



Tesi di Laurea

Pietro Bernabei - matricola:6291312

Anno Accademico 2019/20

Contents

1	Introduzione	3
1.1	Motivazioni	3
1.2	Obbiettivo	3
1.3	Organizzazione del lavoro	4
2	Fondamenti	4
3	Costruzione del dataset	5
3.1	Acquisizione	5
3.2	Sporcatura	5
4	Costruzione del detector	6
4.1	Addestramento del rete convuluzionale	6
5	Esecuzione e risultati	6
6	Conclusioni e lavori futuri	6
7	A Manuale utente	6

1 Introduzione

1.1 Motivazioni

L'avanzamento tecnologico vissuto nell'ultimo periodo, ha portato ha un cambiamento nelle nostre vite, con l'introduzione di nuovi sistemi, applicazioni e infrastrutture, hardware e software, sulle quali si fa sempre più affidamento. Data questa particolare dipendenza dalla loro presenza, si richiede che questi siano continuativi, liberi da possibili errori o malfunzionamenti. Questi sono definiti Sistemi critici. Un particolare sistemi critico, che verrà trattato da questa tesi, è il sistema di guida autonoma di una macchina. Tecnologia in sviluppo negli ultimi anni prevede l'uso di sensori, per interagire e sentire il mondo esterno. Uno dei più importanti e comuni sensori usati all'interno dei sistemi a guida autonoma, sono le telecamere."Dispositivo elettronico in grado di acquisire immagini bidimensionali in sequenza a velocità di cattura prefissate, solitamente nella gamma visibile dello spettro elettromagnetico"[["wikipedia-telecamere"](#)"]". Queste possono subire diversi malfunzionamenti, come la sfuocatura, il congelamento del vetro, la rottura dei pixels, e altri ancora, i quali causano errori nel momento decisionale.

1.2 Obiettivo

Detto ciò, l'obbiettivo della seguente tesi è di andare a definire un sistema software, detto Detector, in grado di andare a rilevare, nel flusso di immagini generate dalla telecamara di un mezzo a guida autonoma, la presenza di malfunzionamenti nel sistema di acquisizione.

1.3 Organizzazione del lavoro

Per lo sviluppo del seguente progetto si sono definite due fasi: Acquisizione immagini Sviluppo CNN Addestramento CNN **Essendo troppo costoso disporre di un mezzo fisico, per lo sviluppo e il testing di sistemi software per la tipologia di progetto, sono stati sviluppati simulatori software di sistemi a guida autonoma. Nello sviluppo del seguente progetto, è stata impiegata la piattaforma simulativa CARLA nella sua versione 0.8.4.

Dato questo dataset, **Il Detector sviluppato nella seguente tesi, è un software che si basa su di una intelligenza artificiale, CNN (convolutional Neural Network), addestrata con dataset di immagini, da poi essere inserita poi all'interno

2 Fondamenti

2.1 Sistemi Informatici

In risposta al crescente uso di calcolatori per il controllo di servizi ed attività critiche, sono state negli anni formalizzate tecniche di progettazione e di analisi che permettano di ottenere e di garantire le necessarie qualità di un sistema dal punto di vista della adeguatezza del servizio erogato.

2.1.1 Dependability

La dependability è una delle proprietà fondamentali dei sistemi informatici insieme a funzionalità, usabilità, performance e costo. Per fornirne una prima definizione, è necessario illustrare i concetti di servizio, utente e funzione del sistema.

Definizione 1 (Servizio, Utente, Funzione). Il servizio fornito da un sistema è il comportamento del sistema stesso, così come viene per-

cepito dai suoi utenti. Un utente di un sistema è un altro sistema che interagisce attraverso l'interfaccia del servizio. La funzione di un sistema rappresenta che cosa ci attendiamo dal sistema; la descrizione della funzione di un sistema è fornita attraverso la sua specifica funzionale. Il servizio 'e detto corretto se realizza la funzione del sistema. Possiamo ora fornire una definizione di dependability.

Definizione 2 (Dependability). Nella sua definizione originale, la dependability è la capacità di un sistema di fornire un servizio su cui 'e possibile fare affidamento in modo giustificato. Una definizione alternativa, che stabilisce un criterio per decidere se un determinato servizio è dependable, definisce la dependability di un sistema come la capacità di evitare fallimenti che siano più frequenti e più severi del limite accettabile [1].

3 Costruzione del dataset

Per addestrare una intelligenza artificiale supervisionata è necessario predisporre un dataset, da usare sia per il training sia per la fase di validation. Tensorflow, richiede per l'addestramento di una cnn, un dataset composto da un training set e da un validation set. Entrambi a loro volta sono composti da dati di una categoria e l'altra divisi fra di loro. Nel caso del progetto, le due categorie di dati sono, immagini pulite e immagini modificate. Per la costruzione del dataset necessario al training del detector, sono state predisposte due fasi.

3.1 Acquisizione

La prima fase del processo di costruzione, prevede l'acquisizione di immagini pulite dal simulatore CARLA. Questo avviene grazie a un progetto di "inserisci nome del progetto github", il quale avvia n simulazioni diverse di guida autonoma. Di ciascuna di queste salva un

numero fisso di immagini, acquisite dal sensore della fotocamera RGB della macchina. Nel caso di questo progetto il numero di simulazioni avviate è stato di 500, da cui per ciascuno sono state prodotte 300 immagini (800*600)

3.2 Sporcatatura

La seconda fase del processo di costruzione, prevede la "sporcatatura" delle immagini salvate nella prima fase, immagazzinandole in maniera tale da essere poi usate dall'IA. Per "sporcatatura" di una immagine, è inteso l'applicazione di un effetto all'immagine che ne simuli un guasto al sensore RGB della vettura. Per fare questo è stato usato il progetto github "progetto secci", adattando le funzioni al progetto. I diversi effetti utilizzati nel progetto sono le seguenti: - - - - - differenza tra death pixel 50 e death pixel 200

4 Costruzione del detector

La costruzione del detector, è costituita da due fasi:

4.1 Addestramento del rete convuluzionale

5 Esecuzione e risultati

6 Conclusioni e lavori futuri

7 A Manuale utente