine.

La funzione “calculate\_iou” calcola la metrica IoU (intersection over union) tra il bounding box rilevato e ciascun bounding box di verità a terra, ovvero il rapporto tra l'area di intersezione dei due bounding box e l'area della loro unione.

1. **True Positives (TP)**: Il numero di istanze correttamente classificate come positive, ovvero con l’IoU >= 0.5.
2. **False Positives (FP)**: Il numero di istanze erroneamente classificate come positive, ovvero se un bounding box rilevato non corrisponde a nessuna annotazione di verità a terra (IoU < 0.5).
3. **False Negatives (FN)**: Il numero di istanze erroneamente classificate come negative, ovvero se esistono annotazioni di verità a terra che non sono state associate a nessun bounding box rilevato (IoU < 0.5).
4. **Precision:** La proporzione di vere istanze positive tra tutte quelle classificate come positive.

È calcolata come: Precision = TP / (FP+TP)​.

1. **Recall**: La proporzione di vere istanze positive tra tutte quelle che realmente sono positive.

È calcolata come: Recall = TP / (TP+FN).

**Statistiche dei 3 dataset:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Statistiche** | **1° Dataset** | **2° Dataset** | **3° Dataset** |
| Total images | 1001 | 301 | 16 |
| Total images processed | 1001 | 301 | 16 |
| Average time per image | 0,08s | 0,08s | 0,10s |
| Total true positives | 517 | 2167 | 35 |
| Total false positives | 192 | 13 | 10 |
| Total false negatives | 42 | 1492 | 6 |
| Precision | 0,73 | 0,99 | 0,78 |
| Recall | 0,92 | 0,59 | 0,85 |

### 

### **Valutazione dei risultati:**

* **Dataset 1**:
  + **Precisione** **moderata** (0,73) significa che una proporzione considerevole dei rilevamenti sono corretti.
  + **Alta** **recall** (0,92) significa che il modello rileva la maggior parte degli oggetti presenti.
* **Dataset 2**:
  + **Altissima precisione** (0,99) indica che quasi tutti i rilevamenti del modello sono corretti.
  + **Bassa recall** (0,59) indica che il modello manca di rilevare una significativa proporzione degli oggetti presenti.
* **Dataset 3:**
  + **Precisione moderata** (0,78) indica che il 78% delle rilevazioni del nostro modello sono effettivamente corrette.
  + **Recall alta** (0,85) rileva la maggior parte degli oggetti presenti.

**Conclusioni:**

Il modello si comporta in modo molto diverso sui tre dataset.

Sul 1° dataset il nostro modello è bilanciato tra precisione e recall.

Il 2° dataset è estremamente preciso ma a scapito di un alto numero di mancati rilevamenti, questo è dovuto all’altissimo quantitativo di automobili presenti in ogni immagine.

Il 3°dataset risulta non molto preciso in quanto, da come si può osservare dalle immagini elaborate, scambia molte automobili come camion.

Ha un buon valore di recall, tuttavia abbiamo notato che non riesce a rilevare correttamente tutti gli elementi presenti se sono aggregati tra loro.