

Riconoscimento oggetti da flussi dati RGB e LIDAR

Gianluca Scarpellini

Università degli studi di Milano - Bicocca

19 luglio 2018

Introduzione

- ▶ Detection e riconoscimento di oggetti in frame a 3 e 4 canali
- ▶ Dataset Kitti Benchmark Suite
- ▶ Reti Neurali Convoluzionali
- ▶ Esperimenti RGB
- ▶ Lidar e immagini di depth
- ▶ Immagini di Depth + Immagini RGB = RGBD (a 4 canali)
- ▶ Detection e riconoscimento di oggetti in frame a 4 canali

Problema

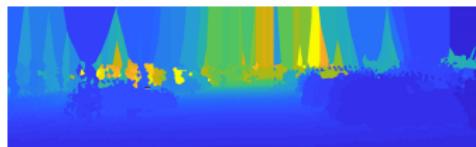
- ▶ Sensori a disposizione: Laser Scanner, Camera RGB
- ▶ Task: identificare pedoni, ciclisti e automobili all'interno della scena
- ▶ Per ogni oggetto identificato, misura della distanza dal veicolo



Figura 1: Risultato della detection su frame RGBD tratto da The Kitti Vision Benchmark Suite)

The Kitti Vision Benchmark Suite

- ▶ Dataset di benchmark per algoritmi di detection
- ▶ Immagini RGB e dati LIDAR acquisiti con diverse condizioni ambientali
- ▶ 7480 frame totali: 3712 di training + 3768 di test
- ▶ 3 classi di oggetti: person, cyclist e car
- ▶ 3 livelli di difficoltà definiti rispetto all'occlusione, al troncamento e alla dimensione dell'oggetto



(a) Depth



(b) RGB

Figura 2: Esempio di immagine di Depth generata e la relativa immagine RGB (The Kitti Vision Benchmark Suite)

Reti Neurali Convoluzionali

- ▶ Reti finalizzate all'individuazione e alla classificazione di oggetti all'interno di un'immagine
- ▶ Input sia esplicitamente un'immagine (eventualmente a più canali)
- ▶ Strutturate come sequenza di layer con funzioni specifiche (convoluzionali, pooling, fully-connected)

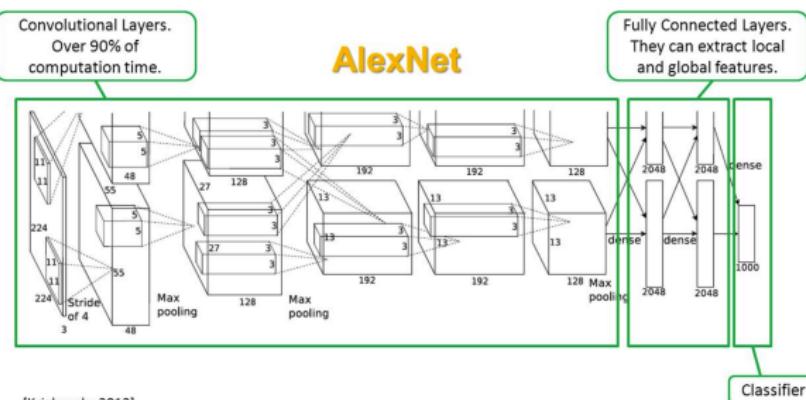


Figura 3: AlexNet, una delle prime CNN (by slideplayer.com)

YOLO

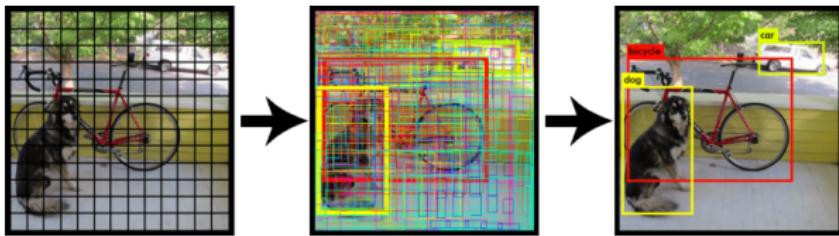


Figura 4: Esempio pipeline YOLO

- ▶ Immagine suddivisa in celle (7x7)
- ▶ Regressione con l'ausilio di anchors predefinite per ogni cella
- ▶ 5 valori predetti per anchor: x, y, w, h, confidence
- ▶ Classe predetta per ciascuna cella
- ▶ nms + threshold sulla confidenza + best IOU per estrarre le bounding box corrette e le relative classi

Metriche

Il valore di Average Precision (AP) si ottiene in due passaggi:

- ▶ Si definisce la curva di precision / recall $p/r(r)$ ponendo per ogni valore di recall r il massimo valore di precision ottenuto per ogni valore di $r' \geq r$

$$p/r(r) = \max(p(r') : r' \geq r)$$

- ▶ Si calcola la metrica AP come l'area sottostante la curva mediante integrazione numerica

La metrica Mean Average Precision (mAP) si ottiene come media aritmetica dei valori di AP calcolati per ogni classe

AP	Easy %	Moderate %	Hard %
Car	43.84	41.11	29.72
Cyclist	20.64	3.66	3.41
Person	37.75	27.01	24.65

Tabella 1: Yolo addestrata sul Kitti Benchmark Suite D usando pesi preaddestrati del EXP5 per 80 epocha

Dati Lidar Time-Of-Flight

- ▶ Un emettitore emette un flusso di beam laser contro le superfici circostanti
- ▶ Un sensore cattura i beam riflessi dagli oggetti

$$d = \frac{c}{\Delta t/2}, \text{ con } c = \text{velocità della luce}$$

- ▶ Risultato: struttura dati matriciale nx3, dove n numero di punti acquisiti e 4 numero di colonne colonne (X, Y, Z e luminanza)

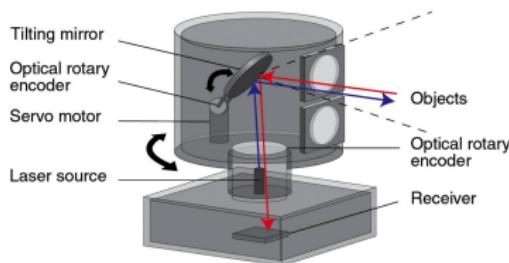


Figura 5: Funzionamento Laser Scanner (by elettroamici.org)

Da Lidar a Depth

Si compone innanzitutto una matrice 4×4 di roto-traslazione che leggi i frame di camera e veldoyne.

$$T_{\text{velo}}^{\text{cam}} = \begin{bmatrix} R_{\text{velo}}^{\text{cam}} & t_{\text{velo}}^{\text{cam}} \\ 0 & 1 \end{bmatrix}$$

Un punto lidar 3D x può essere proiettato sull'immagine della telecamera i tramite la matrice di proiezione:

$$P = P_{\text{rect}}^{L_RGB} R_{\text{rect}} T_{\text{velo}}^{\text{cam}}$$

Dopo aver escluso i punti esterni allo spazio di ripresa della telecamera RGB, si genera l'immagine risultato. Per il canale di profondità è stato utilizzato il canale x dei dati LIDAR grezzi.

Depth

	Easy %	Moderate %	Hard %
Car	43.84	41.11	29.72
Cyclist	20.64	3.66	3.41
Person	37.75	27.01	24.65

Tabella 2: Yolo addestrata sul Kitti Benchmark Suite D usando pesi preaddestrati del EXP5 per 80 epochhe

	Easy %	Moderate %	Hard %
Car	39.44	37.09	26.39
Cyclist	17.99	4.57	3.39
Person	35.93	26.01	23.72

Tabella 3: Yolo addestrata sul Kitti Benchmark Suite RGBD usando pesi preaddestrati del EXP5 per 140 epochhe

Conclusione

- ▶ Classi di difficoltà coerenti
- ▶ Classe bici risulta la più difficile di individuare e riconoscere
- ▶ Possibilità di impiegare modello RGB su video acquisiti con strumenti non professionali

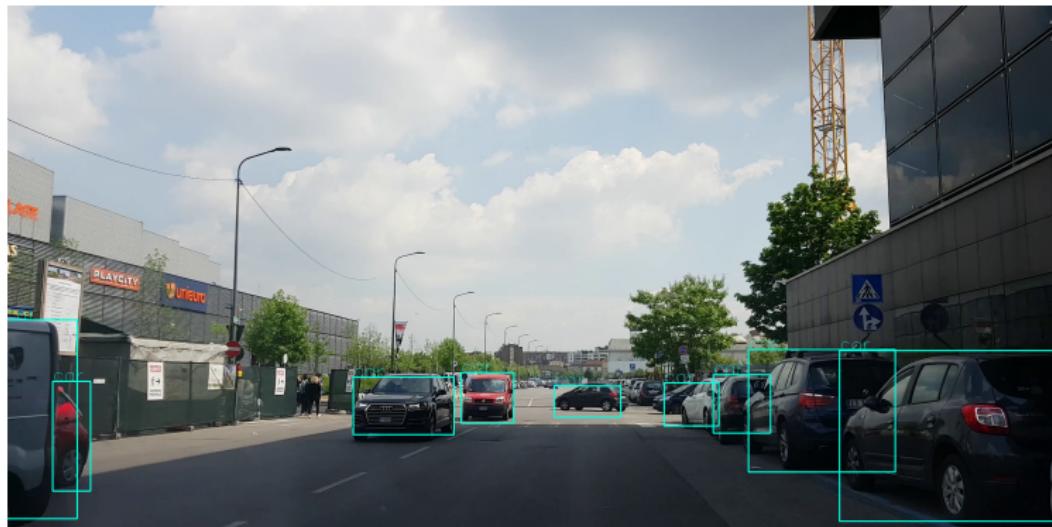


Figura 6: Viale Sarca, Milano (2018)