

Cognitive Robotics

Object Recognition with Pepper

Gruppo 6

Cavallaro Virginia, Coppola Davide, Napolano Carmine, Pandolfi Giuliano

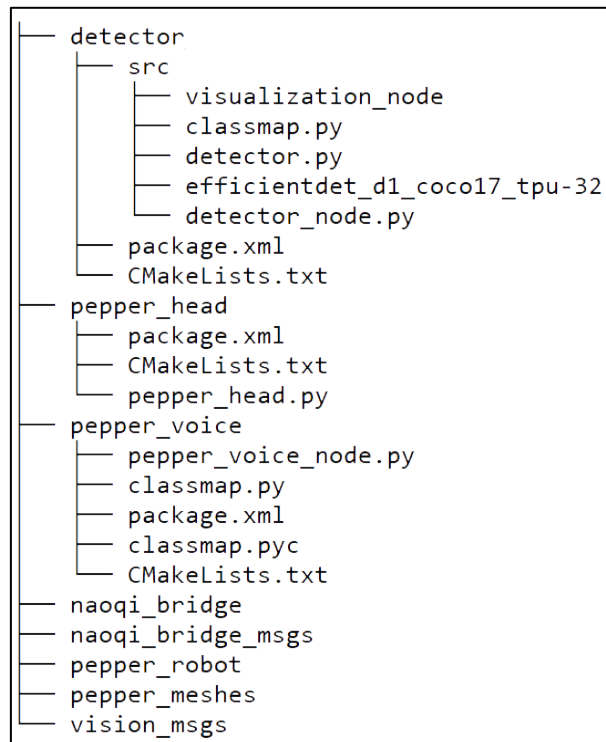
{v.cavallaro@studenti.unisa.it,
d.coppola25@studenti.unisa.it,
c.napolano@studenti.unisa.it,
g.pandolfi5@studenti.unisa.it}

1 Introduzione

Lo scopo del progetto è riuscire a rendere Pepper capace di analizzare e descrivere la scena davanti a sé elencando gli oggetti rilevati nelle tre direzioni principali: destra, centro e sinistra.

2 Descrizione Architettura realizzata

2.1 Workspace



Architettura dello spazio di lavoro

3 Descrizione Architettura e scelte di Design

Il meccanismo di comunicazione tra nodi vede l'impiego della logica *publisher/subscriber*.

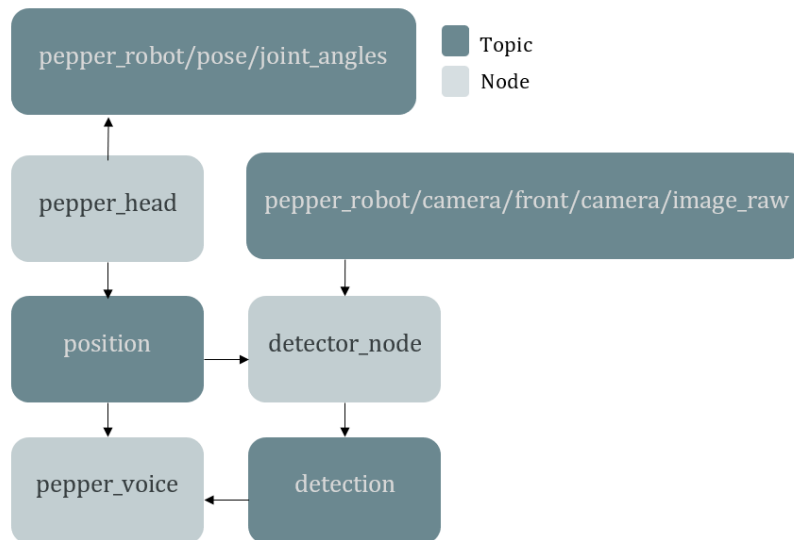
La nostra architettura prevede i seguenti nodi:

- **pepper_robot/camera/front/camera**
Nodo che permette l'acquisizione di immagini dalla camera frontale presente in Pepper. È collezionata un'unica immagine per ciascuna posizione, la quale sarà elaborata dal detector per la rilevazione degli oggetti nella scena.
- **detector_node**
Nodo responsabile della rilevazione e della classificazione degli oggetti nella scena tramite l'utilizzo di un Object Detector.
È sottoscritto a *pepper_robot/camera/front/camera/image_raw*, topic sul quale vengono pubblicate le immagini acquisite dalla camera frontale di Pepper.
Inoltre, è sottoscritto a *position*, topic su cui è pubblicata la posizione della testa di Pepper, così da effettuare la lettura dell'immagine della camera frontale in maniera sincronizzata ai movimenti del robot.
Tale nodo pubblica su *detection* una lista di bounding box con confidence maggiore del 50% e le relative etichette.
- **pepper_voice_node**
Nodo utilizzato per la fase di *speech*, durante la quale Pepper elenca gli oggetti rilevati specificandone la posizione.
È sottoscritto a *position*, in modo da conoscere la posizione in cui ha rilevato gli oggetti.
Inoltre, è sottoscritto a *detection*, da cui recupera la lista di oggetti rilevati nella scena e le etichette che vengono mappate come descritto nel file *classmap.pyc*.
- **pepper_head**
Nodo impiegato per il coordinamento dei movimenti della testa di Pepper nelle tre diverse posizioni, in modo da coprire tutta la visuale frontale.

È possibile raggiungere una posizione specifica o effettuare un movimento completo secondo un parametro specificato all'avvio del nodo.

Il nodo pubblica sul topic *pepper_robot/pose/joint_angles* gli angoli di posizionamento della testa compresi in un range $[-1,1]$.

Il nodo pubblica, inoltre, su *position* la posizione attuale della testa di Pepper.



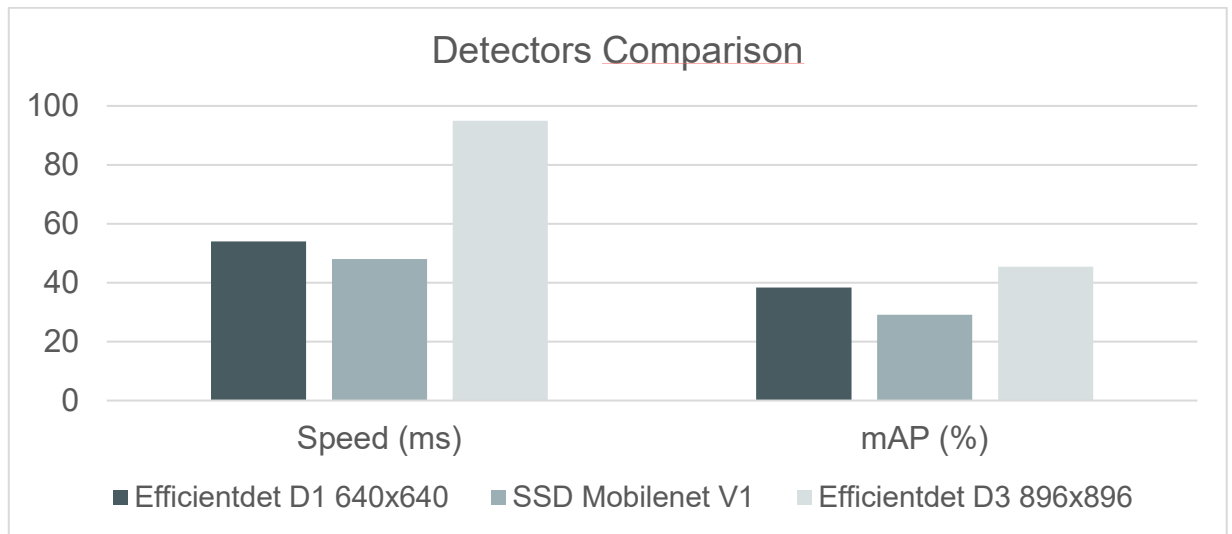
Mappa delle relazioni tra i nodi e i topic

Dall'immagine, di cui sopra, è possibile visualizzare i rapporti tra nodi e topic, specificati da delle frecce:

- le frecce *entranti* in un nodo, si riferiscono ai topic a cui il nodo è sottoscritto.
- le frecce *uscenti* da un nodo, si riferiscono ai topic su cui il nodo pubblica.

4 Detector

Diversi sono stati i detectors testati durante la realizzazione del progetto. Con l'ausilio dell'istogramma possiamo osservare gli indici delle prestazioni per alcuni di questi:



La scelta finale è ricaduta su *EfficientDet D1*, il quale ci garantisce di avere un valido trade-off in termini di velocità e percentuale di corretta precisione rispetto agli altri modelli. Tale scelta, secondo letteratura scientifica, è parte di una famiglia di modelli che risulta essere scalabile ed efficiente per problemi di *object detection*. L'input size scelto è 640x640 per motivi che tengono in considerazione la possibilità di utilizzo del sistema in real-time, oltre che aspetti computazionali.

5 Conclusioni