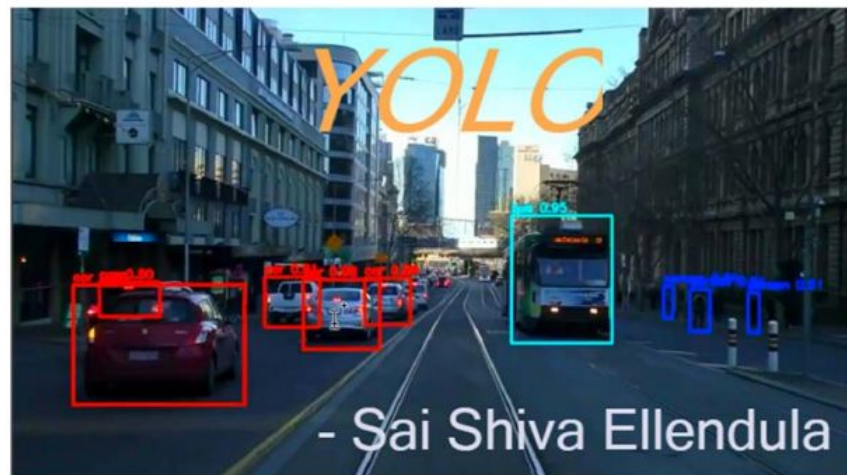


Relazione Programmazione sistemi mobili avanzati

4) Riconoscimento oggetti con YoloV3

Il progetto consiste nell'implementazione su dispositivo mobile di un sistema che utilizzi il famoso *detector* YoloV3 (<https://www.linkedin.com/pulse/object-detection-using-yolo-sai-shiva-ellendula/>) per il riconoscimento automatico di oggetti. YoloV3 è molto veloce ed accurato, ma il progetto dovrebbe:

- Verificare come implementarlo su progetto Android
- Verificarne le prestazioni su dispositivo mobile
- Realizzare un esempio applicativo (da concordare con il docente)



Tutor operativo: Luca Donati (luca.donati@unipr.it)

In allegato tutti i file sia di testing sia app Android.

Fase di testing (python opencv)

Dopo il fallimentare esperimento di luglio, ho implementato la fase di testing all'interno del progetto. In questa fase è stata anche fatta ottimizzazione dei parametri di hough e per i filtri di preprocessing.

E' stata sviluppata in python su libreria openCV. Per il effettuare il testing avviare il terminal e immettere:

```
> python test_park.py [nome_file]
```

I filtri che sono stati utilizzati sono :

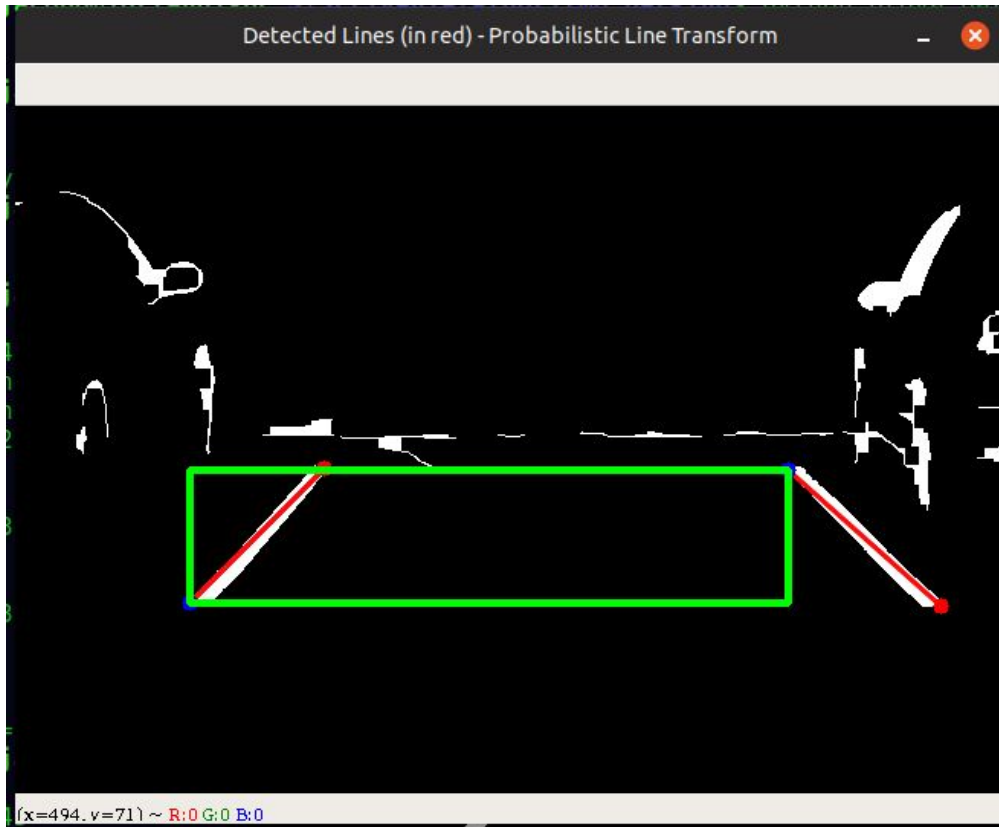
1. Filtro blur 5x5 con intensita di 0.25
2. Filtro Canny con parametri thresholds di 160,230 3x3
3. Chiusura morfologica (no Android) 9x9

PASM

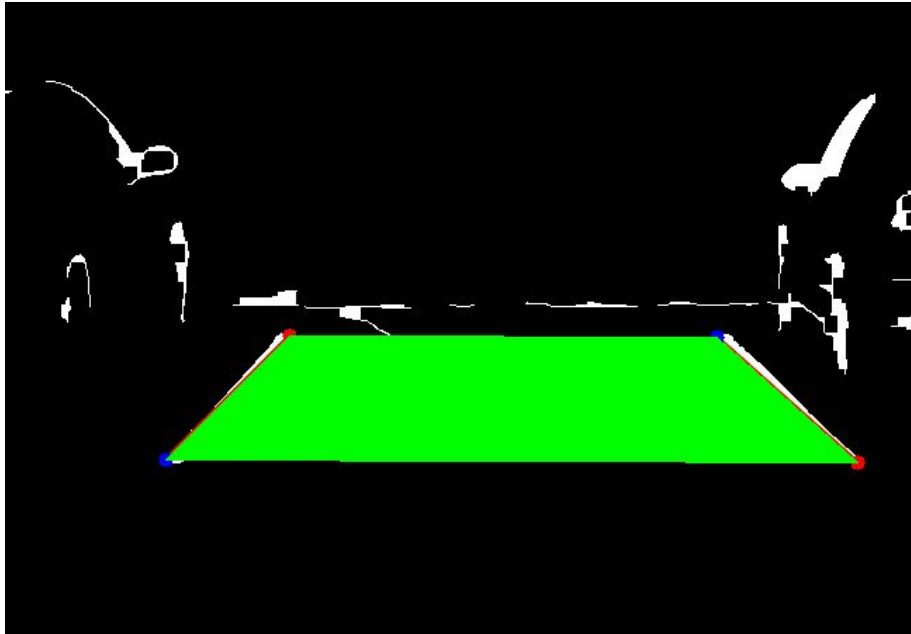
Legends:

- Green parking spot if available
- Red parking spot if not available (car is present inside the parking spot)
- Bounding box for Yolo V3

Di default ho inserito già un immagine.

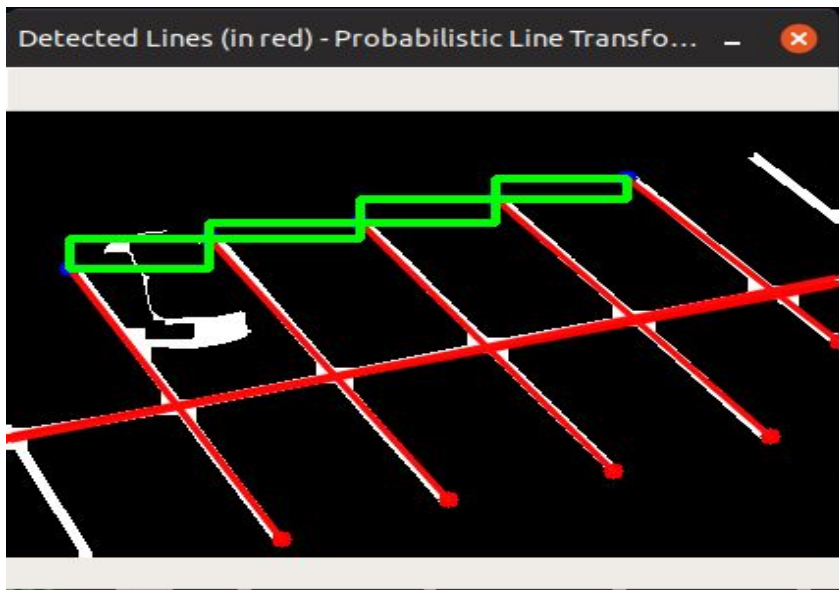


Nuova versione di script test_park.py

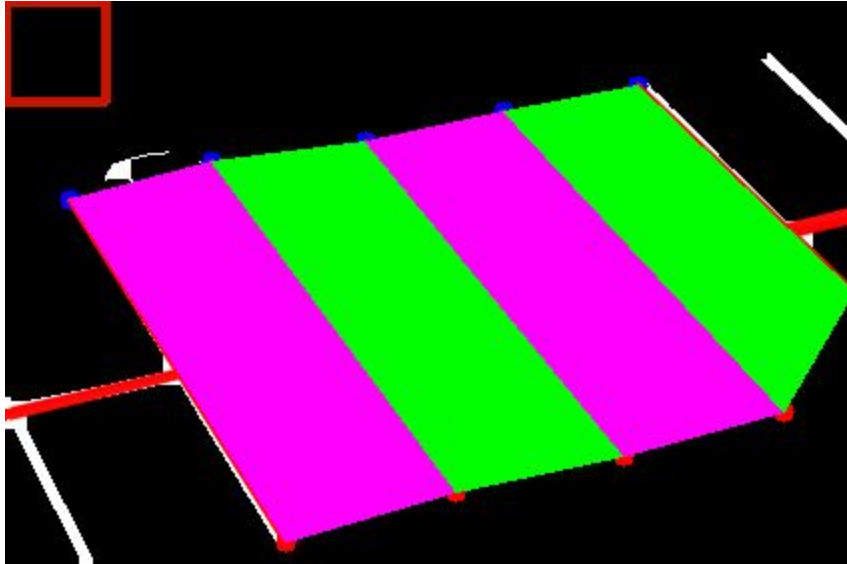


Steps for parking detection aerial view

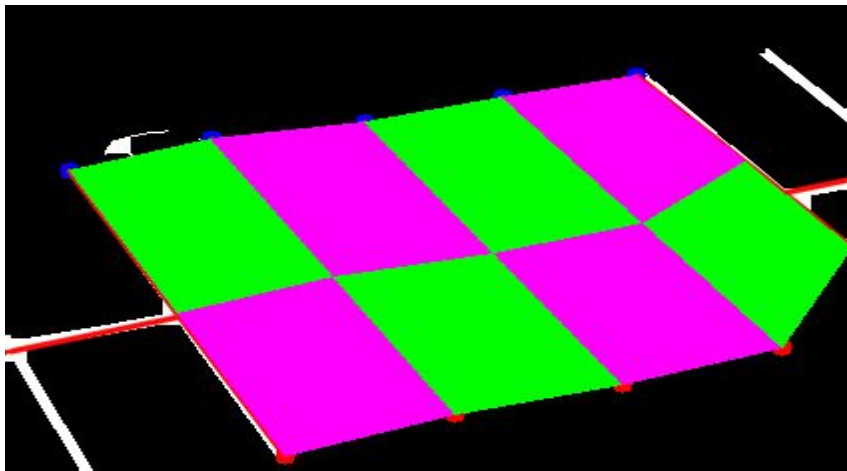
Step 1 first detection of probabilistic hough lines.



Second step. Covering and splitting vertical parking spot. (here green and purple have no meaning about parking status)

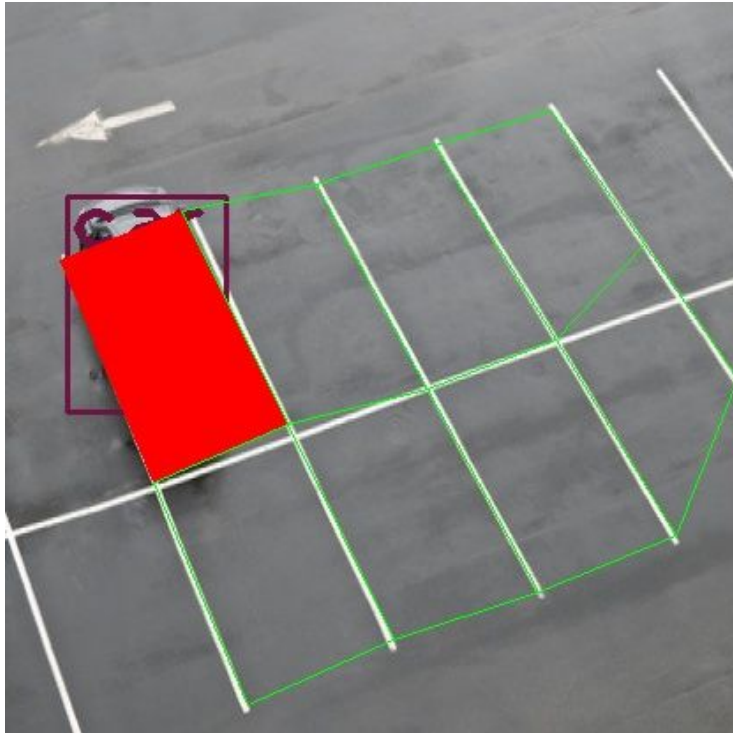


Third step: Splitting parking if multi parking.(here green and purple have no meaning about parking status)

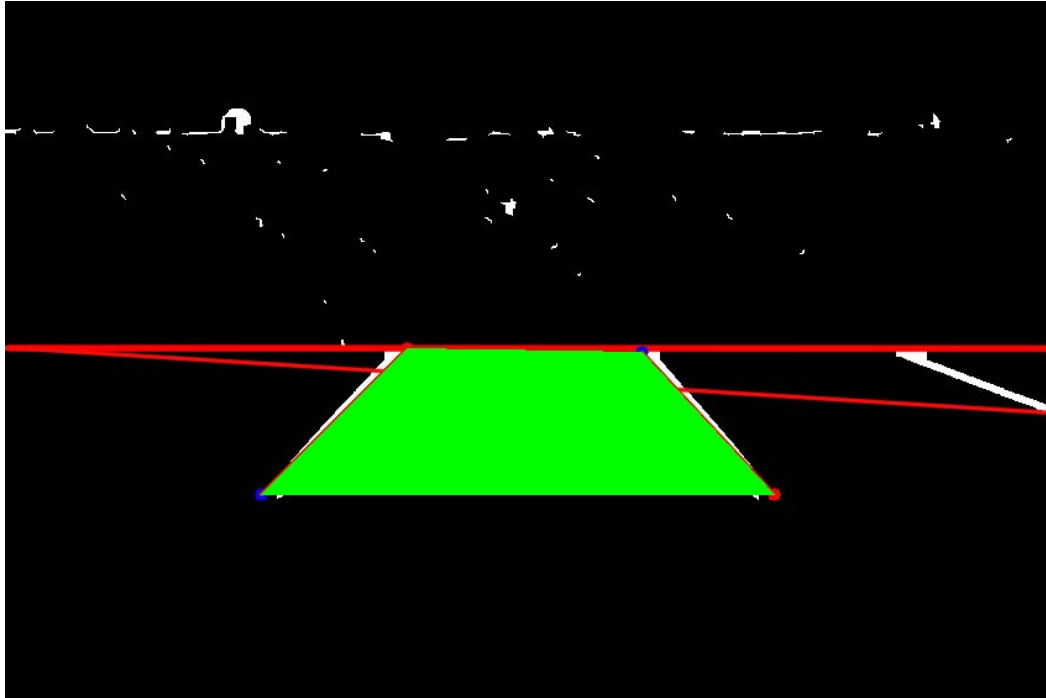


PASM

Final step: YoloV3 integration and detection of parking spot occupied and free.



Parking spot without any cars.

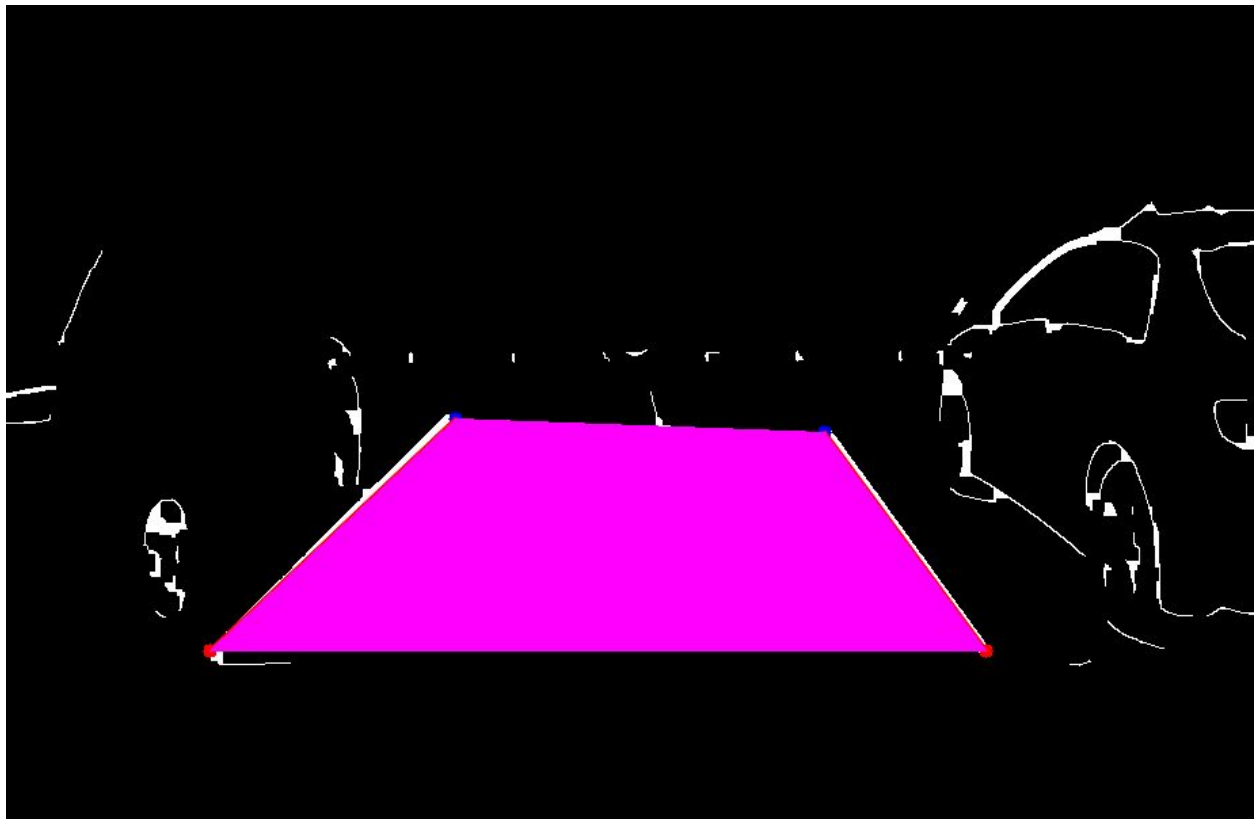


Yolo integration and detection free parkign spot



Detection parking spot with cars

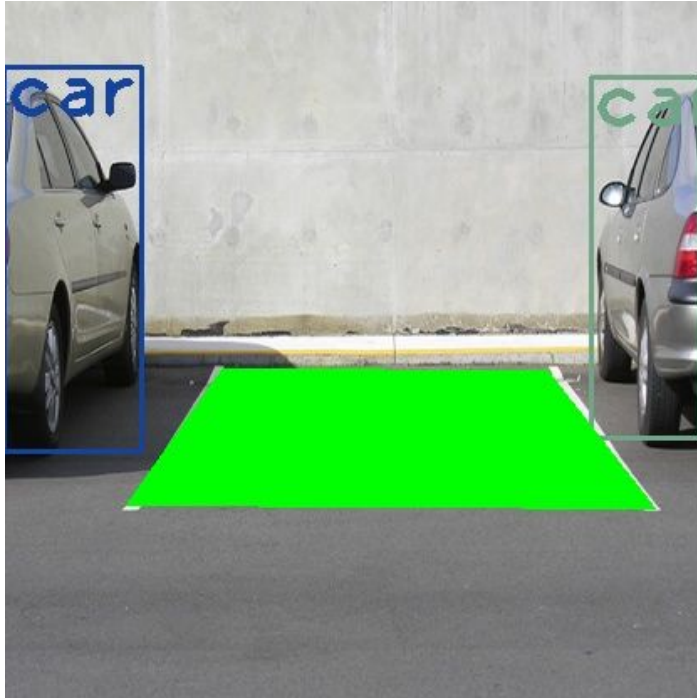
Simple Parking Spot Detection



Yolo Integration and detection if the parking spot is free.



Another little example



Fase di Progettazione

L'idea di fondo dell'applicazione è costruire un sistema per rilevare la presenza di parcheggi attraverso l'utilizzo di object Detection YOLO e sistema per il riconoscimento della segnaletica verticale per le righe dei posteggi attraverso openCV (Probabilistic Hough Transform).

Le difficoltà nella fase di sviluppo non avendo un'esperienza solida nel settore della visione artificiale e competenze Java ha portato ad un risultato grezzo e funzionante per limitati casi.

Si ha sfruttato la object detection di Yolo per la rilevazione delle auto e la conseguente presenza di posteggi all'interno di parcheggi con assenza di macchina. La seconda feature è stata l'implementazione della trasformata di Hough per il riconoscimento della segnaletica orizzontale (strisce del parcheggio bianche).

I casi rilevati di possibile utilizzo sono i seguenti:

1. Parcheggio singolo frontale con la presenza di due macchine ai lati.
2. Parcheggio singolo frontale con la presenza di una solo macchina al lato.
3. Parcheggio singolo frontale senza la presenza di macchine ai lati del parcheggio.
4. Parcheggio multiple frontale con la presenza di macchine e senza la presenza di macchine.

5. Parcheggio con sia macchine che furgoni all'interno della stessa immagine
6. Parcheggio su multi livello di profondità. (ovvero primo e secondo piano).
7. Parcheggi con visuale aerea (non solido).

Tutti questi casi risentono di rumori e di strane configurazioni che evitano il riconoscimento del parcheggio.

I casi in cui non è possibile utilizzare il modello presentato:

1. Situazioni con un forte offset in altezza.
2. Parcheggi Orizzontali o nel momento in cui il Bounding Box della macchina non sia approssimativamente orizzontale. Ovvero quando una macchina è vista in diagonale.
3. Situazioni con parcheggi senza segnaletica verticale per identificare il posteggio.
Invece nel caso in cui non ci fosse la segnaletica ma la presenza di 2 macchine ai lati si (come descritto nel punto 1).

Ho optato per la trasformata di Hough probabilistica in quanto non fornisce delle linee infinite ma oltre alla ricostruzione della linea anche i punti di inizio e fine della linea.

I parametri per la trasformata sono i seguenti:

```
linesP = cv.HoughLinesP(dst, 1, np.pi / 180, 100, None, minLineLength=80, maxLineGap=10)
```

Fase di sviluppo (su esistente repository github)

Il progetto ha ereditato lo scheletro da una versione funzionante di GitHub per il progetto di Yolo. Sono quindi state introdotte le ottimizzazioni ed i filtri adottati in python per un sistema in near real-time dalle buone prestazioni computazionali.

A causa di motivazioni personali ho venduto il tablet per fondi per nuovo cellulare quindi non sono riuscito a gestirlo. Allego una foto di repertorio ma la versione presente è notevolmente migliorata.