

Università degli Studi di Salerno

Dipartimento di Informatica



Tesi di Laurea Triennale in
Informatica

Studio del Robot umanoide Pepper: Modello Prensile e Object Recognition

Relatore

Prof. Andrea Francesco Abate

Candidato

Antonio De Luca

Matr. 0512104900

Correlatore

Dott.ssa Carmen Bisogni

Dott.ssa Lucia Cascone

Anno accademico 2019/2020

Ai miei genitori, per il sostegno e la fiducia.

Ad Anna, per il nostro strano modo di amarci.

A me stesso, per la mia determinazione.

Indice

1 Introduzione	2
1.1 Scopo della tesi.....	3
1.2 Organizzazione del lavoro	4
2 I robot e il loro utilizzo	5
2.1 Introduzione.....	5
2.2 Utilizzo dei sistemi robotici.....	7
2.2.1 Robotica industriale.....	7
2.2.2 Robotica di servizio	8
2.2.3 Statistiche di mercato	9
3 Robot umanoide Pepper	11
3.1 Funzionalità e caratteristiche	11
3.1.1 Telecamere 2D.....	13
3.1.2 Telecamera 3D.....	13
3.2 Choregraphe	14
3.2.1 Barra degli strumenti	16
3.2.2 Moduli di programmazione	17
4 Object Recognition	21
4.1 Che cos'è l'Object Recognition.....	21
4.2 Tecniche di Object Recognition	22
4.2.1 Object Recognition mediante il Machine Learning.....	22
4.2.2 Object Recognition mediante il Deep Learning	23

4.3	Fasi dell' Object Recognition.....	24
4.3.1	Fase di memorizzazione	24
4.3.2	Fase di apprendimento.....	25
4.3.3	Fase di ricerca.....	26
4.4	Pepper e l'Object Recognition.....	26
5	Implementazione del caso d'uso	28
5.1	Idea generale.....	28
5.2	Fasi di realizzazione	29
5.2.1	Apprendimento dell'oggetto.....	29
5.2.2	Creazione del database	31
5.2.3	Riconoscimento dell'oggetto.....	31
5.2.4	Movimento e parte prensile	32
5.3	Implementazione	34
5.4	Testing e problemi riscontrati.....	36
6	Conclusioni	37
Bibliografia		

Elenco delle figure

1.1	Robot Pepper SoftBank Robotic	3
2.1	Automa cavaliere di Leonardo da Vinci	7
2.2	Robot industriali	8
2.3	Robot di servizio domestico.....	9
3.1	Caratteristiche e dimensioni di Pepper.....	12
3.2	Telecamere 2D poste sulla testa di Pepper	13
3.3	Telecamere 3D poste nell'occhio sinistro di Pepper	14
3.4	Software applicativo multiplatforma Choregraphe	15
3.5	Barra degli strumenti di Choregraphe	16
3.6	Esempio di programmazione a blocchi	18
3.7	Modulo di programmazione	18
4.1	Memorizzazione oggetti.....	18
4.2	Apprendimento oggetto.....	24
4.3	Modulo di Object Recognition.....	25
4.4	Video monitor	26
5.1	Learn button	28
5.2	Estremità oggetto	29
5.3	Box informazioni	29
5.4	Fase di riconoscimento.....	31
5.5	Modulo Move To	32
5.6	Timeline box	32
5.7	Implementazione del caso d'uso	33

Abstract

In questa tesi, viene esplorato il problema del rilevamento degli oggetti dal robot umanoide Pepper e la sua implementazione. Il rilevamento degli oggetti è una parte fondamentale dell'interazione uomo-robot e i progressi nel rilevamento di essi miglioreranno sostanzialmente in futuro il livello di interazione che può essere raggiunto. Lo scopo di questa tesi è stato quello di far prendere un oggetto a Pepper. Per fare ciò sono stati utilizzati diversi sensori e sono state sfruttate diverse tecniche. Sono state infatti esplorate diverse tecniche di rilevamento che utilizzano i vari sensori disponibili per Pepper tra i quali due telecamere 2D in HD montate sulla fronte e nella bocca e tattili che consentono di chiudere la mano.

Capitolo 1

Introduzione

Da sempre il progresso, soprattutto quello tecnico e oggi quello tecnologico, ha svolto un ruolo cruciale nell'evoluzione della conoscenza umana. Esso ha fatto parte di tutto il Novecento con un ruolo non marginale, basti pensare alle scoperte dell'epoca giunte ai giorni nostri con significanti miglioramenti che verranno perfezionati anche in futuro. Il progresso tecnologico ha innegabilmente migliorato la nostra vita con una serie di scoperte e invenzioni che hanno permesso all'uomo di guardare oltre i propri orizzonti e persino oltre la propria immaginazione. In passato l'intelligenza artificiale è stata spesso associata alla robotica nella cultura e nella letteratura popolare; l'idea di robot che aiutano le persone nella loro vita quotidiana, capaci di effettuare e svolgere lavori almeno al pari dell'uomo, tuttavia, è ancora forte come prima. Ad oggi, ciò che un tempo era pura immaginazione per noi è diventata una stupefacente quotidianità, consolidata e testata, e ciò che per noi è tutt'ora fantascienza un giorno sicuramente diventerà realtà, con il progredirsi delle varie tecnologie. Quando gli umani guardano una fotografia o un video, possono facilmente individuare persone, oggetti, scene e dettagli visivi; vale lo stesso quando si parla di afferrare una bottiglietta d'acqua o impugnare uno spazzolino. Tutto ciò non è un compito facile e naturale per le macchine come per noi, qui entra in gioco l'object recognition, una tecnica di visione artificiale, applicata in questo lavoro al robot Pepper, che permette di identificare oggetti in immagini o video tramite le sue telecamere ed attraverso questa tecnica riuscire ad afferrare l'oggetto riconosciuto.

1.1 Scopo della tesi

Questa ricerca esplorerà il modo in cui il rilevamento degli oggetti può essere risolto sul robot umanoide Pepper della casa produttrice SoftBank Robotics (Figura 1.1), utilizzando tecniche di riconoscimento delle immagini all'avanguardia su immagini 2D e 3D. Più formalmente, l'obiettivo di questo progetto è quindi quello di:

Sviluppare un metodo di rilevamento degli oggetti usando i diversi sensori disponibili per Pepper e trovare il modo migliore di combinare i diversi metodi di rilevamento per afferrare l'oggetto.

Al fine di raggiungere l'obiettivo della ricerca, è necessario rispondere alle seguenti domande: come si possono utilizzare i sensori di Pepper per rilevare gli oggetti? Che può essere suddiviso nelle seguenti domande secondarie:

1. Quali sensori sono rilevanti per il rilevamento degli oggetti? 2. Quale metodo di rilevamento è adatto per i sensori?



Figura 1.1. Robot Pepper SoftBank Robotic

1.2 Organizzazione del lavoro

Le parti successive della tesi sono organizzate come segue: il secondo capitolo presenta una panoramica sulla robotica e sulla sua storia fino a giungere ai giorni nostri; nel dettaglio sono descritte le prime invenzioni attribuite a robot umanoidi e in seguito sono mostrate le principali caratteristiche dei robot e la loro classificazione.

Nel capitolo successivo viene descritto il robot umanoide Pepper e il suo funzionamento, più nel dettaglio si parlerà dell'ambiente di sviluppo di Pepper e dei suoi moduli.

Nel quarto capitolo ci si soffermerà sulla tecnologia dell'object recognition con le sue fasi più importanti, nonché i moduli di Pepper dedicati ad essa.

Infine, nell'ultimo capitolo è descritto il progetto di tesi, con i vari esperimenti e i relativi risultati.

Capitolo 2

I Robot e il loro utilizzo

In questo capitolo viene affrontata e descritta inizialmente la robotica in generale, successivamente ci si soffermerà in particolare sui robot umanoidi e sul loro utilizzo. L'interazione uomo macchina svolge un ruolo cruciale, infatti sono da non sottovalutare le conseguenze che derivano dalla percezione di persone ed oggetti da parte dei robot. Di seguito, sono riportate alcune applicazioni nate grazie a questo settore della robotica ed eventuali sviluppi in questo campo.

2.1 Introduzione

Il termine robot ha origine moderne e deriva dalla parola "*robota*" della lingua ceca, che significa "lavoro pesante". Lo scrittore ceco Karel Čapek la utilizzò per la prima volta nella sua opera R.U.R. del 1920. Un robot umanoide è sostanzialmente una macchina intelligente la cui struttura riproduce il corpo umano. Come noi, questi robot hanno una testa, un torso, braccia, mani e spesso anche gambe, ma anche sistemi visivi e uditivi. La robotica umanoide ha il non facile obiettivo di riprodurre con queste macchine gli esseri umani, le nostre abilità fisiche, i nostri processi cognitivi, la nostra capacità di rispondere agli stimoli ambientali e di adattarci all'ambiente nel quale ci troviamo. Uno dei primi robot umanoidi è stato "Eric the robot" ideato nel 1928 dall'ingegnere Alan Herbert Reffell e dal capitano William H. Richards, costruito in alluminio e dotato di un alimentatore a 12V, era in grado di muovere testa e mani. Nel decennio successivo, nel 1938, è stato costruito "Elektro", un robot di proprietà della Westinghouse Electric Corporation capace di pronunciare centinaia di parole, muovere le braccia, camminare e distinguere il verde dal rosso. Nel

1962 in Russia è stato ideato e creato Sepulka, un robot guida per un museo che fino al 2010 è stato in grado di fornire indicazioni ai visitatori. Dopo qualche anno, all'interno del centro di ricerca di Stanford è stato progettato Shakey, il primo robot polivalente in grado di ragionare sulle proprie azioni. Wabot-1, il primo robot moderno, è stato progettato all'università di Waseda (Tokyo) ed è in grado di comunicare, misurare distanze e manipolare oggetti grazie ai sensori tattili. La seconda versione del robot, Wabot-2, è stata costruita nel 1984 ed è nato come robot musicista in grado di leggere e suonare musica. Un paio di anni dopo, nel 1986, Honda ha creato alcuni robot sperimentali, la serie E, seguiti dai modelli P1 e P3. Il miglioramento di questi modelli ha dato poi origine al robot più famoso di Honda: ASIMO.

Contrariamente a quanto si possa solitamente pensare, le prime forme di robot sono molto antiche e si possono far risalire agli automi meccanici, lo stato primordiale dei robot umanoidi che conosciamo oggi. La **Colomba di Archita** che risale al IV secolo a.C. è un esempio, c'è una testimonianza dell'antica Cina in cui si descrive che fu mostrato all'imperatore un pupazzo meccanico con la forma e le dimensioni di un umano. Nell'anno 332 a.C., Aristotele afferma nel suo testo "*Politica*" che un giorno gli automi faranno scomparire la schiavitù. Si può quindi ritrovare un concetto primordiale di robotica di servizio, anche se in forma molto meno evoluta, già in epoche passate. Un altro dei precursori della robotica, nella forma degli automi, è Erone di Alessandria che ha contribuito in maniera significativa al loro sviluppo. Successivamente, nel XII secolo d.C. Al-Jazari, inventore e ingegnere meccanico arabo, ha progettato e creato alcune macchine automatiche, tra cui anche degli utensili da cucina ed il primo automa musicale programmabile. Quest'ultimo era costituito da un semplice meccanismo con una ruota programmabile che può attivare, tramite dei denti, le leve collegate che, a loro volta, producono dei suoni o dei movimenti di alcune statuette animate. Altro importante contributo è stato apportato anche da Leonardo da Vinci, che ha creato attorno al 1495 uno dei primi progetti di automa umanoide: l'automa cavaliere (Figura 2.1). La sua creazione consiste in un'armatura meccanica in grado di muovere le braccia, la testa e la mascella e di alzarsi in piedi.

Colomba d' Archita: colomba di legno, vuota all'interno, riempita d'aria compressa e fornita da una valvola che permetteva l' apertura e chiusura.

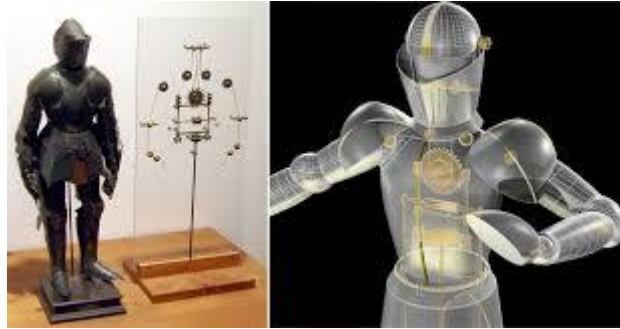


Figura 2.1. Automa cavaliere di Leonardo da Vinci

2.2 Utilizzo dei sistemi robotici

Gli ultimi rapporti recenti sulla robotica secondo le Nazioni Unite [U.N.] e la *International Federation of Robotics* [I.F.R.], evidenziano che essa è raggruppata in due categorie principali: quella industriale e quella orientata a servizi (professionali o personali). Tra i diversi fattori che hanno favorito lo sviluppo della robotica ci sono: la riduzione dei costi di dispositivi di calcolo sempre più potenti, la produzione di strumenti per la rilevazione delle caratteristiche ambientali e di attuatori sempre più robusti e precisi. Allo stato attuale, la maggior parte dei robot operano in ambienti industriali dove eseguono principalmente compiti di assemblaggio o trasporto di materiali. Tutti questi progressi hanno reso possibile lo sviluppo di una nuova generazione di robot di servizio progettati per assistere utenti umani direttamente nel loro posto di lavoro, nel tempo libero e a casa.

2.2.1 Robotica industriale

Il campo della robotica è sempre più collegato alle tecnologie avanzate, quali l'integrazione uomo macchina o sistemi mobili automatici, l'indottrinamento automatico, l'Artificial Intelligence (AI) nonché l'Industrial Internet of Things (IIOT). La parte determinante della

digitalizzazione dell'industria manifatturiera, sono i robot intelligenti. L'industria manifatturiera affronta quotidianamente sfide enormi per la propria sopravvivenza dovute a molteplici fattori quali le tendenze dei propri consumatori in velocissima evoluzione, la carenza di risorse e di manodopera qualificata, le disparate domande di produzioni locali, nonché l'invecchiamento della società. Cruciale quindi diventa l'apporto dei robot intelligenti per la digitalizzazione delle industrie manifatturiere. L'automazione flessibile incentrata su robot industriali, propone la soluzione a tutte queste sfide. Nella robotica industriale, un sistema di automazione dunque, sostituisce un uomo nella catena di montaggio, compiendo sempre lo stesso lavoro ad un ritmo costante e frenetico (Figura 2.2). Tutti gli strumenti meccanici progettati per compiere un determinato lavoro in autonomia rientrano a far parte della robotica industriale.



Figura 2.2. Robot industriali

2.2.2 Robotica di servizio

Detta anche *service robotics*, la robotica di servizio è un innovativo campo della robotica che si occupa di applicare le conoscenze e le tecniche della robotica con lo scopo di fornire un servizio all'uomo oppure sostituirlo nello svolgimento di alcuni compiti per soddisfare alcune sue necessità. Ad oggi, i robot li troviamo ovunque: si adoperano nell'ambito della domotica in faccende domestiche ritenute laboriose e ripetitive e che si possono

facilmente automatizzare. Più formalmente i robot di servizio sono definiti dall'organizzazione ISO (International Organization for Standardization) come “Robot that performs useful tasks for humans or equipment excluding industrial automation applications”, cioè robot che eseguono attività utili agli umani o ad apparecchiature (Figura 2.3), escludendo le applicazioni dell'automazione industriale. Quindi, si escludono tutte quelle applicazioni robotiche più tradizionali che servono esclusivamente ad automatizzare un processo produttivo, ma si considerano invece quei robot che possono collaborare o svolgere un servizio per l'uomo. Sono compresi nella definizione anche quei robot che hanno un'autonomia parziale, ad esempio quelli che operano in remoto, oppure sono completamente autonomi, ovvero non richiedono un'attiva interazione umana.



2.3. Robot di servizio domestico

2.2.3 Statistiche di mercato

I numeri della robotica nel mondo e in Italia continuano a crescere in modo vertiginoso. È questo il sunto che è possibile trarre dall'anteprima dei dati di World Robotics 2019, il rapporto annuale stilato dall' IFR (International Federation of Robotics). Essa ha pubblicato delle statistiche sulla crescita dei robot di servizio, raccogliendo i dati in due categorie principali:

- robot domestici;
- robot per l'intrattenimento.

Il gruppo stima una crescita media pari al 12% all'anno di questo mercato nel periodo tra il 2020 e il 2022. Come prevedibile l'industria automobilistica è regina dell'offerta totale, assorbendo quasi il 30% dell'offerta, seguita da quella elettronica e da quella meccanica con il 10% della domanda globale. A mostrare interesse sono i settori della logistica e dei robot ad uso personale/domestico con incremento inarrestabile. In forte e rapido sviluppo è anche il mercato dei robot dei servizi personali come da fonte IFR. La previsione delle vendite di tutti i robot per compiti personali e domestici è stimata a superare il 22,1 milioni di unità, tradotto in dollari il valore è di circa 4,6 miliardi, nel 2020. Esponenzialmente tali valori potrebbero raggiungere rispettivamente nel 2022 i seguenti valori: 61,1 milioni di unità e 11,5 miliardi di dollari di vendite. L'IFR inoltre afferma che sempre più robot aspirapolvere entreranno nelle famiglie nel mondo, stimando che saranno venduti più di 17,6 milioni di unità di robot domestici (quali aspirapolvere e tosaerba autonomi) nel corso del 2020. Mentre nel 2022 le vendite dovrebbero aumentare mediamente del 46% annuo, quindi si desume che nel 2022 saranno vendute più di 55 milioni di unità.

Capitolo 3

Robot umanoide Pepper

In questo capitolo viene presentato il robot umanoide Pepper prodotto dall'azienda SoftBank Robotics, successivamente ci si sofferma in particolare sul suo funzionamento e le sue caratteristiche principali, nonché i sensori da esso incorporati. In seguito viene introdotto e descritto il funzionamento dell'applicazione multiplatforma Choregraphe che permette l'interazione con il robot e la visualizzazione della sua simulazione virtuale.

3.1 Funzionalità e caratteristiche

La SoftBank Robotics è la casa produttrice del robot umanoide Pepper. È il primo robot umanoide in grado di percepire le principali emozioni (gioia, tristezza, rabbia e sorpresa) e adattare il suo comportamento in base allo stato d'animo degli esseri umani che lo circondano. Questa capacità è possibile grazie all'analisi del tono delle voci e dalle espressioni. Il robot ha un'altezza di circa 120 centimetri e pesa 29 kg (Figura 3.1). Ha anche possibilità di movimento (misurati in gradi): la testa (2°), spalla (2° destra e sinistra), gomito (2 rotazioni destro e sinistro), polso (1° destro e sinistro), mano con 5 dita (1° destro e sinistro), anca (2°), ginocchio (1°), base (3°), tutto azionato da 20 motori. Il sistema operativo NAOqi compone la piattaforma. Il suo movimento può raggiungere 3 chilometri orari. Nella testa del robot sono stati installati quattro microfoni; nella bocca e sulla fronte ha due telecamere HD e dietro gli occhi è inserito un sensore di profondità 3D. Un giroscopio correda il busto mentre alcuni sensori tattili sono inseriti nelle mani e nella testa. Invece la base mobile ha tre sensori paraurti e un giroscopio, sei laser e due sonar per evitare la collisione con ostacoli. Pepper grazie ai tanti sensori adotta un approccio multisensore rilevando umani ed oggetti. Questa ricerca esplorerà il modo in cui il

rilevamento degli oggetti può essere risolto sul Softbank robot Pepper utilizzando tecniche di riconoscimento delle immagini all'avanguardia su immagini 2D e 3D. L'obiettivo formale del progetto è dunque lo sviluppo di un metodo per la rilevazione di oggetti nonché la loro presa grazie ai diversi motori e sensori disponibili per il Softbank robot Pepper e la ricerca sulla migliore combinazione di metodi atti a tale obiettivo. In questa sezione esamineremo le telecamere 2D e il sensore 3D in modo più dettagliato.

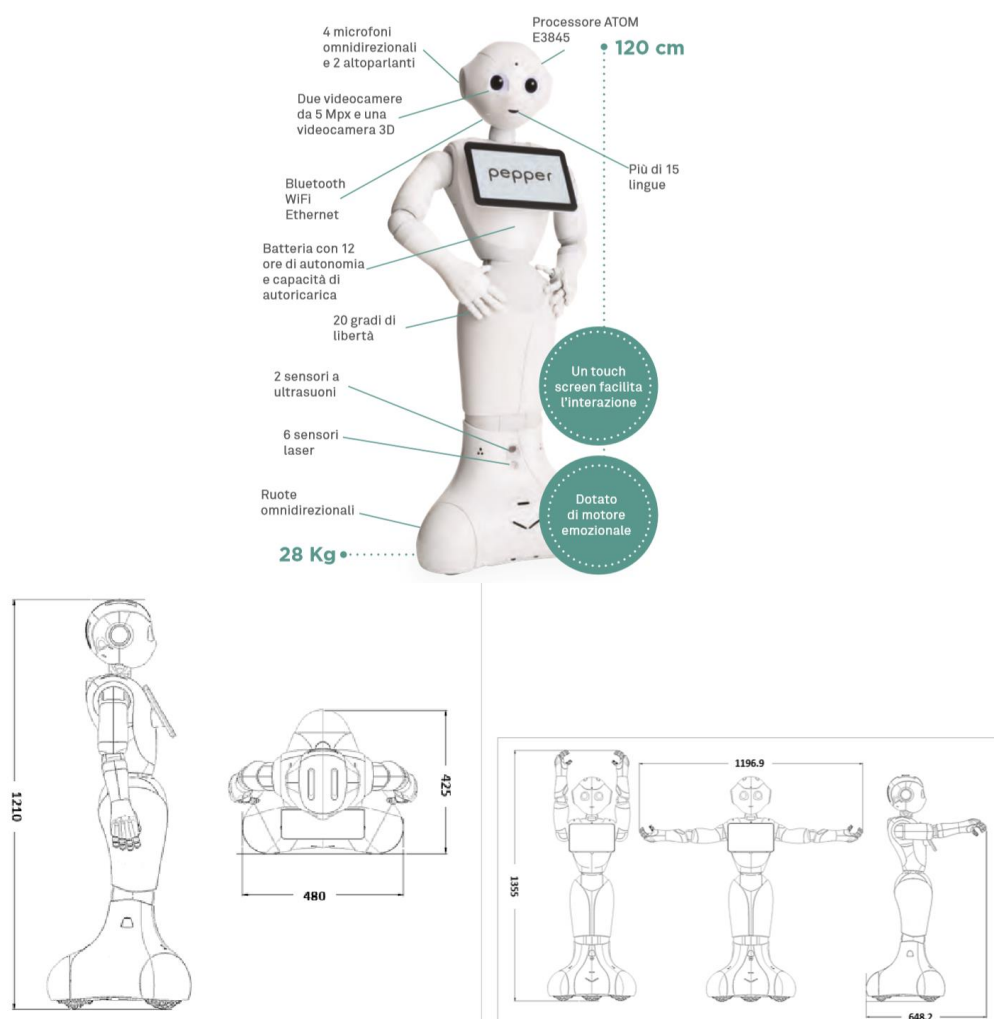


Figura 3.1. Caratteristiche e dimensioni di Pepper

3.1.1 Telecamere 2D

Le telecamere 2D di Pepper sono posizionate sulla fronte e nella bocca del robot, (Figura 3.2). Entrambe le telecamere hanno un campo visivo orizzontale di 55.2° e un campo visivo verticale di 44.3° , i campi visivi delle due telecamere si intersecano a 100 cm. Quando Pepper guarda dritto in avanti, la telecamera inferiore vedrà spesso solo il pavimento. Tuttavia, poiché gli umani sono generalmente molto più alti, saranno generalmente rivolte verso l'alto quando interagiranno direttamente con gli umani.

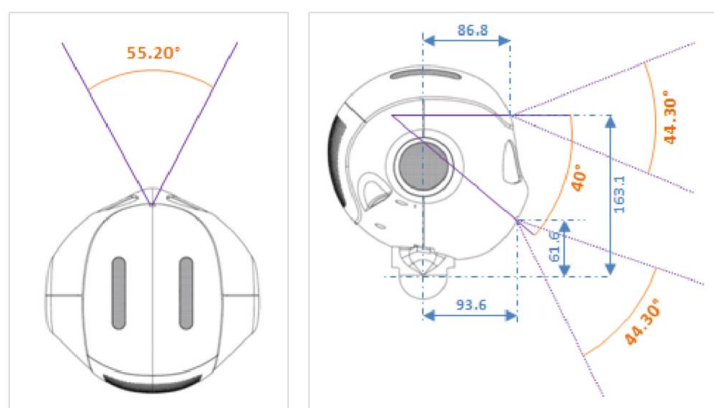


Figura 3.2. Telecamere 2D poste sulla testa di Pepper

3.1.2 Telecamera 3D

Il sensore 3D utilizzato da Pepper si trova dietro gli occhi (Figura 3.3) e puntato nella stessa direzione della telecamera 2D superiore, risulta avere un campo visivo verticale ed orizzontale più esteso. È in grado di riconoscere oggetti nell'intervallo tra i 40 e i 400 cm in un numero di differenti formati. Il sensore a volte non è in grado di trovare punti di matching dell'oggetto quando la superficie riflette il laser, ciò si traduce in dati che, sebbene generalmente stabili, contengono imperfezioni. Mentre il sensore deve riportare informazioni fino a 4 metri, la risoluzione limitata della videocamera produce dati rumorosi. Durante l'ispezione visiva dei dati, gli oggetti sono chiaramente definiti fino a 3 metri e una macchia rimane visibile fino alla portata massima.

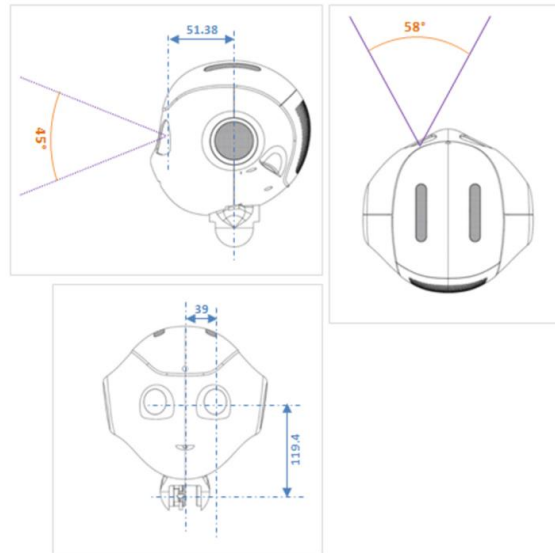


Figura 3.3. Telecamera 3D posta nell'occhio sinistro di Pepper

3.2 Choregraphe

Choregraphe è un'applicazione desktop multiplatforma che permette di monitorare e controllare Pepper, creare animazioni e comportamenti avendo la possibilità di provarli anche su un robot simulato. Choregraphe è un software di programmazione a blocchi, o programmazione visuale. Significa che dispone di una interfaccia grafica intuitiva che lo rende facilmente utilizzabile. Le box contenute in Choregraphe sono scritte interamente in python e possono essere tranquillamente modificate e create consentendo di aggiungere il proprio codice Python a un comportamento di Choregraphe. I comportamenti creati con Choregraphe sono scritti nel suo linguaggio grafico specifico: NAOqi che li interpreta e li esegue, soddisfacendo le esigenze di robotica comuni: parallelismo, risorse, sincronizzazione ed eventi. In esso è stato creato un set di dati di oggetti virtuali che integra gli scenari realistici per generare sequenze simulate utili al fine di potenziare l'implementazione dell'object recognition mediante l'apprendimento di nuove immagini attraverso il video monitor messo a disposizione da Choregraphe.

Quando si utilizza il software di interazione di Pepper (Choregraphe), è disponibile l'opzione per disattivare la vita autonoma del robot. Per impostazione predefinita, Pepper ha una vita autonoma attivata, il che significa che il robot è reattivo e cerca una stimolazione esterna con cui interagire. Quando si disabilita la vita autonoma, Pepper non risponde ma esegue i comandi di programmazione con meno riguardo alla sicurezza. Questo aspetto, sebbene meno sicuro, rende Pepper più facile da manovrare. Le due modalità non sono del tutto compatibili poiché se vogliamo spostare il robot, dobbiamo rinunciare all'aspetto dell'interazione e viceversa. Il nostro metodo di implementazione consiste nel creare una soluzione che gestisca le funzionalità del modulo offerto dal software di object recognition, mentre la vita autonoma di Pepper è disabilitata.

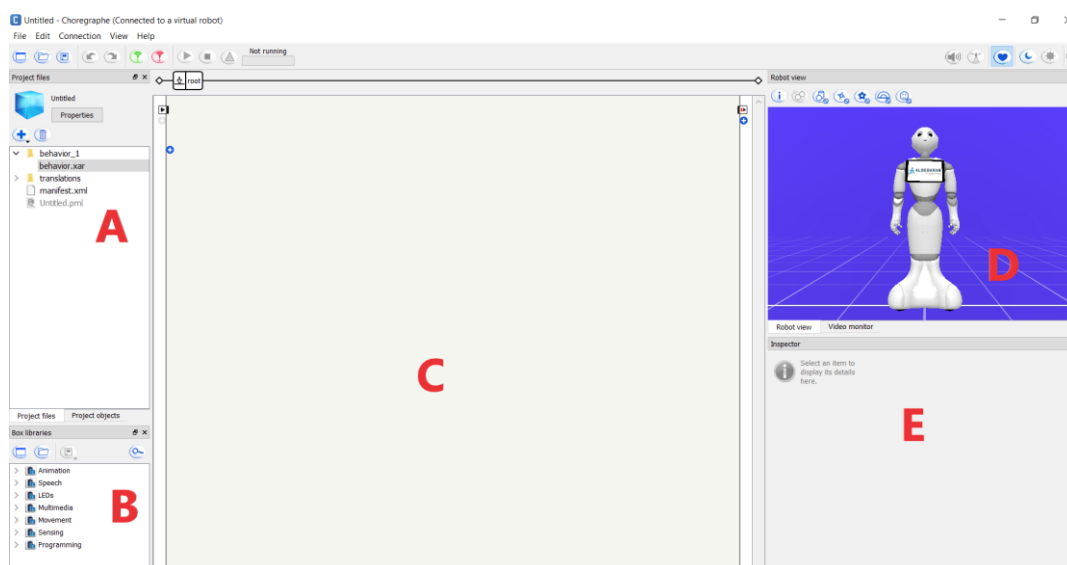


Figura 3.4. Software applicativo multiplatforma Choregraphe

Come impostazione predefinita vengono visualizzati i pannelli raffigurati come da immagine (Figura 3.4). Il pannello A rappresenta il pannello dei file di progetto, il pannello B rappresenta il pannello delle librerie, il

pannello C rappresenta il pannello del diagramma di flusso in cui verranno apposti i blocchi modulari del programma, il pannello D rappresenta la visualizzazione virtuale di Pepper e il pannello E rappresenta il pannello di controllo e quello applicativo del robot.

3.2.1 Barra degli strumenti

Nella barra degli strumenti (Figura 3.5) possiamo trovare dei pulsanti che sono scorciatoie per le azioni spesso necessarie durante la creazione di comportamenti. Si noti che sono disponibili anche le scorciatoie da tastiera.

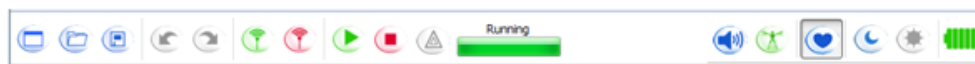


Figura 3.5. Barra degli strumenti di Choregraphe

Le prime tre scorciatoie permettono di creare, aprire o salvare un progetto; le seguenti due scorciatoie permettono di annullare o ripristinare le ultime azioni eseguite nel diagramma; le successive scorciatoie servono per connettere, disconnettere o provare a riconnettere i robot della casa produttrice SoftBank. I tasti riproduci e pausa permettono di riprodurre o interrompere il comportamento corrente. Quando si fa clic sul pulsante *Riproduci*, la barra di avanzamento, mostra lo stato del caricamento del comportamento che può essere:

- Grigio: se il comportamento non è caricato.
- Spostamento di verde e grigio: se il comportamento si sta caricando.
- Verde: se il comportamento è caricato.

La prima scorciatoia sulla parte destra della barra di strumenti consente di impostare il volume degli altoparlanti NAO; la seconda scorciatoia attiva o disattiva la modalità animazione che consente di manipolare facilmente il robot e di memorizzarne la posizione.

Questo pulsante può essere:

- Verde: la modalità animazione è disattivata.
- Arancione: stato intermedio in cui la modalità di animazione sta caricando o scaricando.
- Rosso: la modalità animazione è attivata.

La terza scorciatoia nella parte destra della barra degli strumenti a forma di cuore consente di attivare o disattivare la vita autonoma del robot, il che permette di far interagire il robot con stimolazioni esterne se attiva, o di eseguire il codice senza riguardo alla sicurezza se la funzionalità è disattivata. La quarta scorciatoia permette di attivare la modalità di riposo che imposterà la rigidità del robot che si accovaccerà su se stesso. La scorciatoia successiva permette di risvegliare Pepper dalla modalità di riposo. L'ultima scorciatoia indica il livello di batteria del robot connesso.

Questo indicatore può essere:

- Verde: se il livello della batteria è quasi al massimo.
- Arancione: se il livello della batteria è medio.
- Rosso: se il livello della batteria è molto basso e il robot si spegnerà tra qualche minuto se non lo si collega.

3.2.2 Moduli di programmazione

All'interno del diagramma di flusso, Choregraphe consente di utilizzare moduli di programmazione appropriati che consentono di far svolgere al robot Pepper diverse operazioni, tra le più importanti:

- animazioni;
- iterazione con il tablet;
- iterazione multimediale (play video, play sound, ecc...);
- gestione LED;
- speech e object recognition;

- face detection;

Queste funzionalità sono permesse grazie alla programmazione visuale offerta dal software Choregraphe che permettono la creazione di comportamenti più complessi unendo singoli moduli distinti. Le istruzioni dell'interfaccia grafica possono essere implementate anche con librerie e nuovi blocchi creati in Python.

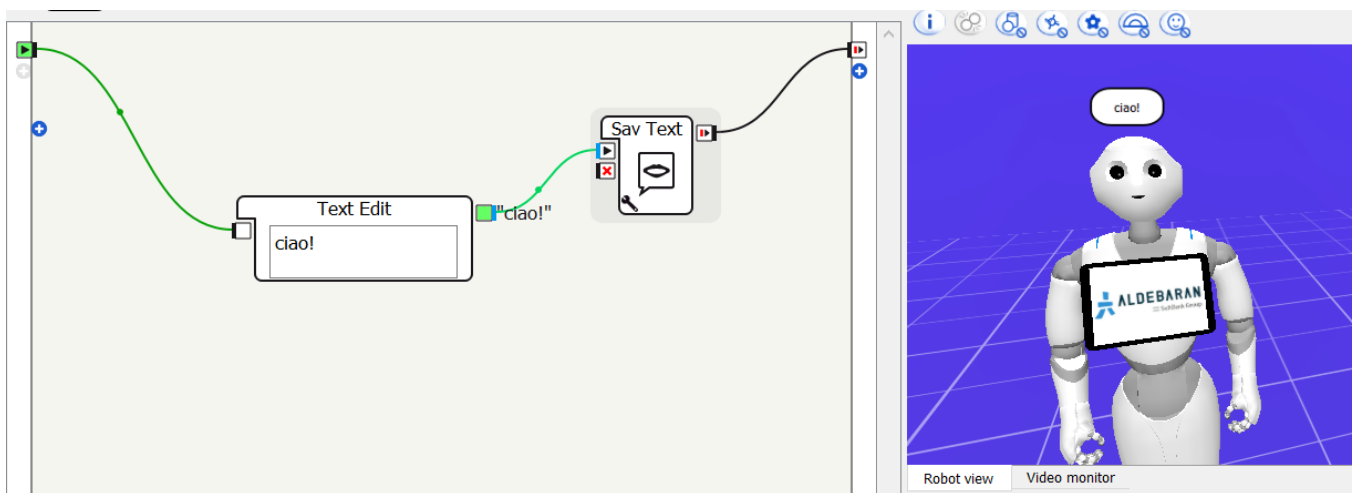


Figura 3.6. Esempio di programmazione a blocchi

Ogni modulo è l'elemento base dei comportamenti, esso può contenere una semplice azione elementare come ad esempio la casella *"Say Text"* (Figura 3.6), nonché un'applicazione molto complessa, costituita da più moduli che consentono di creare una sequenza più articolata. Per comunicare con gli altri moduli e per essere avviato o arrestato, un modulo ha alcuni connettori. I connettori dei moduli sono collegati tramite collegamenti su una logica di comunicazione basata su eventi.

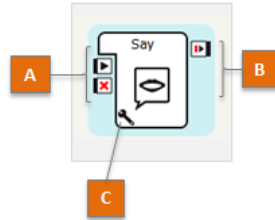


Figura 3.7. Modulo di programmazione

I principali connettori di un modulo raffigurati nella Figura 3.7 descrivono: A, punto d'ingresso dove si ricevono eventi e dove è possibile avviare o arrestare la casella; B, consente di inviare eventi e/o dati durante l'esecuzione del modulo o quando l'esecuzione del modulo viene interrotta; C, permette di visualizzare e modificare i parametri utilizzati dalla casella. L'implementazione dell'applicazione proposta successivamente incorpora diversi moduli specifici di Pepper tra i quali i più importanti ed utilizzati sono:

- **ALMotion:** fornisce metodologie relative al movimento del robot; contiene quattro principali gruppi di metodi per controllare: rigidità articolare, posizione congiunta, movimento e robot effector nello spazio cartesiano.
- **ALNavigation:** la funzionalità di navigazione consente a Pepper di apprendere posizioni precedentemente sconosciute, la definizione di una mappa, l'eliminazione degli ostacoli e il recupero dei punti associati alla localizzazione sulla mappa.
- **ALVisionRecognition** è un modulo di visione in cui il robot cerca di riconoscere diverse immagini, lati di oggetti o persino posizioni apprese in precedenza. Questo modulo si basa sul riconoscimento di punti chiave visivi ed è inteso solo a riconoscere oggetti specifici che sono stati appresi in precedenza.
- **ALMotionRecorder** o Modalità animazione: è una modalità che offre sia la possibilità di registrare le pose dei motori del robot, sia di creare facilmente movimenti in combinazione con l'Editor sequenza temporale. In questa modalità, il robot si comporta come un burattino che si può manipolare, permettendo di registrare la sua postura in una linea temporale.

- **ALSensor**: è responsabile della raccolta di eventi corrispondenti ai sensori del robot.
- **ALTouch**: è un modulo che genera l'evento `TouchChanged()` ogni volta che viene toccato il robot. L'evento `TouchChanged()` offre un elenco di `ALValue` con una parte del corpo e un valore booleano per lo stato del tocco.

Capitolo 4

Object Recognition

4.1 Che cos'è l' Object Recognition

L'object recognition è una tecnica di visione artificiale che consente di identificare oggetti in immagini o video. Il riconoscimento degli oggetti è un risultato chiave degli algoritmi di deep learning e machine learning. Un sistema di riconoscimento di oggetti trova, tramite un'immagine o un video del mondo reale circostante, il matching di modelli di oggetti che sono noti a priori tramite la memorizzazione di quest'ultimi all'interno di un database. Gli umani eseguono il riconoscimento di oggetti senza sforzo e istantaneamente, questo compito è sorprendentemente difficile per le macchine. L'obiettivo è insegnare a un computer a fare ciò che è naturale per l'uomo: ottenere un livello di comprensione di ciò che un'immagine contiene. In questo capitolo verranno prese in considerazione diverse tecniche inerenti al riconoscimento degli oggetti e introdotte alcuni algoritmi che sono stati utilizzati per il riconoscimento di oggetti in molte applicazioni, verranno descritte le fasi dell'object recognition e la loro implementazione attraverso l'utilizzo del robot Pepper. Molto spesso il riconoscimento degli oggetti viene frainteso con il rilevamento di oggetti, una tecnica molto simile per identificare gli oggetti, ma che varia nella sua esecuzione. Il rilevamento di oggetti è il processo di ricerca di istanze di oggetti nelle immagini. Nel caso del deep learning, il rilevamento degli oggetti è un sottoinsieme del riconoscimento degli oggetti, in cui l'oggetto non è solo identificato, ma lo si riesce a collocare anche in un'immagine. Ciò consente di identificare e localizzare più oggetti all'interno della stessa immagine. È possibile utilizzare una varietà di approcci per il riconoscimento degli oggetti. Recentemente, le tecniche di machine learning e deep learning sono diventate approcci popolari ai problemi di

riconoscimento degli oggetti. Entrambe le tecniche imparano a identificare gli oggetti nelle immagini, ma differiscono nella loro esecuzione.

4.2 Tecniche di Object Recognition

Le sezioni seguenti spiegano le differenze tra deep learning e machine learning per il riconoscimento di oggetti e mostrano come implementare entrambe le tecniche.

4.2.1 Object Recognition mediante il Machine Learning

Le tecniche di apprendimento automatico, anche note come tecniche di *machine learning*, sono anche popolari per il riconoscimento di oggetti e offrono approcci diversi rispetto al deep learning. Uno degli algoritmi di machine learning più utilizzato è l'algoritmo Viola-Jones, che può essere utilizzato per riconoscere una varietà di oggetti, inclusi volti e la parte superiore del corpo delle persone. Per eseguire il riconoscimento degli oggetti utilizzando un approccio di apprendimento automatico o *machine learning*, si inizia con una raccolta di immagini (o video) e si selezionano le funzionalità pertinenti in ciascuna immagine. Ad esempio, un algoritmo di estrazione delle caratteristiche potrebbe estrarre le funzionalità di linee o angoli che possono essere utilizzate per differenziare le classi nei differenti dati. Queste funzioni vengono aggiunte a un modello di apprendimento automatico, che le separerà in categorie distinte e quindi utilizzerà queste informazioni durante l'analisi e la classificazione di nuovi oggetti. È possibile utilizzare una varietà di algoritmi di apprendimento automatico e metodi di estrazione delle funzionalità, che offrono molte combinazioni per creare un modello di riconoscimento degli oggetti accurato. L'uso dell'apprendimento automatico per il riconoscimento degli oggetti offre la flessibilità di scegliere la migliore combinazione di funzionalità e classificatori per l'apprendimento. Può ottenere risultati accurati con dati minimi, il quale lo distingue dal deep learning poiché necessita una grande

quantità di dati e di una buona apparecchiatura hardware per ridurre il tempo di addestramento dei modelli.

4.2.2 Object Recognition mediante il Deep Learning

Le tecniche di *deep learning*, anche conosciute come tecniche di apprendimento profondo, sono una sottocategoria del machine learning (che letteralmente viene tradotto come apprendimento automatico) e rappresentano quella branca dell'Intelligenza Artificiale che fa riferimento agli algoritmi ispirati alla struttura e alla funzione del cervello umano, anche note come reti neurali artificiali. Il *deep learning* è un caso particolare del machine learning che si basa sull'utilizzo delle reti neurali artificiali. Le tecniche di *deep learning* sono diventate un metodo molto utilizzato per l'object recognition. La rete neurale convoluzionale, o “*CNN*”, è l'algoritmo più utilizzato e conosciuto per il *deep learning* in cui un modello impara a svolgere compiti di classificazione direttamente da immagini, video, testi o suoni. Sono utilizzate per apprendere automaticamente le caratteristiche intrinseche di un oggetto al fine di identificarlo. Ad esempio, una *CNN* può imparare a identificare le differenze tra cani e gatti analizzando migliaia di immagini di addestramento e apprendendo le caratteristiche che li rendono diversi. Esistono due approcci per eseguire il riconoscimento degli oggetti usando il *deep learning*:

- Addestramento di un modello da zero: per addestrare una rete di deep learning da zero, si raccoglie un set di dati con una categoria molto ampia di oggetti e si progetta un'architettura di rete che apprenderà le funzionalità e costruirà il modello specifico. I risultati possono essere impressionanti, ma questo approccio richiede una grande quantità di dati di allenamento.
- Utilizzo di un modello deep learning allenato: la maggior parte delle applicazioni *deep learning* utilizzano l'approccio di apprendimento trasferibile, un processo che prevede la messa a punto di un modello predefinito e già addestrato. Si parte con una rete esistente, come AlexNet o GoogLeNet, e si inseriscono nuovi dati contenenti classi precedentemente sconosciute. Questo metodo richiede meno tempo e

può fornire un risultato più rapido perché il modello è già stato addestrato su migliaia o milioni di immagini.

Il deep learning utilizza enormi modelli di reti neurali con varie unità di elaborazione; sfrutta i progressi computazionali e tecniche di allenamento per apprendere modelli complessi attraverso una enorme quantità di dati al contrario del machine learning che utilizza dei semplici algoritmi che analizzano un insieme di dati per costruire un modello previsionale in grado di classificare gli oggetti in autonomia e rispondere correttamente alle domande su un particolare dominio della conoscenza.

4.3 Fasi dell' Object Recognition

La tecnica si suddivide sostanzialmente in tre fasi principali: la fase di memorizzazione, la quale permette di raccogliere e conservare le immagini degli oggetti da riconoscere; la fase di apprendimento in cui le informazioni sulle immagini degli oggetti vengono estratte ed elaborate; la fase di ricerca, consente di analizzare la scena confrontandola con i dati immagazzinati; le prime due fasi, dipendenti e sequenziali tra loro, sono indispensabili per la realizzazione di un dataset utilizzabile. L'ultima fase, invece, è indipendente dalle altre e può essere eseguita a priori se si possiede già un dataset di immagini.

4.3.1 Fase di memorizzazione

In questa fase vengono raccolte e immagazzinate le immagini degli oggetti (Figura 4.1), utilizzando telecamere che consentano di inquadrare l'oggetto in primo piano e da diverse prospettive, il che consentirà di migliorare l'efficienza e la qualità dell'archivio digitale che si andrà a creare con le nuove immagini inserite.



Figura 4.1. Memorizzazione oggetti

4.3.2 Fase di apprendimento

In questa fase bisogna segmentare le immagini appena apprese in modo da facilitare il riconoscimento del singolo oggetto (Figura 4.2), ripetendo l'operazione per tutte le immagini che rappresentano lo stesso oggetto da diverse angolazioni e prospettive.



Figura 4.2. Apprendimento oggetti

4.3.3 Fase di ricerca

In questa fase, indipendente e successiva alla creazione di un database di immagini, si effettua la ricerca all'interno del dataset tramite calcoli matematici di probabilità consentendo di riconoscere l'oggetto tramite i punti immagazzinati in fase di apprendimento all'interno dell'immagine.

4.4 Pepper e l' Object Recognition

Choreographe, l'ambiente di sviluppo ideato per Pepper, possiede un modulo di *Object Recognition* definito dal modulo di visione "*ALVisionRecognition*" (Figura 4.3) che consente di riconoscere e apprendere oggetti in pochi secondi, basandosi sui punti chiave immagazzinati in fase di apprendimento mediante le sue telecamere 2D posizionate sulla fronte e nella bocca del robot.

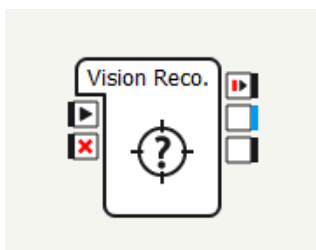








Figura 4.3. Modulo di Object Recognition

Il modulo se riconosce l'oggetto all'interno dell'immagine converte l'array dell'oggetto riconosciuto in stringhe passategli sequenzialmente. Il modulo di Object Recognition consente di visualizzare gli eventuali errori nel caso non riesca a riconoscere gli oggetti. Le immagini per essere riconosciute devono essere precedentemente apprese, Choregraphe dispone di un video monitor che mostrerà ciò che Pepper sta guardando con una delle due fotocamere attive, tutto ciò che egli apprenderà, come tutte le operazioni di estrazione, hanno come destinazione *ALMemory* essa è una memoria

centralizzata utilizzata per memorizzare tutte le informazioni chiave relative alla configurazione hardware del robot. Più nello specifico, in essa risiedono le informazioni sullo stato corrente degli attuatori e dei sensori. *ALMemory* può anche essere utilizzato, come nel nostro caso, per archiviare e recuperare immagini ed oggetti riconosciuti tramite il video monitor. Il video monitor (Figura 4.4) mette a disposizione varie operazioni.



Figura 4.4. Video monitor

Pulsante	Funzione
	Riproduce o mette in pausa il recupero del video dalla telecamera attiva sul robot.
	Passa dalla modalità di visione semplice alla modalità di apprendimento.
	Importa un database di riconoscimento visivo dal tuo computer a Choregraphe.
	Esporta l'attuale database di riconoscimento visivo di Choregraphe per salvarlo sul computer.
	Si rimuove qualsiasi oggetto già insegnato al robot dall'attuale database di riconoscimento di Choregraphe.
	Invia l'attuale database di riconoscimento di Choregraphe al robot connesso.

Capitolo 5

Implementazione del caso d'uso

5.1 Idea generale

Il caso d'uso preso in considerazione riguarda l'utilizzo del robot Pepper in un contesto domotico. Basti pensare che Pepper, grazie alla sua popolarità è diventato un membro effettivo di molte famiglie. Viene utilizzato in questo studio in una progettazione che va ben oltre l'idea originaria dei creatori di Pepper, infatti, Pepper originariamente è stato ideato come robot di compagnia e non di utilità. È per questo che più che un robot “operaio” dotato di abilità “fisiche” (capace di prendere o sollevare un oggetto, fare le pulizie o cucinare) è stato immaginato e creato come un robot da compagnia. “Farne un compagno di vita è l'obiettivo di Softbank”. L'obiettivo dunque di questo progetto è quello di utilizzare il robot Pepper per riconoscere e afferrare degli oggetti per poter essere utile in ambito assistenziale a persone anziane o comunque per chi, costretto a vivere da solo per buona parte della giornata, necessita di un supporto. Pepper seguirà un percorso preimpostato, nel percorrerlo dovrà riconoscere un oggetto che risulta utile per lo scopo, una volta esaminato e riconosciuto dovrà prenderlo e portarlo a destinazione. La robotica umanoide è un campo di ricerca emergente destinato ad aumentare, nei prossimi decenni, l'attenzione e gli sforzi di scienziati e case costruttrici. Pepper è un esempio di questa ricerca robotica, in grado di interagire con l'uomo e di adattarsi all'ambiente circostante.

5.2 Fasi di realizzazione

Il progetto è stato realizzato mediante l'utilizzo del software Choregraphe e dei suoi moduli di programmazione. Esso può essere suddiviso principalmente in due macro fasi quali: la fase di object recognition con tutte le sue sottofasi intermedie, e successivamente il modello prensile. In questo capitolo, dunque, verranno descritte inizialmente le sottofasi comprese nell'analisi dell'algoritmo di object recognition che comprende la fase di apprendimento, la creazione di un database composto dalle immagini apprese ed infine, il riconoscimento dell'oggetto; l'ultima parte del progetto comprende la realizzazione del modello prensile che consentirà la presa dell'oggetto riconosciuto.

5.2.1 Apprendimento dell' oggetto

Choregraphe ci mette a disposizione il video monitor, un'interfaccia che ci permette di visualizzare la telecamera attiva del robot connesso, con cui è possibile apprendere un nuovo oggetto. Per far sì dunque che Pepper riconosca un oggetto bisogna innanzitutto connettersi al robot reale, dopodiché cliccare sul video monitor, che ci mostrerà la visuale di una delle due telecamere 2D attive sul robot; bisogna premere sul bottone "*Learn*" il che, consentirà di apprendere una nuova immagine (Figura 5.1).



Figura 5.1. Learn button

Inizierà poi un conto alla rovescia di 4 secondi, che darà il tempo di posizionare correttamente l'oggetto che si vuole far apprendere. Una volta immortalata l'immagine viene consentito di disegnare e definire i contorni dell'oggetto contenuto all'interno dell'immagine per consentire una migliore efficienza sul riconoscimento dell'immagine stessa (Figura 5.2).

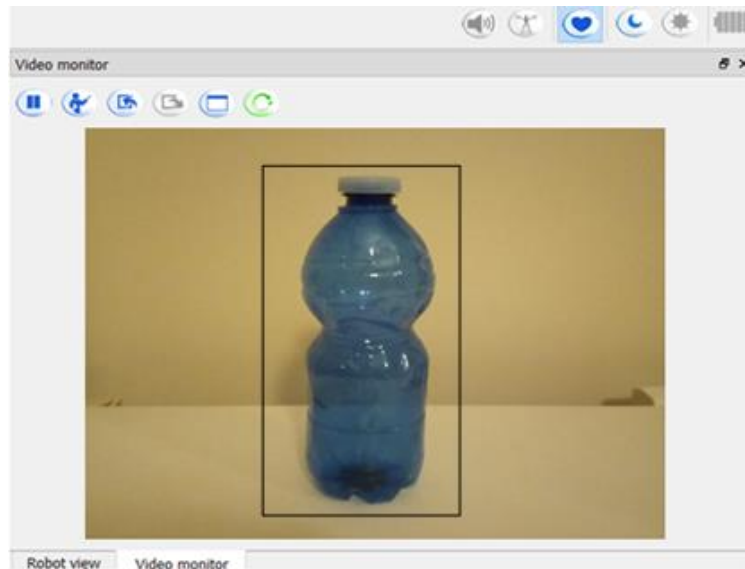


Figura 5.2. Estremità oggetto

Una volta tracciato il contorno uscirà un box per l'inserimento delle informazioni sull'immagine appena acquisita (Figura 5.3).

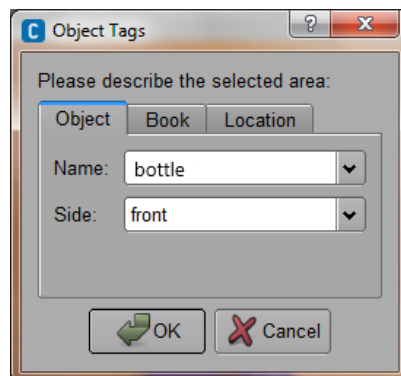


Figura 5.3. Box informazioni

Una volta inserite le informazioni necessarie, basterà cliccare sul pulsante "OK". Se l'apprendimento dell'immagine ha avuto esito positivo, verrà

mostrato a schermo un messaggio di conferma. Nel caso si voglia registrare una nuova immagine basterà fare clic sul pulsante “*play*” per riattivare la visualizzazione di ciò che Pepper sta guardando e tornare al primo passaggio dell’apprendimento.

5.2.2 Creazione Database

Il database è parte essenziale del progetto in quanto senza di esso Pepper non sarebbe in grado di riconoscere oggetti. Infatti gli oggetti vengono riconosciuti tramite un confronto tra ciò che c’è nel database e ciò che Pepper vede. Il database è memorizzato in una directory specifica. Per creare diversi database con elenchi di oggetti diversi o condividerli su altri computer, è possibile cancellare, esportare e importare i database utilizzando i pulsanti nella barra di video monitor. Nel dettaglio i pulsanti relativi al database sono:

- Importa database di visione che permette di importare un database già esistente.
- Esporta database corrente che permette di esportare e salvare un database appena creato.
- Nuovo database che permette appunto di creare un nuovo database.
- Invia database corrente al robot che permette di inviare un database appena creato al robot.

5.2.3 Riconoscimento dell’oggetto

Una volta creato il database e inserite le immagine all’interno di esso, Pepper sarà in grado di riconoscere gli oggetti appresi e dunque, sarà possibile utilizzare i moduli di programmazione offerti dal software Choregraphe specifici per l’object recognition. Nel caso d’uso il modulo di object recognition viene utilizzato successivamente alla selezione della telecamera da attivare sul robot, una volta che Pepper riconoscerà l’oggetto appreso, tramite il modulo “*Say Text*” denominato “*oggetto riconosciuto*”

Pepper dar  un feedback vocale che confermer  con successo la ricognizione dell’oggetto (Figura 5.4).

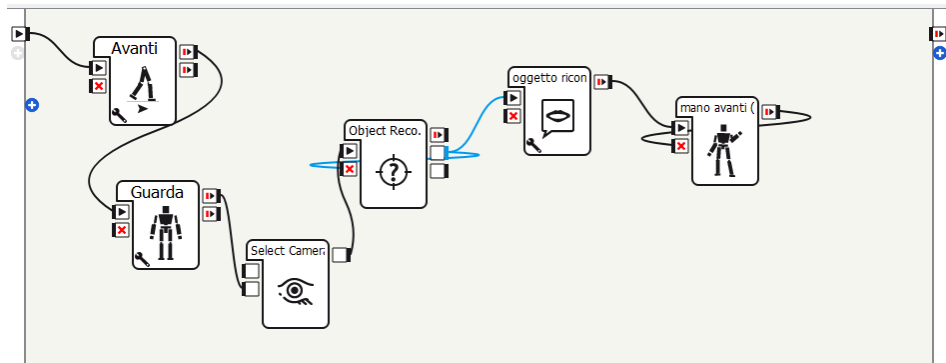


Figura 5.4. Fase di riconoscimento

5.2.4 Movimento e parte prensile

Per raggiungere ed afferrare l’oggetto   necessario utilizzare dei moduli di programmazione offerti da Choregraphe che consentono di attivare i vari motori dotati dal robot e farlo muovere, nonch  utilizzare i “*Timeline*” box per afferrare l’oggetto. Dopo il modulo di object recognition e il box vocale che conferma con successo il riconoscimento dell’oggetto, sar  utilizzato il modulo di programmazione utilizzato per raggiungere l’oggetto riconosciuto, il box di movimento “*Move To*” consente quindi di far muovere il robot attraverso lo spazio fisico e consentendo anche la rotazione di quest’ultimo (Figura 5.5). Una volta raggiunto l’oggetto verr  interpellato quindi il modulo “*Timeline*” (figura 5.6) in cui, tramite un’apposita interfaccia,   possibile realizzare dei nuovi movimenti e comportamenti di Pepper, in questo caso permetterà di regolare il braccio e piegarlo alla giusta angolazione per permettere di prendere l’oggetto su di un piano rigido.

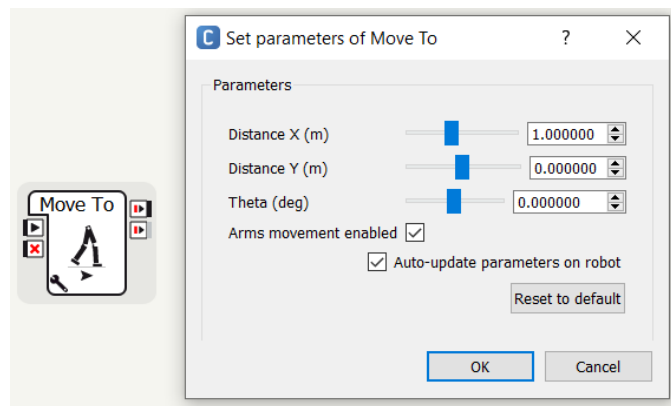


Figura 5.5. Modulo Move To

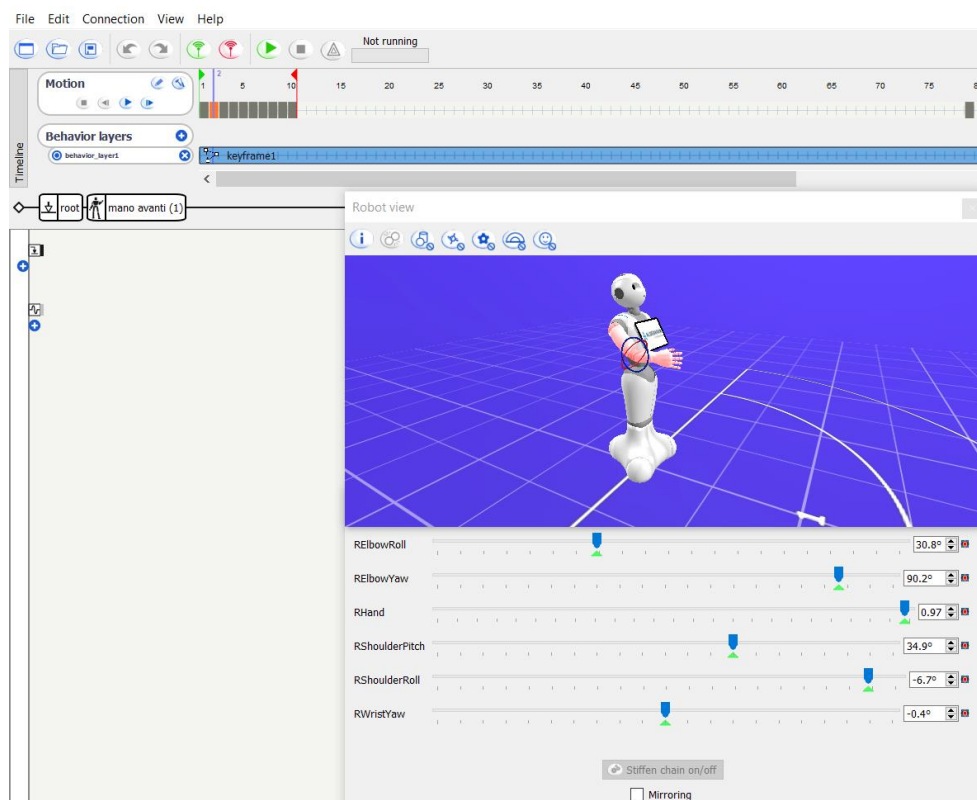


Figura 5.6. Timeline box

5.3 Implementazione

Riprendendo le fasi descritte in precedenza analizziamo nel dettaglio la loro implementazione relativa al progetto. Innanzitutto tramite la funzione del video monitor di Choregraphe e le sue relative fasi sono state apprese le immagini dell'oggetto da riconoscere dopodichè è stato creato un database costituito da esse, l'oggetto preso in considerazione in questo progetto è stato una bottiglietta, essendo cilindrica ed omogenea è bastata una sola immagine per il suo riconoscimento. Una volta salvato l'oggetto si è testato tramite la box di "*Vision Recognition*" il riconoscimento della bottiglietta. Fatto ciò si è passati alla realizzazione del progetto inserendo la parte relativa al movimento e al raggiungimento dell'oggetto nonché l'utilizzo delle "TimeLine" boxes per permettere la giusta postura del braccio destro di Pepper per consentirgli di afferrare l'oggetto nel migliore dei modi.

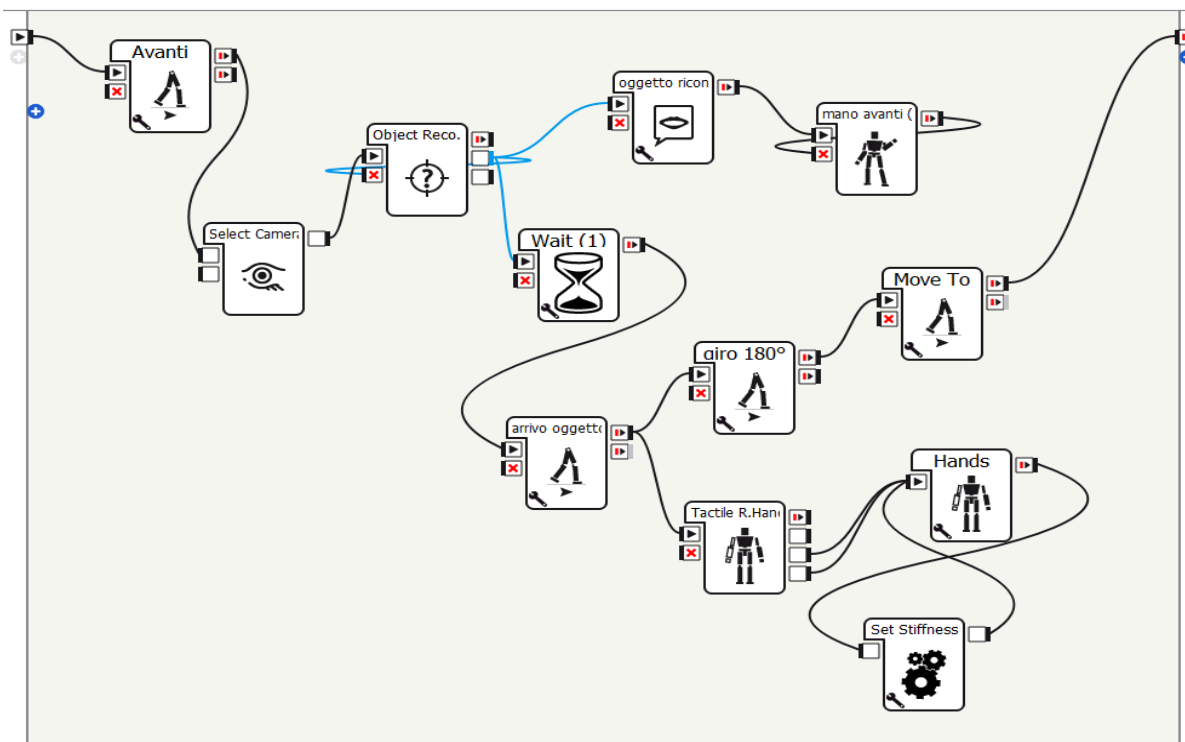


Figura 5.7. Implementazione del caso d'uso

La Figura 5.7 chiarisce il processo logico del progetto dove il primo box, denominato “*Avanti*”, permette al robot Pepper di raggiungere il primo oggetto. Il secondo modulo, denominato “*Select Camera*”, permette di selezionare quale telecamera attivare per permettere al robot di osservare l’oggetto, in questo caso è stata scelta la telecamera 2D posta in bocca. Il terzo modulo, denominato “*Object Reco.*”, già trovato in precedenza, permette la ricognizione dell’oggetto, una volta riconosciuto l’oggetto partiranno allo stesso tempo due box: la prima rappresenta il box vocale, denominato “*Oggetto Riconosciuto*”, che consente a Pepper di dire una parola, in questo caso viene esclamato “Oggetto riconosciuto!”; il secondo box, denominato “*Wait*”, è un modulo che consente di impostare un timer di attesa prima di passare al box successivo, questo permetterà di eseguire prima della fine del timer il modulo di programmazione successivo al box vocale, denominato “*Mano Avanti*”, che si riferisce al *TimeLine* inerente alla posizione che deve impostare Pepper per afferrare l’oggetto. Il box successivo al timer, denominato “*Arrivo Oggetto*”, permette al robot di raggiungere l’oggetto, successivamente da esso partono due box contemporaneamente in cui Pepper, avendo il braccio piegato nella posizione anatomica per poter afferrare l’oggetto, con il primo box in alto, denominato “*Giro 180°*”, inizia a girare su se stesso di 180° e allo stesso tempo, con il secondo box, denominato “*Tactile R. Hand*”, vengono attivati i sensori della mano destra di Pepper che permettono di stabilire quando il palmo della mano tocca la bottiglietta. Quando Pepper, ruotando sul suo asse verticale tocca con il palmo della mano la bottiglietta, verrà eseguita il box “*Hands*” che permette al robot di stringere la mano e quindi afferrare l’oggetto; il modulo “*Set Stiffness*” permette di bloccare i motori della mano e quindi mantenere stretta la presa. Infine, una volta che Pepper finirà di ruotare su se stesso di 180° con la bottiglietta in mano, verrà eseguito l’ultimo box, denominato “*Move To*”, che consentirà al robot di raggiungere la destinazione.

5.4 Testing e problemi riscontrati

Approcciarsi comunque ad una nuova tecnologia in ambito informatico è sempre una piacevole sfida, in questo caso, la novità tecnologia di Pepper ha richiesto sicuramente l'effettuazione di testing accurati per verificare la sua efficienza nell'ambito dell'object recognition. Dai vari test eseguiti in ambito di ricognizione degli oggetti si è notato che oggetti piccoli e sottili non sono facilmente riconosciuti, come un anello o una penna. Ciò che incide negativamente sul risultato è anche la luce e le tante variabili in fase di esecuzione che non possono essere pianificate in fase di sviluppo; un ambiente con una luce molto forte può portare il robot a perdere l'oggetto o addirittura non permettere il riconoscimento di quest'ultimo. Purtroppo con il software di programmazione Choregraphe non è possibile tenere conto di alcune variabili fondamentali per la rilevazione di oggetti all'interno delle immagini, come ad esempio l'intensità della luce, e quindi di conseguenza, saper a quale distanza il robot riconoscerà l'oggetto. Quindi possiamo assumere con certezza che è preferibile un ambiente chiuso con una luce regolare e oggetti di media e grande misura. Anche un pavimento non regolare può distorcere il risultato del programma, poiché un piccolo rialzamento o irregolarità potrebbero deviare il percorso di Pepper che dovendosi trovare ad una perfetta distanza dall'oggetto e posto alla giusta angolazione potrebbe non riuscire ad inquadrare l'oggetto e quindi riconoscerlo. Durante l'esecuzione della parte prensile, è stato riscontrato un problema del robot: eseguendo il box di programmazione che consente la chiusura della mano di Pepper e facendo ripetere questa azione tramite un ciclo, Pepper esegue un movimento anomalo nei motori di quest'ultima, egli infatti produce uno scatto che fa chiudere e riaprire impercettibilmente la mano, movimento, che seppur minimo, potrebbe causare la caduta di un possibile oggetto che potrebbe aver afferrato, come nel caso di studio preso in considerazione. La risoluzione di questo difetto, è stata raggiunta con il modulo "*Set stiffness*" che ha consentito di bloccare del tutto i motori, quindi è stato possibile ovviare il problema chiudendo innanzitutto la mano e poi irrigidendo i motori di quest'ultima per non consentire ulteriori movimenti ed avere una presa ferrata.

Capitolo 6

Conclusioni

In questo capitolo verranno esposti i risultati del progetto e i possibili sviluppi futuri che potrebbero essere intrapresi. Nel capitolo precedente è stato dimostrato come implementare la tecnica dell'object recognition e come integrarla con il modello prensile per realizzare un'applicazione utile per la società. Tale risultato è stato reso arduo anche dalla documentazione poco dettagliata e chiara della casa produttrice di Pepper; oltretutto Choregraphe presenta alcune limitazioni inerenti ai moduli di programmazione. Essendo una tecnologia recente non sono presenti sul web testimonianze o applicazioni sufficienti tali da essere utili a sviluppi di terzi. La risoluzione di questi problemi è sicuramente uno degli obiettivi fondamentali al fine di migliorare il lavoro delle persone che si approcceranno al robot in futuro. Il lavoro pone anche una base per lo sviluppo di applicazioni che potrebbero utilizzare la tecnologia dell'object recognition per apprendere molti oggetti e tramite una richiesta vocale recepire quale di questi si necessita per poi cercarlo all'interno di una stanza e afferrarlo, potrebbero essere utilizzate entrambe le braccia per consentire di afferrare oggetti più grandi. Qualsiasi nuova tecnologia è affascinante e sicuramente Pepper sarà in grado di incentivare l'attenzione necessaria per approfondire ulteriori studi nonché migliorare le funzionalità già presenti e idearne di nuove.

Bibliografia

- [1] “I robot fanno lavori difficili al posto nostro” – eni,
<https://www.eni.com/it-IT/ricerca-scientifica/automazione-robotica.html>
- [2] SoftBank Robotics Documentation “Box” - Aldebaran documentation, <http://doc.aldebaran.com/2-4/software/choregraphe/objects/box.html?highlight=box>
- [3] Redazione ANSA. “I robot umanoidi debuttano negli ospedali italiani” - Ansa,
http://www.ansa.it/canale_scienza_tecnica/notizie/tecnologie/2019/01/23/i-robot-umanoidi-debuttano-negli-ospedali-italiani-_74ba3b17-7f1b-4a48-91f9-d3b3ccd612bc.html (23 Gennaio 2019)
- [4] SoftBank Robotics Documentation “Timeline box” - Aldebaran documentation, http://doc.aldebaran.com/2-4/software/choregraphe/objects/timeline_box.html
- [5] SoftBank Robotics Documentation “ALVisionRecognition” - Aldebaran documentation, <http://doc.aldebaran.com/2-4/naoqi/vision/alvisionrecognition.html#alvisionrecognition>
- [6] “Robot Pepper, un robot per amico” – Robotiko,
<https://www.robotiko.it/robot-pepper/>
- [7] SoftBank Robotics Documentation “Menus, Panels and Toolbar in a glance” - Aldebaran documentation, <http://doc.aldebaran.com/2-4/software/choregraphe/interface.html>
- [8] Matthias Hirschmanner, Stephanie Gross, Brigitte Krenn, Friedrich Neubarth, Martin Trapp, Markus Vincze. Grounded Word Learning on a Pepper Robot. Proceedings of the 18th International Conference on Intelligent Virtual Agents. (Novembre 2018)

- [9] SoftBank Robotics Documentation “Flow diagram panel” - Aldebaran documentation, http://doc.aldebaran.com/2-4/software/choregraphe/panels/flow_diagram_panel.html
- [10] Diana Tartaglia. “Robot intelligente per l'assistenza ad anziani con Alzheimer” – la Repubblica, https://www.repubblica.it/salute/medicina-e-ricerca/2018/12/17/news/robot_intelligente_assiste_anziani_con_alzheimer-214280260/. (16 Aprile 2019)
- [11] SoftBank Robotics Documentation “Video monitor panel” - Aldebaran documentation, http://doc.aldebaran.com/2-4/software/choregraphe/panels/video_monitor_panel.html
- [12] Pepper (robot) – Wikipedia, [https://en.wikipedia.org/wiki/Pepper_\(robot\)](https://en.wikipedia.org/wiki/Pepper_(robot)). (2015)
- [13] SoftBank Robotics Documentation “Recognizing objects” - Aldebaran documentation, http://doc.aldebaran.com/2-4/software/choregraphe/tutos/recognize_objects.html#choregraphe-tuto-learn-object
- [14] Dignan, Larry. "Softbank, Aldebaran launch Pepper, an emotional robot". zdnet.com. CBS Interactive. (2014)