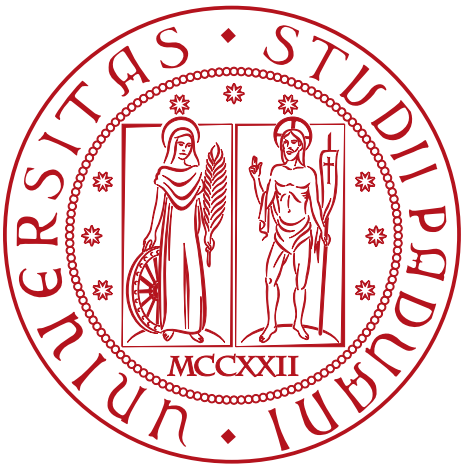
Università degli studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Predizione della Profondità con Deep Learning da Immagini di Telecamera Monoculare

Tesi di laurea

Relatore

Prof. Lamberto Ballan

Laureando

Riccardo Toniolo

Matricola 2042332

ANNO ACCADEMICO 2023-2024

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di trecentoventi ore, dal laureando Riccardo Toniolo, affiancante la dottoranda Elena Izzo, presso il gruppo di ricerca VIMP Group, Università degli studi di Padova.
Gli obbiettivi da raggiungere erano molteplici. In primo luogo era richiesto lo sviluppo di … In secondo luogo era richiesta l'implementazione di un … Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obbiettivo era l'integrazione …

Indice

Elenco delle Figure

Elenco delle Tabelle

Capitolo 1.

Introduzione

1.1. Stimare le profondità

L'avere la possibilità di misurare la profondità di un'immagine e, quindi, potenzialmente la profondità di ciascun frame all'interno di uno streaming video, apre le porte alla risoluzione di una vasta gamma di problemi che richiedono una stima precisa delle distanze tra gli oggetti all'interno di un determinato campo visivo.
Alcuni problemi appartenenti a questa classe includono:

- Prevenzione delle collisioni: lo sviluppo di algoritmi che, controllando un oggetto fisico in movimento, cercano di evitare impatti con altre entità lungo il suo percorso;
- Percezione tridimensionale: una serie di algoritmi che analizzano dati di profondità per ricostruire una scena tridimensionale dell'ambiente circostante;
- Realtà aumentata: un settore che prevede, attraverso l'uso di visori e altri dispositivi, il posizionamento di elementi grafici virtuali nel campo visivo dell'utente in modo che si integrino naturalmente con la realtà presente.

Esistono soluzioni hardware per avere delle misurazioni di profondità con alta precisione. I sistemi hardware più famosi ed utilizzati sono:

- Sensori di profondità, come ad esempio il **LiDAR**;
- Sistemi di fotografia stereoscopica: come ad esempio la **stereocamera**.

Il problema con questi sistemi hardware risiede, tuttavia, in due aspetti principali:

- Nel caso dei sensori LiDAR, il costo elevato può rendere il prodotto finale meno competitivo sul mercato o ridurre i margini di profitto per l'azienda che lo fornisce. Ad esempio, se si volesse integrare un LiDAR in un robot da giardino, il costo aggiuntivo potrebbe influire negativamente sulla competitività del prodotto.
- D'altro canto, l'integrazione di un sistema basato su fotocamere stereoscopiche presenta altre sfide pratiche. Non è sempre semplice trovare spazio per le due fotocamere necessarie e gestire la loro calibrazione può essere complicato. Questi problemi possono limitare l'applicabilità della tecnologia in ambienti dove lo spazio è ristretto o dove la calibrazione precisa è difficile da ottenere.

1.2. *Monocular Depth Estimation*

Un'altra soluzione promettente è quella di sviluppare una rete neurale in grado di prevedere correttamente le profondità di un'immagine a partire da

una singola immagine di input, un approccio noto come *Monocular Depth Estimation* (**MDE**). Se riuscisse ad essere realizzata con successo, tale soluzione permetterebbe il vantaggio di utilizzare una sola fotocamera, con il potenziale di un sensore LiDAR.
Tuttavia, questo tipo di approccio, oltre alle tradizionali sfide del *machine learning*, come la ricerca di dataset adeguati e la costruzione di un modello adatto, presenta ulteriori difficoltà, in particolare nei sistemi **embedded**, che hanno vincoli significativi in termini di memoria, energia e di potenza computazionale.
Modello particolarmente interessante di *machine learning* è PyDNet [1], [2], in quanto fortemente leggero (meno di 2 milioni di parametri) e decentemente performante per la sua dimensione. Per questo motivo è stato scelto per essere il modello di riferimento da usare come punto di partenza del tirocinio.

1.3. Obbiettivi

Il tirocinio si è quindi strutturato su due macro obbiettivi:
1. Studiare come il problema di **MDE** è stato affrontato da PyDNet V1 [1] e V2 [2]:
   - Studiarne i paper e i relativi codici;
   - Inserire nella procedura di allenamento un sistema di *logging* per analizzare i costi provenienti dalle varie *loss function*;
   - Riprodurre l'allenamento di [1] per verificare i risultati enunciati nel *paper*;
   - Migrare tutto il codice di [1] e il codice del modello di [2] da **Tensorflow** a **PyTorch**;
   - Riprodurre l'allenamento di [1] nella sua versione migrata per verificare che si ottengano gli stessi risultati e che quindi la versione migrata sia equivalente all'originale.
2. Esplorare soluzioni per ottenere modelli migliori in termini di efficacia e efficienza:
   - Verificare come variano le prestazioni di [1] al variare degli iperparametri;
   - Esplorare eventuali nuove tecniche e strategie al fine di creare un modello migliore nel compito di **MDE**.

1.4. Organizzazione del documento

Relativamente al documento sono state adottate le seguenti convenzioni tipografiche:
- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- I termini riportati nel glossario utilizzano la seguente formattazione: **parola**;
- Dopo la prima citazione del soggetto di un articolo o di un testo di riferimento ritrovabile in bibliografia, questo verrà poi menzionato solo dal suo numero di riferimento (i.e. [1]) e non più dal suo nome;
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Relativamente ai capitoli:

Glossario

stereocamera: Particolari tipi di fotocamere dotate di due obbiettivi paralleli. Questo tipo di fotocamera viene utilizzata per ottenere due immagini della stessa scena a una distanza nota. Queste immagini vengono successivamente introdotte in un algoritmo che, cercando di trovare la corrispondenza dei vari pixel tra le due immagini e conoscendo la distanza tra i due obbiettivi, triangola la profondità di tali pixel. **1.**

LiDAR: Strumento di telerilevamento che permette di determinare la distanza di una superficie utilizzando un impulso laser. **1.**

MDE: Monocular depth estimation, è il campo che si occupa di trovare soluzioni in grado di stimare le profondità a partire da una sola immagine in input. **2.**

PyTorch: Libreria *open source* per l'apprendimento automatico sviluppata da Meta AI. **2.**

Tensorflow: Libreria *open source* per l'apprendimento automatico sviluppata da Google Brain. **2.**

embedded: Un dispositivo si dice *embedded* quanto, è progettato per eseguire operazioni di elaborazione e analisi dei dati localmente, vicino alla fonte dei dati stessi, piuttosto che inviarli a un server centrale o al cloud. **2.**

Bibliografia

[1] M. Poggi, F. Aleotti, F. Tosi, e S. Mattoccia, «Towards real-time unsupervised monocular depth estimation on CPU». IEEE/JRS Conference on Intelligent Robots and Systems (IROS), 2018.

[2] M. Poggi, F. Tosi, F. Aleotti, e S. Mattoccia, «Real-time Self-Supervised Monocular Depth Estimation Without GPU». IEEE Transactions on Intelligent Transportation Systems, 2022.