

# Rilevamento di oggetti e calcolo della distanza con sensore Kinect e algoritmo YOLO.

## Introduzione

Il progetto è incentrato sull'integrazione di un sensore Kinect con un sistema di computer vision che sfrutta l'algoritmo di rilevamento YOLOv3 per rilevare volti in tempo reale e stimare la loro distanza rispetto al sensore.

Questo sistema sfrutta la mappa di profondità fornita dal sensore per ottenere informazioni da unire alle capacità dell'algoritmo.

L'obiettivo principale è quello di sviluppare un sistema di rilevamento dei volti e di misurazione della distanza in tempo reale, con applicazioni possibili in ambiti come la robotica, la guida autonome e la sorveglianza.

## Librerie, algoritmi e framework utilizzati

- **YOLOv3**: Un modello di deep learning per il rilevamento rapido di oggetti, addestrato su un dataset personalizzato
- **MEGAFACE**: Dataset contenente le immagini di volti/persone per il training
- **Darknet**: Framework per il training di YOLO.
- **freenect**: Libreria per interfacciarsi con il Kinect e accedere ai dati rgb e di profondità.
- **OpenCV**: Utilizzato per il pre-processing delle immagini e per il framework DNN che gestisce YOLO e per la rappresentazione dell'output.
- **NumPy**: Per la gestione di matrici e operazioni numeriche.

## Struttura del sistema

Il sistema è composto da diverse fasi, ognuna delle quali svolge una funzione specifica nella percezione.

### ➤ Calibrazione

Il Kinect è inizialmente calibrato per regolare l'angolo di inclinazione, che consente al sensore di acquisire immagini in un campo visivo ottimale. Questo processo avviene utilizzando il modulo freenect, che fornisce funzioni per controllare l'angolo del sensore.

Nel codice, la calibrazione è eseguita a  $27^\circ$  e  $-27^\circ$ , infine viene riportata a  $0^\circ$  per posizionare il sensore orizzontalmente.

### ➤ Acquisizione delle Immagini e della Mappa di Profondità

In questa fase, il sensore inizia a catturare due tipi di dati:

- **Flusso video RGB**: L'immagine a colori dell'ambiente circostante.
- **Mappa di profondità**: La distanza tra il sensore e gli oggetti rilevati, espressa in valori di profondità.

Entrambi i dati vengono acquisiti in parallelo, in particolare, l'immagine RGB viene convertita da BGR (output del Kinect) a RGB per essere utilizzabile da OpenCV, mentre la mappa di profondità viene memorizzata per stimare le distanze.

### ➤ **Rilevamento degli Oggetti con YOLO**

Una volta acquisita l'immagine, questa viene preparata per essere processata dal modello YOLO. Il modello YOLOv3 è caricato utilizzando il framework DNN di OpenCV, che permette di eseguire il rilevamento su una vasta gamma di oggetti. Nel codice:

- Viene caricata la configurazione di yolo e i pesi addestrati con darknet "yolov3.cfg", "yolov3.weights".
- Le classi degli oggetti vengono lette dal file "coco.names".

L'immagine RGB convertita viene trasformata in un blob, ovvero una rappresentazione che normalizza l'immagine per essere compatibile con il modello YOLO.

L'algoritmo successivamente elabora l'immagine e restituisce le bounding box degli oggetti rilevati, insieme alla loro classe e al livello di confidenza.

Se il livello di confidenza è sufficientemente alto,  $\geq 0.5$  e la classe dell'oggetto è quella di interesse allora salva la posizione della bounding box.

### ➤ **Stima della Distanza degli Oggetti**

Parallelamente al rilevamento degli oggetti, la mappa di profondità del Kinect viene utilizzata per stimare la distanza tra il sensore e gli oggetti rilevati da YOLO.

Ogni oggetto di interesse rilevato ha associata una regione nella mappa di profondità, dalla quale è possibile estrarre un valore medio di distanza.

Questo valore rappresenta la distanza approssimativa tra l'oggetto e il sensore, consentendo al sistema di fornire non solo la posizione degli oggetti nel piano 2D, ma anche la loro distanza dal sensore.

### ➤ **Visualizzazione e Feedback**

Gli oggetti rilevati vengono visualizzati sull'immagine RGB con un bounding box e la loro classe (Human Face), la distanza calcolata viene sovrapposta alla visualizzazione, permettendo all'utente di osservare in tempo reale sia la classificazione che la distanza dal sensore.

## **Conclusioni**

Questo progetto integra con successo l'utilizzo di un sensore Kinect e l'algoritmo di YOLO per creare un sistema di rilevamento degli oggetti e stima della distanza in tempo reale. La combinazione di visione RGB e profondità rende il sistema applicabile in una varietà di contesti. Il codice può essere facilmente esteso per includere ulteriori funzionalità come il rilevamento di altri oggetti o l'integrazione di altri sensori.