

Esercitazione

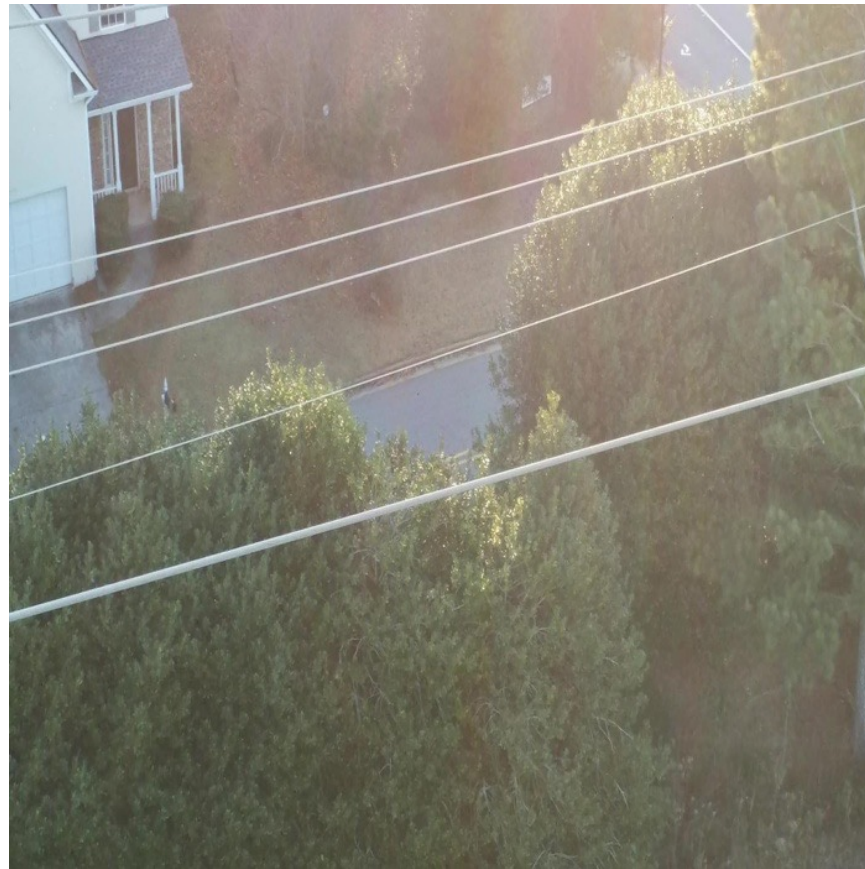
Progetto

Progetto

- Cable and tower detection and segmentation
- Task: identificare i pixel dei cavi e delle torri
- Costruire un modello di segmentazione

https://github.com/R3ab/ttpla_dataset

Dataset



Dataset

- Il progetto è costituito da 2 dataset:
 - Train: 842 immagini 700x700
 - Test: 400 immagini 700x700
- Per ogni immagine sono date le informazioni su etichetta e maschera degli oggetti presenti e sono memorizzate nei file train.json e test.json nelle relative folder

Progetto

- Si deve costruire un modello di segmentazione che individui gli oggetti e i relativi pixel
- La scelta del modello, della backbone e dei parametri è oggetto di valutazione.

https://drive.google.com/file/d/1lwGSf6c1hZoh3gaAEsD7JmOE1PR7Tgrb/view?usp=drive_link

Suggerimenti

- Colab/Azure/ecc. con salvataggio dello stato
- Test su porzioni del dataset per la stima degli iper-parametri
- «Arricchimento» del dataset di training
- Personalizzazione del modello
- Modelli semplici vs modelli complessi
- Loss standard vs loss custom

Info:

https://pytorch.org/tutorials/beginner/saving_loading_models.html

Suggerimenti (2)

- Analisi del dataset
 - Che tipo di immagini contiene? È possibile aggiungere altre immagini simili?
- Scelta del modello
 - Quale scegliere? Studio della letteratura
 - Quali architetture? Quali parametri?
- È possibile aggiungere informazioni di contesto (i.e. forma geometrica) ?
- Analisi degli errori
- Tuning dei modelli
 - Analisi della loss e delle maschere generate

Requisiti

- Definire e addestrare un modello di segmentazione
- Descrivere in una presentazione le scelte progettuali (in particolare modello e loss utilizzate) e tutti i parametri utilizzati nella sperimentazione. Analisi di alcuni campioni (analisi qualitativa).
- Consegnare notebook (e/o file sorgente) utilizzati per il training, il notebook di test, presentazione e dump del modello.
- Generare un file json **matricola.json** con le predizioni (bbox e segmentazione)

Requisiti (2)

- File json **matricola.json** con le predizioni in formato coco (vedi notebook di test)

Es. per i box

```
[{'image_id': 0, 'category_id': 0, 'bbox': [291.0, 40.0, 86.0, 653.0], 'score': 0.3679635524749756}, ...]
```

Per le mask

```
[{'image_id': 0, 'category_id': 0, 'segmentation': {'size': [700, 700], 'counts': 'kUj68_e08i\\10ScN4W]10ebN4oR2CeZ06`bN3j\\10Q]N@lFa0_l19_bN2n\\1E\\cN0hV1FPZNe0a8[0`Gf0Qe1DQ[N7V8HlG7Se1NlZN;h7EYH;oU1Z0lQ0a0WH0c7M`H0nn0BkQ0Le6f0dH4R7LoH3^7_0QI4Z73^H4n64lH0Q7MRI0Sd1FjjNGe\\1>WcNESbi6'}, 'score': 0.3679635524749756}, ...]
```

Protocollo di valutazione

- Valutazione generale del progetto
- Valutazione della mAP sul test set.

Si accede all'orale se il valore (BASELINE 1)

- Box - mAP^{50%} -> 44
- Mask - mAP^{50%} -> 34

Seconda baseline (BASELINE 2)

- Box - mAP^{50%} -> 48
- Mask - mAP^{50%} -> 36

Protocollo di valutazione

Si accede all'orale se si superano i valori minimi di mAP.

Per tutti i progetti che superano la baseline 1 si costruisce una classifica.

Metriche da considerare: $mAP^{50\%}$ su box e maschere

$$pt(metrica) = \begin{cases} \text{non ammesso} & \text{se } p < baseline_1 \\ 17 * \frac{p - baseline_1}{baseline_2 - baseline_1} & \text{se } baseline_1 \leq p \leq baseline_2 \\ 17 + 3 * \frac{p - baseline_2}{MAX(p) - baseline_2} & \text{se } p > baseline_2 \end{cases}$$

Il voto finale è la media dei punteggi su entrambe le metriche. p è lo score ottenuto sulla singola metrica

Protocollo di valutazione

```
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=3.23s).
Accumulating evaluation results...
DONE (t=0.40s).
```

Average Precision	(AP)	@[IoU=0.50:0.95	area= all	maxDets=100] = 0.239
Average Precision	(AP)	@[IoU=0.50	area= all	maxDets=100] = 0.460
Average Precision	(AP)	@[IoU=0.75	area= all	maxDets=100] = 0.238
Average Precision	(AP)	@[IoU=0.50:0.95	area= small	maxDets=100] = 0.177
Average Precision	(AP)	@[IoU=0.50:0.95	area=medium	maxDets=100] = 0.187
Average Precision	(AP)	@[IoU=0.50:0.95	area= large	maxDets=100] = 0.318
Average Recall	(AR)	@[IoU=0.50:0.95	area= all	maxDets= 1] = 0.221
Average Recall	(AR)	@[IoU=0.50:0.95	area= all	maxDets= 10] = 0.319
Average Recall	(AR)	@[IoU=0.50:0.95	area= all	maxDets=100] = 0.333
Average Recall	(AR)	@[IoU=0.50:0.95	area= small	maxDets=100] = 0.224
Average Recall	(AR)	@[IoU=0.50:0.95	area=medium	maxDets=100] = 0.269
Average Recall	(AR)	@[IoU=0.50:0.95	area= large	maxDets=100] = 0.491

Appelli

- 24 - gennaio - 9:00 - Studio Docente
 - Consegna entro «5gg prima dell'esame». Venerdì 19 ore 14
- 23 - febbraio - 9:00