**ALMA MATER STUDIORUM -
UNIVERSITA' DI BOLOGNA**

CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA

E INGEGNERIA

Corso di laurea in ingegneria e scienze informatiche

TITOLO DELL'ELABORATO

TECNICHE DI VISIONE ARTIFICIALE PER LA GUIDA AUTONOMA

Elaborato in

VISIONE ARTIFICIALE

Relatore

Prof. Raffaele Cappelli

Presentata da

Enrico Roncuzzi

Sessione III

Anno Accademico 2019/2020

Sommario

Capitolo 1

Introduzione

L’ambiente in cui ogni essere umano vive è sorprendentemente ostile, tanto che, per poter sopravvivere, l’uomo ha bisogno di un apparato sensoriale estremamente completo e raffinato. Il senso maggiormente usato è la vista, perché è in grado di fornire una descrizione dettagliata del mondo circostante. Quando, in ambito robotico, è stato affrontato il problema di come far percepire l’ambiente alle macchine, la scelta di utilizzare sensori in grado di emulare la visione umana è stata perciò naturale.

La visione artificiale è la branca dell’intelligenza artificiale che si propone di ricavare informazioni a partire dalle immagini e dai flussi video. I dati provenienti da una telecamera sono i pixel che compongono l’immagine: si tratta, perciò, di un quantitativo davvero elevato di informazione di basso livello; in altre parole, i pixel sono pressoché inutilizzabili senza un’opportuna elaborazione, il cui scopo è quello di fornire in uscita pochi dati di alto livello. Questo compito può risultare davvero molto gravoso in termini di complessità computazionale.

La visione artificiale trova applicazione in numerosissimi campi, ed ha ormai raggiunto un certo livello di maturità, tanto da essere ampiamente utilizzata anche in ambito industriale, per il controllo qualità e la sorveglianza dei macchinari pericolosi, oltre alla videosorveglianza dei luoghi aperti al pubblico e la lettura targhe (sistemi tutor installati sulle autostrade italiane e varchi elettronici), per citare qualche esempio.

Anche sul versante della ricerca, i sistemi di visione artificiale hanno raggiunto ormai un notevole livello di complessità; inoltre, contrariamente a quello che accadeva fino ad una decina di anni fa, oggi non è più necessario disporre di hardware dedicato all’elaborazione di immagini, poiché la capacità di calcolo disponibile a bordo di un comune personal computer è

sufficiente per l'esecuzione di algoritmi anche molto complessi in tempi ragionevoli.

Un ambito in cui la visione artificiale ha ricevuto un notevole impulso negli ultimi anni è quello automotive. Le industrie automobilistiche, infatti, sono particolarmente interessate a sistemi capaci di assistere il guidatore e di capire quando si sta verificando una situazione potenzialmente pericolosa, eventualmente intervenendo per mitigarne le conseguenze. Questo tipo di ricerca è attivamente sostenuto anche da importanti istituzioni civili, come l'Unione Europea, al fine di diminuire l'impressionante numero di decessi per incidenti stradali. La ricerca sta inoltre progredendo anche con lo sviluppo di sistemi di guida totalmente automatica, una prospettiva accarezzata dai ricercatori già negli anni '80, e ritornata attuale negli ultimi anni; a questo secondo ambito sono particolarmente interessate le organizzazioni militari, il cui obiettivo è quello di avere a disposizione flotte di veicoli autonomi da utilizzare negli scenari di guerra.

Sia che si tratti di assistenza al guidatore che di guida automatica, una delle funzioni più importanti che la visione artificiale può svolgere è il rilevamento di ostacoli; tra essi, quelli più vulnerabili sono senza dubbio i pedoni, e una particolare attenzione deve essere loro dedicata. Rilevare i pedoni è un compito particolarmente difficile da portare a termine, perché hanno una forma complessa che si modifica sia in funzione della postura che dei vestiti indossati; inoltre, alcune caratteristiche salienti, come la simmetria e lo sviluppo marcatamente verticale, sono comuni anche ad altri ostacoli, come i tronchi d'albero, i pali, le colonne, che costituiscono il maggior numero di falsi positivi dei sistemi di rilevamento pedoni.

Con questo elaborato si vuole mostrare qual è attualmente lo stato dell'arte per quanto riguarda la guida autonoma e fare una panoramica delle varie metodologie utilizzate all'interno di questo ambito. Viene mostrato all'interno del secondo capitolo tutto ciò che serve per poter attuare tecniche di guida autonoma, partendo dai vari sensori fino alle tecniche per farli lavorare al meglio tra di loro. Nel terzo capitolo vengono analizzate le varie

tecniche per poter effettuare il rilevamento degli ostacoli, e più in generale come vengono gestiti i vari oggetti presenti nello spazio circostante. Si conclude con il quarto capitolo, il quale mostra i principali metodi di apprendimento end-to-end; ovvero i metodi che forniscono le prestazioni migliori all'interno di questo ambito. Infine, il capitolo cinque riassume le principali conclusioni di questo lavoro di rassegna e discute possibili sviluppi futuri.

Capitolo 2

Veicoli a guida autonoma: stato dell’arte

2.1 Introduzione

I veicoli autonomi (AV) sono veicoli senza conducente che possono percepire l’ambiente circostante senza l’intervento dell’essere umano. Si propongono per essere parte integrante di un sistema di trasporto intelligente che coordina la consapevolezza ambientale, le decisioni di pianificazione e la guida assistita multilivello. Con i rapidi progressi nella ricerca e nello sviluppo di AV e nelle sue strutture di supporto, come la tecnologia di rete mobile di quinta generazione, si prevede che gli AV occuperanno il 25% del mercato globale dei veicoli privati entro il 2040. Nonostante diversi potenziali svantaggi come i prezzi elevati, la perdita di posti di lavoro nel settore dei trasporti e della logistica e le disuguaglianze sociali [1], l'introduzione di AV all’interno del mercato dei veicoli privati può fornire numerosi vantaggi alla società. Per citarne alcuni, questi vantaggi includono una migliore sicurezza del traffico, vivibilità urbana ed esperienza degli utenti [2]. In primo luogo, gli AV possono migliorare la sicurezza riducendo la probabilità di collisione del veicolo. Ricerche precedenti hanno osservato che la maggior parte degli incidenti stradali sono causati da errori umani e valutazioni errate. Gli AV possono superare i limiti dei sensi e delle reazioni umane impiegando tecnologie affidabili come radar, GPS, sensori a infrarossi e visione artificiale. In secondo luogo, si prevede che la vivibilità urbana migliori a causa delle ridotte congestioni del traffico dovute alla ridotta proprietà dei veicoli, alle ridotte infrastrutture di trasporto, come i parcheggi, che liberano terreni per attività commerciali o ricreative e al ridotto inquinamento atmosferico [3], migliorando così salute pubblica grazie al percorso ottimizzato, alla frenata più fluida e alle regolazioni della velocità.

Con lo sviluppo dei sistemi di trasporto, la sicurezza all'interno del traffico stradale è diventata una delle principali preoccupazioni in tutto il mondo negli ultimi decenni. Secondo il rapporto statistico dell'Organizzazione mondiale della sanità (OMS), gli incidenti stradali in tutto il mondo contano più di 1,24 milioni di vittime [4]. Tuttavia, il traffico stradale è un sistema complicato e influenzato da una varietà di fattori di rischio. Tra i principali abbiamo le caratteristiche del conducente, le prestazioni del veicolo, la geometria stradale e così via. Sebbene da vari anni siano state fatte le ricerche sulla sicurezza del traffico, ancora non c'è soluzione completa che tenga conto di tutti i fattori. Gli ambiti su cui lavorare per poter garantire sicurezza stradale possono essere generalmente classificati in due categorie: ricerca sull'ambiente stradale e ricerca sul veicolo. Durante l'ultimo decennio, i veicoli autonomi sono stati proposti come approccio globale, che combina i vantaggi dei due tipi di ricerca elencati sopra. Da un lato, gli AV hanno il potenziale per ottenere informazioni sull'ambiente circostante in modo più efficiente rispetto agli esseri umani, grazie all'aiuto di sensori ad alta precisione. D'altro canto, si pensa che gli AV possano colmare alcune lacune che presenta la guida umana come stanchezza e guida pericolosa. Pertanto, si può ritenere che gli AV forniranno una soluzione efficace a tutti i problemi relativi ai trasporti, come la sicurezza, l'efficienza e il consumo di energia [5][6][7].

Con gli sforzi congiunti di ricercatori e case produttrici, AV ha compiuto alcuni progressi stimolanti. Ad esempio, Waymo ha annunciato che il loro chilometraggio di prova su strada pubblica ha raggiunto i 10 milioni [8]; Nvidia ha raggiunto miliardi di miglia di guida virtuale in sicurezza all'interno del loro ambiente di simulazione chiamato DRIVE Constellation [9]. Tuttavia, l'incidente della macchina a guida autonoma di Uber avvenuto nel marzo 2018 [10] e l'incidente mortale di Tesla nel maggio 2016 [11] hanno ricordato alle persone un fatto importante: fino ad ora, le scoperte tecnologiche raggiunte in materia non sono sufficienti per garantire una guida sicura. Secondo il rapporto annuale del California Department of Motor Vehicles (DMV), ci sono stati 49 incidenti di AV verificatisi in California

nel 2018, di cui il 57% a causa di tamponamenti, del resto, la maggior parte è stata causata da ragioni non hardware, come una previsione errata o una pianificazione errata [12].

Le indagini indicano che, in assenza di altri partecipanti al traffico, i prototipi AV possono garantire il normale funzionamento dipendendo solo da una percezione accurata e da informazioni sulla posizione. Tuttavia, possono verificarsi incidenti quando i prototipi AV interagiscono con altri partecipanti al traffico [13]. Sistemi autonomi che operano in ambienti dinamici complessi richiedono modelli che generalizzino situazioni imprevedibili e ragionino in modo tempestivo. Per di più, decisioni ottimali richiedono una percezione accurata, ma la maggior parte dei modelli di visione artificiale esistenti sono ancora inferiori alla percezione e al ragionamento umano. Gli approcci esistenti alla guida autonoma possono essere classificati approssimativamente in procedure modulari e approcci monolitici di apprendimento end-to-end. Entrambi gli approcci sono messi a confronto a livello concettuale nella Figura 2.1. La procedura modulare è l'approccio standard alla guida autonoma, quello più seguito nel settore. L'idea chiave è scomporre la complessa funzione di mappatura da input di alto livello a variabili di controllo a basso livello in moduli che possono essere sviluppati, addestrati e testati in modo indipendente. Nella Figura 2.1 (in alto), questi moduli comprendono percezione di basso livello, analisi delle scene, pianificazione del percorso e controllo del veicolo. Tuttavia, questo è solo un esempio di una possibile suddivisione in moduli della guida autonoma. Gli approcci esistenti in genere sfruttano l'apprendimento automatico (ad esempio reti neurali profonde) per estrarre funzionalità di basso livello o per analizzare la scena in singoli componenti. Al contrario, la pianificazione del percorso e il controllo dei veicoli sono dominati dalle classiche macchine a stati, algoritmi di ricerca e modelli di controllo. Il vantaggio principale delle procedure modulari è che forniscono rappresentazioni intermedie interpretabili dall'uomo per ottenere informazioni sulle modalità di guasto del sistema.

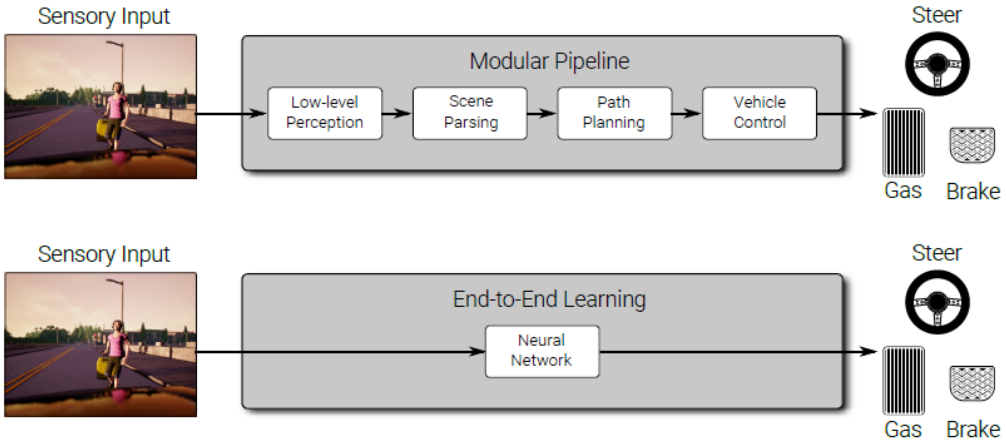


Figura 2.1: Approcci alla guida autonoma. Procedura modulare classica (in alto) vs. approccio monolitico di apprendimento end-to-end (in basso).

Uno dei principali svantaggi degli approcci modulari è il fatto che le rappresentazioni intermedie progettate dall'uomo non sono necessariamente ottimali per l'attività di guida. Inoltre, la maggior parte dei moduli vengono addestrati e convalidati indipendentemente l'uno dall'altro, facendo uso di funzioni ausiliarie di perdita. Consideriamo il problema del rilevamento di oggetti come esempio. La maggior parte degli oggetti nella scena non sono direttamente rilevanti per l'attività di guida, tuttavia l'algoritmo di apprendimento non è informato sulla rilevanza di ogni oggetto e quindi assegna a una rete neurale il compito di rilevare tutti gli oggetti con uguale importanza. Pertanto, la rete spreca capacità su oggetti irrilevanti. Ciò dimostra la difficoltà di definire rappresentazioni intermedie appropriate e opportune funzioni obiettivo.

Un'alternativa alle procedure modulari sono i modelli basati sull'apprendimento end-to-end che cercano di apprendere come risolvere l’intero problema utilizzando un modello generico come una rete neurale

profonda (Figura 2.1). I parametri di rete possono essere appresi tramite l'apprendimento per imitazione replicando il comportamento di un insegnante o utilizzando l'apprendimento per rinforzo esplorando l'ambiente e intraprendendo azioni che comportano ad una ricompensa specificata dall'utente. Tuttavia, gli approcci di apprendimento per rinforzo soffrono di problemi di assegnazione dei crediti e di formazione della ricompensa, sono generalmente lenti e possono essere applicati solo in ambienti di simulazione sicuri. L'apprendimento tramite imitazione, d'altra parte, soffre di un adattamento eccessivo e non si generalizza facilmente a scenari nuovi. Inoltre, gli approcci olistici basati sulla rete neurale sono spesso difficili da interpretare poiché si presentano come "scatole nere" all'utente e non rivelano il motivo per cui si è verificato un certo errore [16].

2.2 Sensori

I sensori sono componenti fondamentali per tutti i tipi di veicoli autonomi perché forniscono i dati necessari per percepire l'ambiente circostante e quindi aiutare il processo decisionale. Un requisito fondamentale di qualsiasi veicolo autonomo è riuscire a identificare il suo ambiente circostante e prendere decisioni intelligenti in tempo reale. Ciò richiederà una ricerca approfondita per capire quale tipo di sensore è più adatto per un particolare scopo, come può essere implementato all'interno di un sistema per funzionare in modo ottimale e affidabile e come può essere combinato con altri sensori per creare un sistema di percezione più intelligente [14].

La navigazione dei veicoli autonomi viene gestita solitamente da un insieme di sensori che ne comprende diversi tipi, come ad esempio telecamere, sensori ottici (per la gestione dell'odometria della ruota) e sensori di portata (SONAR, RADAR e LiDAR). L'insieme delle informazioni provenienti dai diversi sensori consente di sfruttare le loro caratteristiche al meglio e di affrontare i limiti dei singoli sensori, ad esempio la perdita di informazioni sulla struttura nelle telecamere o informazioni sui colori mancanti in alcuni tipi di sensori.

L'odometria della ruota misura la rotazione di una ruota e può essere utilizzata per stimare la distanza percorsa da un veicolo autonomo. Tuttavia, tale sensore non fornisce la posizione completa del veicolo (cioè tutti i sei gradi di libertà) ed è quindi tipicamente combinato con l'odometria visiva o le tecniche SLAM. I sensori di profondità, cioè SONAR, RADAR, LiDAR, forniscono informazioni aggiuntive sulla geometria e la struttura della scena. I sensori a ultrasuoni (SONAR) emettono onde sonore ad alta frequenza e misurano il tempo in cui le onde sonore raggiungono gli oggetti vicini. La distanza dagli oggetti viene calcolata attraverso il tempo di viaggio (andata e ritorno) poiché la velocità delle onde sonore è nota. RADAR e LiDAR funzionano con lo stesso principio ma utilizzano onde elettromagnetiche e impulsi di luce laser al posto delle onde sonore. A causa della maggiore lunghezza d'onda, i sensori RADAR beneficiano di un range di lavoro maggiore rispetto a Li-DAR e SONAR, ma al contempo la precisione risulta inferiore. Poiché le telecamere sono economiche, passive e facili da installare, rappresentano una scelta di sensore interessante per le auto a guida autonoma. Diversi sistemi di assistenza alla guida esistenti si basano su telecamere per il mantenimento della corsia o il rilevamento dei pedoni. Discutiamo ora brevemente i tipi di sensori più utilizzati e forniamo una breve panoramica [15].

2.2.1 LiDAR (Light Detection & Ranging)

LiDAR è una tecnologia di telerilevamento utilizzata per misurare le distanze. Utilizza sensori attivi che emettono impulsi energetici per generare illuminazione. Funziona in base al principio del tempo di volo (TOF) inviando un impulso di luce laser e misurando il tempo necessario affinché l'impulso venga riflesso. Queste misurazioni possono essere utilizzate per generare rappresentazioni 3D dell'ambiente circostante. Gli odierni sensori LiDAR sono in grado di misurare distanze a velocità superiori a 150 kilohertz (150.000 impulsi al secondo) [17] e possono essere classificati come sensori a lungo raggio con una portata di oltre 250 m.

I sensori LiDAR offrono molti vantaggi per la tecnologia dei veicoli autonomi grazie alla loro elevata precisione e accuratezza.

Figura 2.2: Esempio di sensore LiDAR

I dati raccolti da questi sensori sono necessari per tecniche di localizzazione e mappatura accurate come SLAM (Simultaneous Localization & Mapping) [18]. Il potenziale di questi sensori non è stato ancora completamente esplorato a causa del loro costo elevato e della loro scarsa disponibilità. Sono sistemi di specchi meccanici complessi che offrono una visibilità completa a 360 ° e possono costare decine di migliaia di euro. Al giorno d'oggi si sta procedendo verso lo sviluppo di sensori LiDAR a basso costo che sono più adatti per l'implementazione su larga scala. Due di questi sistemi includono LiDAR a stato solido e LiDAR a infrarossi.

2.2.2 RADAR

Il RADAR è una tecnologia che utilizza le onde radio per misurare la distanza, l'angolo e la velocità degli oggetti. Funziona sul principio della radiazione elettromagnetica che può essere utilizzata in più bande di frequenza. (ad es. 24 GHz, 77 GHz, 79 GHz). Frequenze più alte forniscono una risoluzione maggiore che consente al sistema di fare distinzione tra più oggetti in tempo reale. I sensori RADAR sono classificati come sensori a corto/medio raggio (50 - 100 m), tuttavia, alcuni RADAR sono in grado di rilevare un oggetto a una distanza di oltre 150 m [18]. I sensori radar godono di una forte robustezza in tutti i tipi di condizioni ambientali. A differenza dei sensori LiDAR, il RADAR è significativamente meno costoso e facilmente disponibile. I sensori radar sono comunemente usati nei moderni veicoli stradali che contengono sistemi ADAS (Advanced driver assistance systems). Questi sistemi sono progettati per fornire funzioni di controllo automatico della velocità e rilevamento delle collisioni. Un'altra caratteristica importante di questi sensori è il fatto di essere in grado di determinare il movimento relativo degli oggetti rilevati [19].

2.2.3 Telecamere

Una telecamera funziona secondo il principio dei sensori di luce passivi per produrre un'immagine digitale di una regione coperta. Le telecamere sono in grado di rilevare oggetti sia in movimento che statici all'interno dell'ambiente circostante. Il vantaggio principale delle telecamere rispetto a qualsiasi altro tipo di sensore è che hanno la capacità di vedere colori e trame. Questo è un enorme vantaggio per aumentare il sistema di percezione di un veicolo autonomo poiché la tecnologia consente al veicolo di identificare segnali stradali, semafori, indicazioni di corsia ecc. Le telecamere sono anche in grado di determinare la distanza da un particolare oggetto, ma ciò richiede algoritmi di elaborazione piuttosto complessi [20].

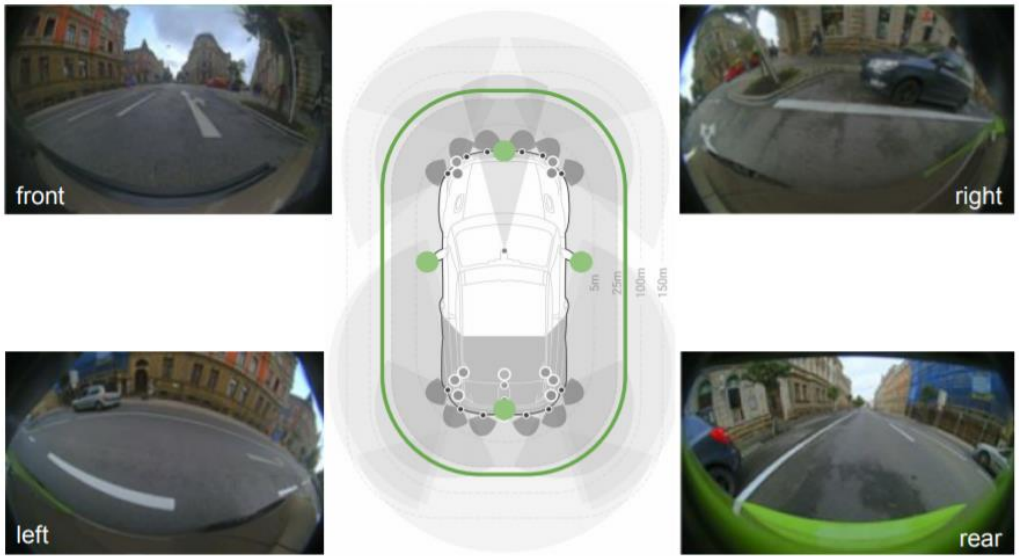


Figura 2.3: Quattro telecamere fisheye (contrassegnate in verde) forniscono una vista a 360 gradi

Un altro importante vantaggio di questa tecnologia è il basso costo e l'elevata disponibilità. Sebbene la potenza di elaborazione richiesta per analizzare i dati possa essere costosa, il tutto risulta essere comunque più economico rispetto ai sistemi LiDAR. I principali svantaggi delle telecamere sono la sensibilità alla luce a bassa intensità e l’influenza derivante dalle condizioni meteorologiche avverse. La maggior parte dei produttori di veicoli a guida autonoma sostiene che le telecamere siano una tecnologia fondamentale per la navigazione completamente autonoma, ma per essere utilizzate al massimo delle loro potenzialità devono essere interfacciate a sistemi LiDAR o RADAR.

2.2.4 GPS (Global Positioning System)

Il GPS è un sistema di radionavigazione satellitare che fornisce geolocalizzazione e informazioni sull'ora a un ricevitore GPS in qualsiasi punto della Terra, purché vi sia una vista libera verso quattro o più satelliti GPS. Un ricevitore GPS può calcolare la sua posizione cronometrando i segnali inviati dai satelliti in orbita utilizzando un metodo di "trilaterazione". La trilaterazione è il processo di determinazione della posizione assoluta o relativa di punti misurando le distanze, utilizzando la geometria di cerchi, triangoli o sfere. Il GPS è un sensore comunemente usato per la navigazione e la localizzazione nei veicoli autonomi. La maggior parte dei ricevitori GPS portatili utilizza segnali GPS a singola frequenza che possono raggiungere una precisione di circa 3 m, mentre i ricevitori di tipo commerciale possono ottenere una precisione di circa un metro. Le unità GPS di livello topografico, che sono tipicamente a doppia frequenza, hanno una precisione di pochi centimetri, andando a costare però decine di migliaia di euro.

Il principale svantaggio dell'utilizzo della tecnologia GPS per la navigazione autonoma è di avere molti fattori che possono ridurre la precisione del posizionamento. I ricevitori GPS richiedono una vista diretta con i satelliti, ma questi segnali possono spesso essere ostruiti a causa di edifici alti, alberi, tunnel, ecc. La navigazione all'interno di ambienti interni o sotterranei sono fattori importanti che compromettono l’affidamento ai sistemi GPS [21].

2.3 Fusione di sensori

La fusione dei sensori è una tecnica che combina dati forniti da fonti diverse in modo tale da creare informazioni coerenti. Le informazioni risultanti sono più certe e precise di quelle derivanti dal singolo sensore isolato. Ciò risulta essere particolarmente importante. Ad esempio, su un veicolo autonomo, è importante avere una telecamera per imitare la visione umana, ma l'informazione della distanza dall'ostacolo sarà più precisa se ottenuta da sensori come LiDAR o RADAR. La combinazione di informazioni da lidar e radar fornirà informazioni più certe sulla distanza dell'ostacolo davanti al veicolo o sulla distanza generale degli oggetti nell'ambiente.

2.3.1 Fusione di sensori per il rilevamento di oggetti 3D

Le tendenze attuali nello sviluppo di veicoli autonomi hanno mostrato un maggiore utilizzo del LiDAR. La fusione dei dati provenienti dalla telecamera e dal LiDAR offre una soluzione ottimale in termini di complessità hardware del sistema. Solo due tipi di sensori sono integrati, la telecamera per la visione e il LiDAR per il rilevamento degli ostacoli, completandosi a vicenda. Qui i dati dell'immagine vengono fusi con i dati della “nuvola di punti” 3D e, come risultato, vengono ottenute ipotesi sui box 3D degli oggetti nella scena. Una delle nuove soluzioni è l’utilizzo della rete “PointFusion” [22]. Questo metodo trova applicazione nel rilevamento di oggetti 3D.

I nuovi approcci per ottenere la fusione dei sensori utilizzando reti neurali tendono a trattare ogni segnale con una rete neurale diversa e ad integrare le rappresentazioni risultanti in una nuova rete neurale, avendo una fusione di alto livello (Figura 2.4). Questo consente di avere una soluzione concettualmente semplice, che è la tendenza attuale nello sviluppo di reti neurali, secondo il motto "le piccole reti neurali sono belle" [23].

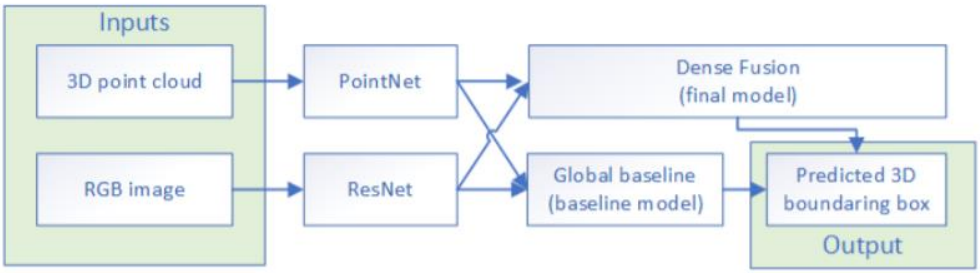


Figura 2.4: Schema a blocchi del modulo di rilevamento oggetti 3D da dati di immagini e “nuvole di punti” 3D

2.3.2 Fusione di sensori per la mappatura della griglia di occupazione

La mappatura della griglia di occupazione viene utilizzata per la navigazione e la localizzazione di veicoli autonomi in ambienti dinamici (Figura 2.5). Per implementare questa funzione, viene utilizzata la fusione di sensori di telecamere e LiDAR. Questo processo elabora ogni sensore in modo complementare agli altri: la telecamera fornisce informazioni 2D di alto livello come colore, intensità, densità, informazioni sui bordi, mentre LiDAR fornisce dati sulla “nuvola di punti” 3D. L'approccio usuale nella mappatura della griglia di occupazione consiste nel filtrare in modo indipendente tutte le celle della griglia. Tuttavia, le nuove tendenze vanno nella direzione di utilizzare dei superpixel per la mappa della griglia, dove le celle della griglia occupate da un ostacolo non vengono omesse [24].

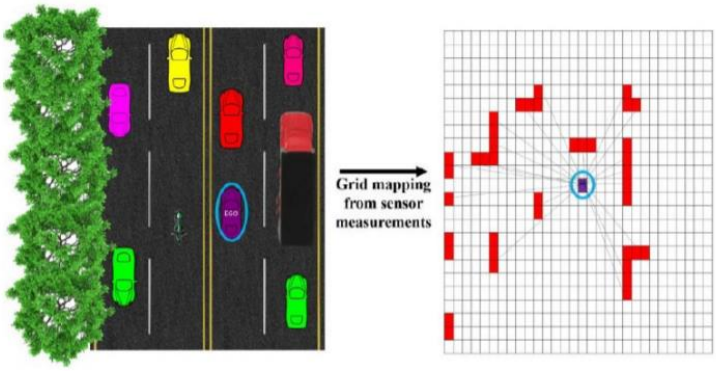


Figura 2.5: Mappa di occupazione della griglia

2.3.3 Fusione di sensori per il rilevamento e l'inseguimento di oggetti in movimento

Il rilevamento e il monitoraggio di oggetti in movimento è uno degli aspetti più impegnativi nel campo dei veicoli autonomi. Poiché la risoluzione di questo problema è fondamentale per la guida autonoma, la valutazione e le prestazioni della soluzione sono molto importanti. Quindi è normale che tutti i sensori esistenti montati sul veicolo vengano utilizzati. Solitamente la fusione comprende i dati provenienti dalla telecamera, dal RADAR e dal LiDAR. I primi approcci al rilevamento e al tracciamento di oggetti in movimento si concentravano sulla fusione dei dati provenienti dai sensori che seguono il tracciamento combinandole con informazioni aggiuntive date da un modulo di localizzazione e mappatura simultanea (SLAM). Può essere infine utilizzata una fusione aggiuntiva a livello dell'inseguimento degli oggetti per avere una percezione generale dell'ambiente. Un approccio più recente esegue il rilevamento a livello RADAR e LiDAR. Vengono inviate le regioni di interesse dalle "nuvole di punti 3D" del lidar al classificatore basato su telecamera, dopodiché tutte queste informazioni vengono fuse insieme. Le informazioni dal modulo di fusione alimentano il modulo di tracciamento, per ottenere l'elenco degli oggetti in movimento. Includendo la classificazione degli oggetti rilevati da più sensori, il modello percepito dell'ambiente viene migliorato [25].

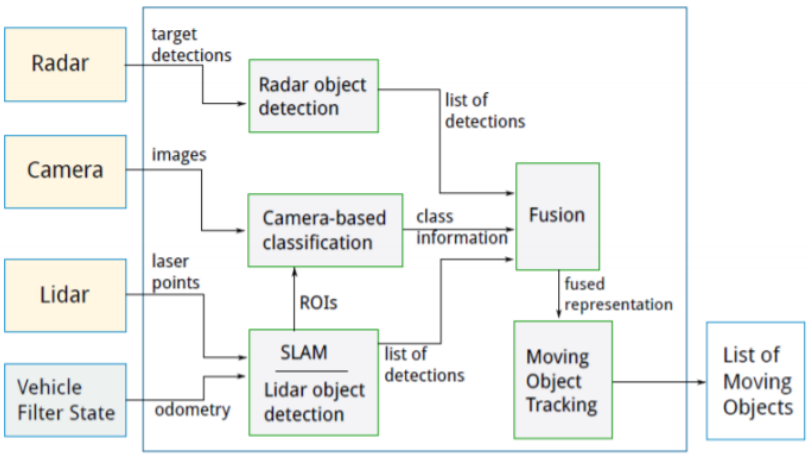


Figura 2.6: Sistema di percezione a più sensori

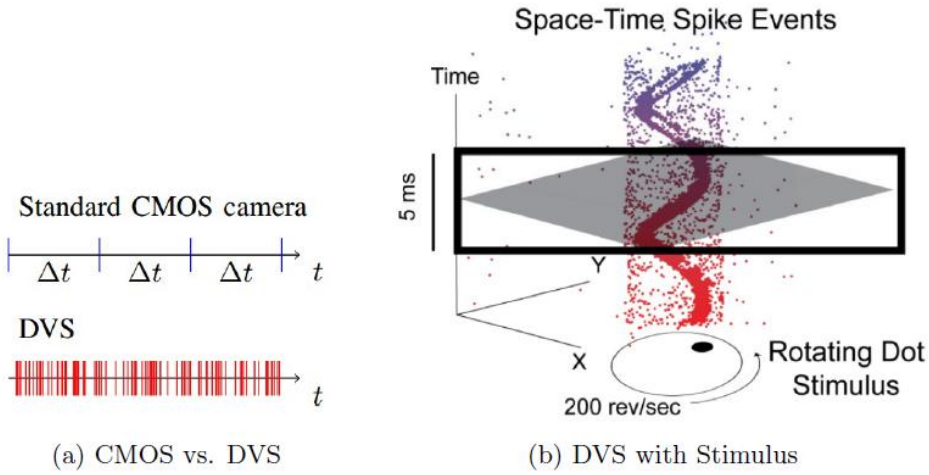
Nella fusione tra telecamera, RADAR e LiDAR, è normale applicare la fusione a basso livello dei dati provenienti da RADAR e LiDAR, i quali vengono pre-processati per estrarre caratteristiche ed informazioni sugli oggetti. Dopodiché le informazioni entrano in un blocco di fusione di alto livello che considera anche gli input della telecamera. In questo contesto, la fusione di basso livello è responsabile per quanto riguarda la localizzazione e la mappatura, mentre il rilevamento e la classificazione sono risultati della fusione di alto livello. L'unione delle informazioni ottenute dalla fusione di basso livello come input per la fusione di alto livello può essere considerata una delle fasi più critiche nella percezione dei veicoli autonomi. Il miglioramento delle prestazioni nell'associazione dei dati e della classificazione del movimento può essere ottenuto utilizzando le informazioni sulla forma e sulla classe dell'oggetto per la scelta del metodo per il rilevamento degli oggetti. In base alla distanza dell'oggetto dal veicolo, un sistema di tracciamento può essere commutato tra un punto e un modello box 3D [26]. Ciò porta alla conclusione che le informazioni ottenute da una telecamera sono fondamentali anche nelle attività di localizzazione e tracciamento e la tendenza per il futuro è quella di estrarre informazioni contestuali dagli ambienti di traffico urbano per migliorare l'efficienza del sistema di tracciamento.

2.4 Calibrazione

La calibrazione geometrica riguarda la stima dei parametri intrinseci ed estrinseci di uno o più sensori al fine di far corrispondere accuratamente i punti del mondo 3D alle misurazioni 2D. Marcatori fiduciari e scacchiere sono spesso usati per facilitare la stima dei parametri [27].

Dall'inizio degli anni '70 è possibile trovare vari metodi per la calibrazione della telecamera. Heikkila e Silven [28] furono i primi a considerare l'intera procedura di calibrazione, inclusa l'estrazione del punto di controllo, l'elaborazione del modello e la correzione dell'immagine. Hanno proposto una procedura in quattro fasi per ottenere i parametri di un modello fisico di telecamera e affrontare il problema della compensazione delle distorsioni dell'immagine. I veicoli moderni sono in genere dotati di più sensori diversi tra loro con l'obiettivo di aumentare la robustezza e la copertura. Sono state proposte diverse procedure di calibrazione per soddisfare le esigenze di insiemi di sensori così grandi.

Mentre i primi approcci si basano sull'estrazione manuale dei punti di interesse nelle scansioni laser, Kassir e Peynot e Andreasson e Lilienthal [29] propongono i primi sistemi completi di calibrazione automatica da camera a campo. Heng [30] affronta il problema della stima dei parametri intrinseci ed estrinseci di un impianto multicamera senza sovrapposizioni di campo visivo. Sempre Heng estende questo lavoro rimuovendo l'obbligo di modificare l'ambiente utilizzando una mappa e caratteristiche naturali invece di marcatori fiduciari.



(a) CMOS vs. DVS (b) DVS with Stimulus

Figura 2.7: Telecamere per eventi. (a) Una telecamera CMOS standard invia le immagini a una frequenza di fotogrammi fissa (blu) mentre un sensore di visione dinamico (DVS) invia eventi di picco nel momento in cui si verificano (rosso). Ogni evento corrisponde a un cambiamento di luminosità locale a livello di pixel. (b) Visualizzazione dell'output di un DVS guardando un punto rotante. I punti colorati segnano i singoli eventi. Gli eventi che non fanno parte della spirale sono causati dal rumore del sensore.

2.5 Dataset

I Dataset (ovvero “insiemi di dati”) hanno svolto un ruolo chiave nel progresso di molti campi di ricerca, fornendo esempi di problemi specifici con corrispondenti dati reali. Le valutazioni quantitative dei diversi approcci forniscono informazioni chiave sulle loro capacità e limitazioni. Esempi emblematici nel campo della visione artificiale includono i benchmark di Middlebury per il flusso stereo e ottico [31] e le sfide di riconoscimento degli oggetti VOC PASCAL [32]. In particolare, molti di questi set di dati, forniscono anche server di valutazione online che consentono un confronto equo sui set di test detenuti e forniscono ai ricercatori una panoramica aggiornata sullo stato dell'arte.

In questo modo, i progressi attuali e le sfide in corso possono essere facilmente identificati dalla comunità di ricerca. Nel contesto dei veicoli autonomi sono stati introdotti stimolanti benchmark per la ricostruzione e la stima del movimento, le attività di riconoscimento e di monitoraggio. Il tutto ha contribuito a colmare il divario tra gli esperimenti di laboratorio e le situazioni presenti nel mondo reale.

Solo pochi anni fa, i set di dati con poche centinaia di esempi annotati erano considerati sufficienti per molti problemi. L'introduzione di set di dati con molte centinaia di migliaia di esempi etichettati ha portato a scoperte spettacolari in molti campi della visione artificiale, consentendo di addestrare modelli ad alta capacità in modo supervisionato. Tuttavia, raccogliere una grande quantità di dati annotati non è un'impresa facile, in particolare per attività come il flusso ottico o la segmentazione semantica in cui sono richieste annotazioni a livello di pixel. Un'alternativa all'annotazione manuale è offerta dalle moderne tecniche di computer grafica che consentono di generare dataset sintetici su larga scala con annotazioni a livello di pixel. Tuttavia, la creazione di mondi virtuali fotorealistici richiede tempo e denaro. La popolarità di film e videogiochi ha portato a un'industria che crea contenuti 3D molto realistici che alimentano la speranza di sostituire completamente i dati reali utilizzando set di dati sintetici. Di conseguenza, diversi set di dati sintetici sono stati proposti e vengono utilizzati dai ricercatori di IA. Rimane una questione aperta, tuttavia, capire se il realismo e la varietà raggiunti saranno sufficienti per sostituire i set di dati del mondo reale e se i modelli addestrati su dati sintetici saranno in grado di generalizzare gli input del mondo reale. Le sfide includono la forma e l'aspetto degli oggetti complessi, nonché condizioni ambientali avverse come illuminazione diretta, riflessi da superfici speculari, nebbia o pioggia. Lo studio delle prestazioni di un sistema nel tempo, ad esempio in caso di cambiamenti ambientali o situazioni rare, è un altro aspetto importante per i veicoli autonomi.[33]

Dataset	Realism	Diversity	Autonomous Driving	Evaluation Server	Stereo	Reconstruction	Optical Flow	Object Detection	Traffic Sign Detection	Semantic Segmentation	Road Detection	Lane Detection	Tracking
Middlebury [581]	+	--		✔	XS	XS	XS						
EPFL Multi-View [627]	++	+		✔		XS							
DTU MVS [319]	+	-				S							
ETH3D [591]	++	+		✔	S	S							
Tanks and Temples [351]	++	+		✔		S							
SlowFlow [316]	++	++					S						
HCI Benchmark [357]	++	+	✔	✔			M						
MPI Sintel [92]	O	+		✔	M		M						
Flying Chairs [174]	--	--					L						
Flying Things [450]	-	O	(✔)		L		L						
ImageNet [160]	++	++						XL		XL			
PASCAL VOC [194]	++	++						XL		XL			
Microsoft Coco [420]	++	++						XL		XL			
Cityscapes [133]	++	+	✔	✔				L		L			
EuroCity Persons Dataset [68]	++	++	✔	✔				L					
Mapillary [487]	++	++	✔							L			
ApolloScape [307]	++	+	✔					L		XL		XL	XL
NuScenes [93]	++	+	✔					XL		XL			
Berkeley DeepDrive [755]	++	+	✔					XL		XL	XL	XL	
German Traffic Sign Recognition Benchmark [623]	++	+	✔	✔				XL	L	XL	XL	XL	
German Traffic Sign Detection Benchmark [299]	++	+	✔	✔				XL	M	XL	XL	XL	
Tsinghua-Tencent 100K [793]	++	+	✔					XL	XL	XL	XL	XL	
SYNTHIA [558]	O	+	✔							XL			
Playing for Data [551]	+	+	✔							L			
Playing for Benchmarks [550]	+	+	✔	✔			XL	XL		XL			XL
Caltech Lanes Dataset [8]	++	+	✔									M	
VPGNet Dataset [394]	++	+	✔									L	
MOTChallenge [389]	++	+		✔									M
Caltech Pedestrian Detection [172]	++	+	✔										XL
KITTI [238]	++	+	✔	✔	S	S	S	M		S	S	S	M
VirtualKITTI [221]	O	+	✔	✔	S	S	L	L		L			L

Figura 2.8: Datasets popolari in Computer Vision e Self-Driving. Panoramica dei set di dati più diffusi per stereo, ricostruzione, flusso ottico, rilevamento di oggetti, rilevamento di segnali stradali, segmentazione semantica, rilevamento di strade, rilevamento di corsie, inseguimento. I set di dati specifici per lo scenario di guida autonoma sono contrassegnati da un segno di spunta nella colonna corrispondente. Le dimensioni dei set di dati extra piccoli (XS) sono nell'ordine delle decine di esempi / scene per l'addestramento, di piccole dimensioni (S) nell'ordine delle centinaia, di medie dimensioni (M) nell'ordine di migliaia, grandi (L) ed extra grandi Set di dati di dimensioni (XL) nell'ordine di 10 e> 100 migliaia, rispettivamente. Valutazione del realismo e della diversità con { --, -, O, +, ++ } da basso ad alto.

Capitolo 3

Riconoscimento degli ostacoli

3.1 Definizione del problema

Il rilevamento affidabile di oggetti è un requisito fondamentale per la realizzazione della guida autonoma. Poiché il veicolo condivide la strada con molti altri partecipanti al traffico, in particolare nelle aree urbane, la consapevolezza della posizione degli altri partecipanti e degli ostacoli è necessaria per evitare incidenti. Il rilevamento nelle aree urbane è particolarmente difficile a causa dell'ampia varietà di possibili posizioni relative dei singoli oggetti e occlusioni causate da altri oggetti. Inoltre, la somiglianza degli oggetti tra loro o con lo sfondo e altri elementi come ombre proiettate o riflessi possono rendere il tutto ancora più complesso.

Il rilevamento affidabile dei pedoni è particolarmente difficile a causa del loro movimento complesso e altamente variabile, dai vestiti e dalle pose articolate. Questo problema è stato analizzato a fondo, ad esempio nei sistemi avanzati di assistenza alla guida per aumentare la sicurezza stradale. I sistemi di protezione dei pedoni (PPS) rilevano la presenza di persone attorno a un veicolo in movimento per mettere in guardia il conducente da situazioni potenzialmente pericolose. Mentre il conducente può ancora gestire i rilevamenti mancati di un PPS, un'auto autonoma necessita di un sistema di rilevamento dei pedoni impeccabile, indipendente dalle condizioni meteorologiche ed efficiente, per poter effettuare il rilevamento in tempo reale. Il problema del rilevamento degli oggetti è stato affrontato utilizzando varie modalità di input. Le videocamere sono il tipo di sensore più economico e più comunemente utilizzato per il rilevamento di oggetti. Lo spettro visibile (VS) viene tipicamente utilizzato per i rilevamenti diurni, mentre lo spettro infrarosso offre maggiore accuratezza per i rilevamenti notturni [34].

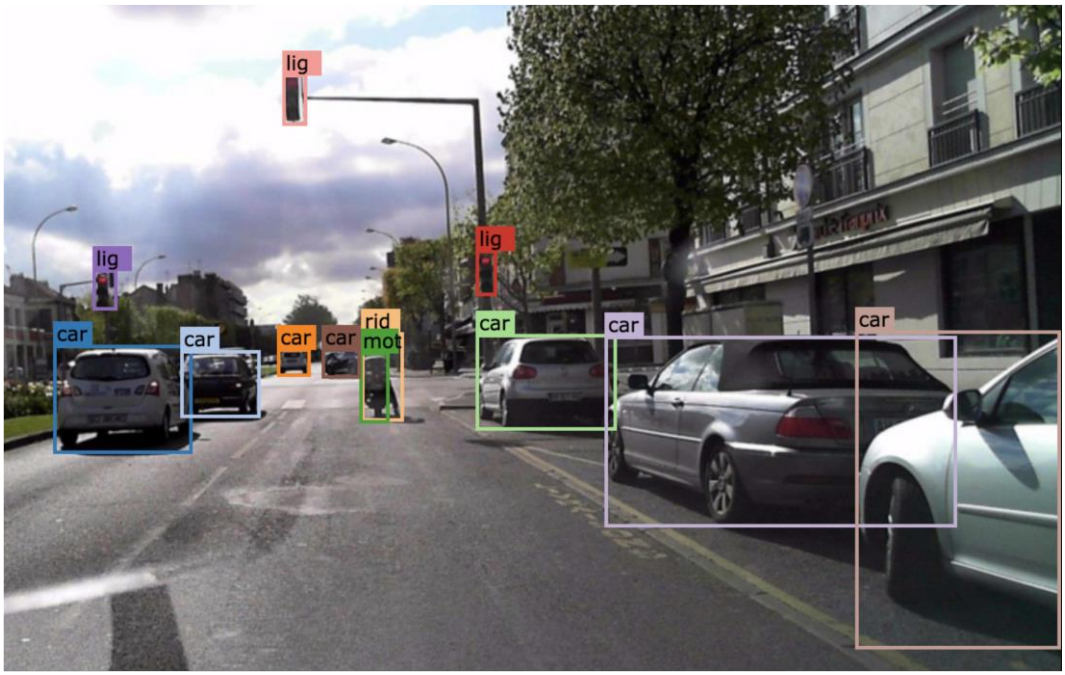


Figura 3.1: Rilevamento di oggetti. Nel rilevamento di oggetti, siamo interessati a trovare tutti gli oggetti di determinate classi in un'immagine. Questi rilevamenti sono generalmente rappresentati con riquadri di delimitazione (“bounding boxes”).

Le termocamere a infrarossi (TIR) acquisiscono la temperatura relativa, il che consente di distinguere oggetti caldi come i pedoni da oggetti freddi come la vegetazione o la strada. I sensori attivi che emettono segnali e analizzano la loro riflessione, come i sensori laser, possono fornire informazioni sulla portata utili per rilevare un oggetto e localizzarlo in un ambiente tridimensionale. Tuttavia, gli scanner laser spesso hanno una risoluzione inferiore rispetto alle videocamere. A seconda delle condizioni meteorologiche, dell'ora del giorno o delle proprietà del materiale, può essere problematico fare affidamento su un solo tipo di sensore. [35]

3.2 Tecnologie utilizzate

Il deep learning si è mostrato molto promettente negli ultimi anni nel campo del rilevamento e riconoscimento di oggetti. Le reti neurali convoluzionali (CNN) sono dedicate agli approcci basati sulla visione e sono abbastanza compatibili con l'accelerazione GPU (Graphics Processing Unit) nelle applicazioni in tempo reale. Le GPU, originariamente progettate per la modellazione e il rendering 3D, stanno ora risolvendo problemi di elaborazione delle immagini e visione artificiale e forniscono un enorme miglioramento della velocità rispetto alle implementazioni su CPU. Le GPU, quando utilizzate nel sistema di percezione dei veicoli autonomi, potrebbero elaborare i fotogrammi video a un frame rate sufficientemente alto per facilitare la guida ad alta velocità rilevando gli ostacoli molto prima della pianificazione del movimento.

3.2.1 Convolutional Neural Network (CNN)

Le CNN [48] sono speciali reti neurali multistrato progettate specificamente per dati 2D, come video e immagini. Le CNN sono animate da requisiti minimi di preelaborazione dei dati e ricevono in gran parte immagini di input non elaborate ed estraggono funzionalità autonomamente. Piccole porzioni di immagine vengono inviate al livello più basso della struttura gerarchica. Le informazioni vengono solitamente trasmesse attraverso diversi livelli della rete. Alcuni filtri digitali vengono eseguiti in modo tale da ottenere le caratteristiche salienti dei dati a ciascun livello. Il livello di rete iniziale ha una mappa delle caratteristiche che si ottiene come risultato del processo di convoluzione insieme ad alcuni bias aggiunti in precedenza. La fase successiva viene sottoposta a un processo di sottocampionamento, che in genere riduce la dimensione eseguendo un'operazione di elaborazione della media 2x2. Questa mappa delle caratteristiche, dopo il sotto campionamento, riceve un bias addestrabile e una ponderazione che viene quindi inviata a una funzione di attivazione.

Infine, gli output della funzione di attivazione vengono inoltrati a una rete feedforward completamente connessa che fornisce il risultato finale del sistema. Lo strato di convoluzione e sotto campionamento può essere ripetuto in una CNN. La CNN, quindi, estrae autonomamente le caratteristiche salienti dalle immagini e le classifica. I pesi vengono aggiornati durante il processo di addestramento mediante retro-propagazione con riferimento alla funzione obiettivo. Una funzione obiettivo basata su Support Vector Machines (SVM) o su Softmax viene generalmente utilizzata nelle CNN al livello finale Fully Connected (FC).

3.2.2 Region-Based CNN

Le CNN sono in grado di fornire una migliore precisione media (mAP) nella classificazione degli oggetti ma consumano molto tempo se applicate direttamente al rilevamento della posizione oggetti. Al fine di ottimizzare il tempo di rilevamento e il tempo di addestramento, è stata proposta una versione modificata della CNN chiamata R-CNN [49] composta da tre moduli. Il primo modulo genera una serie di proposte di possibili regioni indipendenti dalla categoria. Nel secondo modulo, un vettore di caratteristiche di lunghezza fissa viene estratto da una grande CNN per ciascuna delle possibili regioni. Il terzo modulo è costituito da un insieme di SVM lineari che sono specifiche per ogni classe (Figura 3.2)

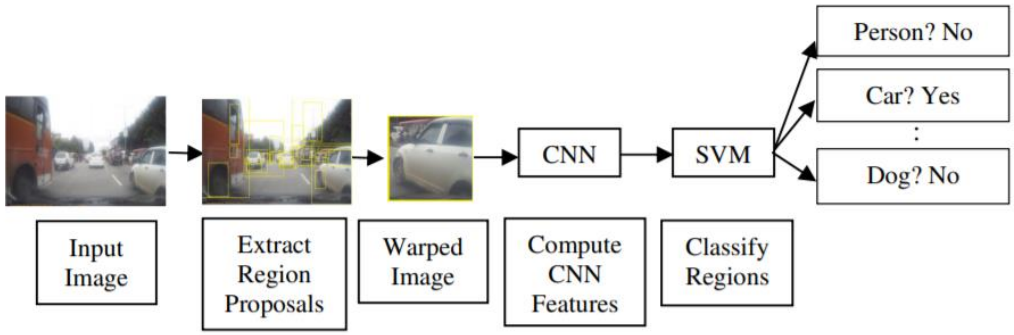


Figura 3.2: Rappresentazione di una R-CNN

3.2.3 Fast R-CNN

Fast R-CNN [50] è un nuovo metodo per realizzare R-CNN con prestazioni migliori su mAP e tempo di rilevamento. In questo metodo, le proposte di regioni vengono fatte da una rete convoluzionale separata chiamata Region Proposal Network (RPN). Le caratteristiche convoluzionali sono condivise con la rete di rilevamento.

3.3 Metodi

I sistemi classici per il rilevamento di oggetti di solito sono costituiti da più fasi che vengono applicate consecutivamente per svolgere l'attività di rilevamento. Con il successo delle reti neurali profonde (“deep neural networks”), la maggior parte di queste fasi e persino l'intera pipeline sono stati sostituiti da modelli progettati e successivamente addestrati. Iniziamo a vedere le pipeline classiche, seguite da approcci più moderni.

3.3.1 Pipeline classica

Una pipeline di rilevamento classica di solito comprende le seguenti fasi: preelaborazione, estrazione della regione di interesse (ROI), classificazione degli oggetti e verifica/raffinamento. Nella fase di preelaborazione, vengono solitamente eseguite attività come l'esposizione e la regolazione del guadagno, nonché la calibrazione della telecamera e la rettifica dell'immagine. Le regioni di interesse possono essere estratte utilizzando un approccio a finestra scorrevole, che sposta una finestra sull'immagine a diverse scale. Poiché la ricerca esaustiva è molto costosa, sono state proposte diverse euristiche per ridurre lo spazio di ricerca. Tipicamente, il numero di valutazioni si riduce assumendo un certo rapporto, dimensione e posizione dei riquadri di selezione candidati. Oltre a ciò, è possibile sfruttare le caratteristiche dell'immagine o il flusso ottico per concentrare la ricerca sulle regioni più pertinenti.

Il passaggio successivo è l'elaborazione delle regioni dell'immagine candidate, per verificarle e classificare gli oggetti. La classificazione di tutti i candidati in un'immagine può essere computazionalmente piuttosto onerosa a causa dell'enorme quantità di regioni dell'immagine che devono essere elaborate. Pertanto, è necessaria una decisione rapida che scarti rapidamente i candidati nella regione di sfondo dell'immagine. Viola et al. [36] combinano classificatori semplici ed efficienti, appresi utilizzando AdaBoost, in una modalità che consente loro di scartare rapidamente i falsi candidati mentre trascorrono più tempo in regioni promettenti.

Con il lavoro di Dalal e Triggs [37], le SVM (Support Vector Machines) lineari in combinazione con le funzionalità Histogram of Orientation (HOG) sono diventate strumenti popolari per la classificazione. Enzweiler e Gavrila [38] forniscono una panoramica degli approcci classici per il rilevamento monoculare dei pedoni. Fanno l'osservazione che SVM con funzionalità HOG funziona bene a risoluzioni più elevate pur avendo un tempo di elaborazione più elevato rispetto agli approcci a cascata che sono migliori con risoluzioni più basse e raggiungono prestazioni quasi in tempo reale. Benenson et al. [39] concludono dicendo che il numero e la diversità delle caratteristiche è chiaramente un fattore importante per le prestazioni dei classificatori poiché il problema della classificazione diventa più facile con rappresentazioni dimensionali superiori. Di conseguenza, oggi, tutti i sistemi di rilevamento di oggetti all'avanguardia utilizzano reti neurali convoluzionali per apprendere le caratteristiche espressive in modo end-to-end da grandi set di dati.

Rilevamento di oggetti da più sensori: mentre la maggior parte dei sistemi di rilevamento di oggetti si basa su singole immagini come input, esistono diversi approcci che dimostrano che l'utilizzo di più dettagli come informazioni temporali e di struttura possa migliorare le prestazioni.

3.3.2 Approcci “Part-based”

Apprendere l'aspetto di oggetti complessi è difficile perché è necessario considerare tutte le possibili variazioni. L'idea degli approcci basati sulla divisione in parti consiste nel dividere l'aspetto complesso di oggetti in movimento non rigidi come gli esseri umani in parti più semplici e di rappresentare l'articolazione utilizzando queste parti, come illustrato nella figura 3.2. Ciò fornisce una maggiore flessibilità e riduce il numero di esempi di “allenamento” necessari per l'apprendimento dell'aspetto di ciascuna parte.

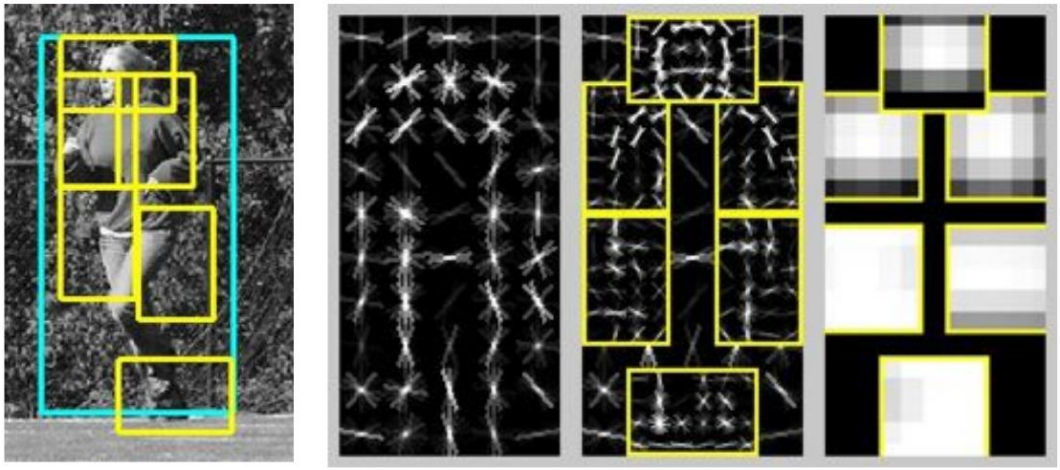


Figura 3.3: Approcci “Part-based”. Illustrazione del Deformable Part Model (DPM) proposto da Felzenszwalb et al. [40]. Il modello è costituito da un modello globale poco preciso (al centro e a sinistra), diversi modelli di parti ad alta risoluzione (al centro e a destra) e la posizione (a destra).

The Deformable Part Model (DPM), di Felzenszwalb et al. [40], tenta di scomporre l'aspetto complesso degli oggetti in parti più semplici. Viene addestrata una SVM con variabili di struttura latenti che rappresentano la configurazione del modello (posizioni delle parti) le quali devono essere

dedotte al momento dell'addestramento. Viene usato un modello globale più grossolano che copre l'intero oggetto e modelli delle singole parti a risoluzione più elevata per studiare l'aspetto di ogni parte. Un'alternativa a questa rappresentazione è l'Implicit Shape Model proposto da Leibe et al. [41], che apprende una rappresentazione altamente flessibile della forma dell'oggetto. Vengono estratte le caratteristiche locali attorno ai punti di interesse e viene eseguito il raggruppamento per costruire un “codebook” di aspetti locali che sono caratteristici per la particolare classe di oggetti in esame. Infine, apprendono le occorrenze delle voci del “codebook” per ogni oggetto. Tuttavia, Benenson et al. [42] osservano nella loro indagine sugli approcci di rilevamento che modelli basati su parti migliorano i risultati solo leggermente rispetto all'approccio molto più semplice di Dalal e Triggs [37].

I modelli basati su parti appena discussi non possono rappresentare le relazioni tra diversi oggetti, tra le loro parti e la scena, che, ad esempio, è necessaria per ragionare sulle occlusioni.

3.3.3 Deep Learning per l’apprendimento

Tutti i metodi precedenti si basano su caratteristiche definite manualmente, che sono difficili da progettare e possono essere limitate nelle loro capacità di rappresentazione. Con l’introduzione dell'apprendimento profondo (“deep learning”), le reti neurali convoluzionali sono state applicate al problema del rilevamento di oggetti, con un conseguente aumento delle prestazioni. Esempi delle tre architetture più popolari sono illustrati nella figura 3.3. Sermanet et al. [43] ha applicato le CNN al problema del rilevamento dei pedoni apprendendo l'estrazione di caratteristiche espressive in modo non supervisionato utilizzando codificatori automatici convoluzionali. Alla fine, addestrano un classificatore in modo supervisionato end-to-end estraendo le caratteristiche con uno schema a finestra scorrevole e sintonizzando congiuntamente gli auto-encoder.

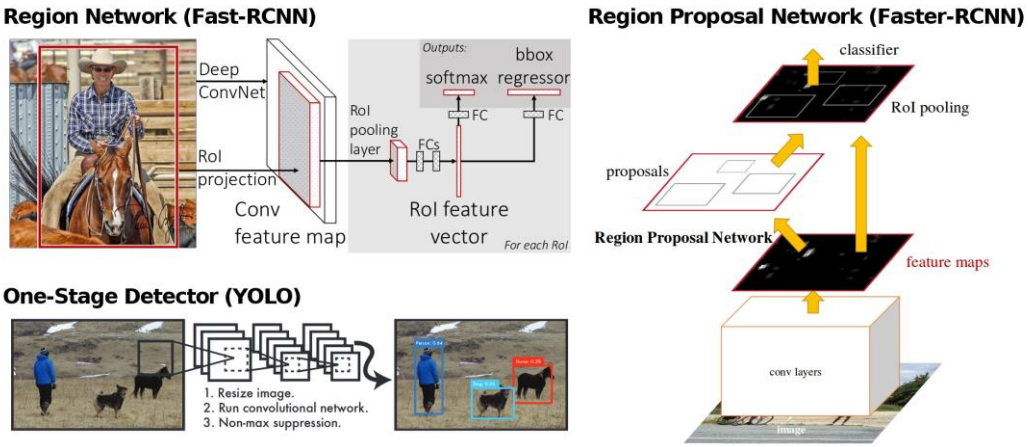


Figura 3.4: Reti per il rilevamento degli oggetti. Illustrazione di tre popolari reti di rilevamento di oggetti. In alto a sinistra: rete basata sulla regione RCNN che funziona sulle regioni. A destra: rete di proposte regionali RCNN più veloce che impara a estrarre regioni. In basso a sinistra: rilevatore a uno stadio YOLO che formula il compito di rilevamento come un problema di regressione.

Tuttavia, viene utilizzata una rete superficiale con un piccolo campo ricettivo, che consente la localizzazione precisa degli oggetti utilizzando un approccio a finestra scorrevole. Al contrario, reti più profonde con campi ricettivi più grandi complicano la localizzazione precisa perché le informazioni locali vengono estratte nei livelli precedenti, mentre le informazioni di alto livello sono rappresentate negli strati più profondi. Pertanto, Girshick et al. [44] propongono le R-CNN per risolvere il problema della localizzazione della CNN tramite il paradigma del " recognition using regions". Vengono generate molte proposte di regione utilizzando la ricerca selettiva [45], viene estratto un vettore di caratteristiche a lunghezza fissa per ciascuna proposta utilizzando una CNN e viene classificata ciascuna regione con una SVM lineare. Nelle CNN basate sulla regione, l'algoritmo di proposta della regione classica è rimasto il principale collo di bottiglia computazionale e il principale fattore limitante per le prestazioni. Pertanto, Ren et al. [46] (Faster-RCNN) ha introdotto le Region Proposal Network (RPN), che condividono caratteristiche convoluzionali a piena immagine con la rete di rilevamento, ciò non comporta a costi di calcolo aggiuntivi. Gli

RPN sono addestrati end-to-end per generare proposte di regioni di alta qualità, che sono classificate usando il rivelatore Fast R-CNN.

Alla fine, i rivelatori a uno stadio rimossero completamente la fase di proposta della regione formulando l'attività di rilevamento di oggetti come un problema di regressione. Il primo rivelatore a uno stadio di Sermanet et al. [47] era una versione convoluzionale profonda dell'approccio della finestra scorrevole. Vengono estratte le caratteristiche con una CNN e applicano una rete classificata basata su AlexNet sulle mappe delle caratteristiche estratte in modo scorrevole. Tuttavia, i rivelatori a uno stadio non potevano competere con gli algoritmi di proposta della regione. Una delle ragioni del divario di prestazioni è lo squilibrio di classi fra oggetti in primo piano e sfondo.

3.3.4 Rilevamento dei segnali stradali

Il rilevamento e il riconoscimento affidabile dei segnali stradali sono essenziali per i veicoli autonomi. L'introduzione del benchmark tedesco per il riconoscimento dei segnali stradali (GTSRB) da parte di Stallkamp et al. [51] e il benchmark tedesco di rilevamento dei segnali stradali (GTSDB) di Houben et al. [52] ha reso disponibili quelli che oggi sono i set di dati più popolari per il rilevamento dei segnali stradali. Tuttavia, le recenti CNN raggiungono già la massima accuratezza su GTSRB e GTSDB con una precisione del 100%. Pertanto, Zhu et al. [53] ha recentemente presentato Tsinghua-Tencent 100K, un nuovo benchmark per il rilevamento dei segnali stradali, che introduce nuove sfide per la comunità scientifica.

Diversi rilevatori di oggetti sono stati presi in considerazione per il rilevamento dei segnali stradali, fra cui SVM, tecniche di pattern matching, schemi di voto come rilevatori simmetrici radiali e caratteristiche dei canali integrali. Tuttavia, i recenti progressi nel deep learning hanno portato anche a una migliore classificazione dei segnali stradali.

Ciresan et al. [54] propongono un insieme composto da una CNN addestrata sulle immagini e un MLP addestrato sui descrittori delle

caratteristiche HOG per classificare i segnali stradali. Al contrario, [55] propone una CNN multiscala per apprendere caratteristiche significative invece di utilizzare funzioni artigianali come HOG. Aghdam et al. [56] propongono un rilevatore a finestra scorrevole che estrae le caratteristiche utilizzando una CNN.

Tuttavia, viene applicata la CNN utilizzando convoluzioni dilatate su diverse risoluzioni per apprendere il rilevamento dei segnali stradali a scale diverse. Infine, viene addestrata una rete convoluzionale con livelli completamente connessi per classificare le caratteristiche estratte. Garcìa et al. [57] confronta i rilevatori di oggetti generici sul popolare dataset GTSDB. Le reti regionali e i rivelatori generici a uno stadio hanno difficoltà con i segnali stradali su piccola scala. I segnali stradali possono apparire molto piccoli nell'immagine a seconda delle dimensioni, della distanza e delle occlusioni. Le reti Region-Proposal [58] offrono prestazioni migliori dei rilevatori generici e, in combinazione con Inception V2 [59] per l'estrazione di caratteristiche, ottengono risultati comparabili con GTSDB[56].

3.3.5 Rilevamento di oggetti 3D da immagini 2D

Le rappresentazioni geometriche 3D delle classi di oggetti possono recuperare molti più dettagli rispetto alle singole immagini 2D, tuttavia, la maggior parte dei rilevatori di oggetti odierni si concentra su una robusta corrispondenza 2D. Al contrario, Zia et al. [60] sfruttano il fatto che sono presenti modelli CAD 3D di alta qualità per molte classi importanti di oggetti. Da questi modelli, ottengono modelli wireframe 3D (Figura 3.5) meno precisi utilizzando l'analisi dei componenti principali e addestrano rilevatori per i vertici del wireframe.

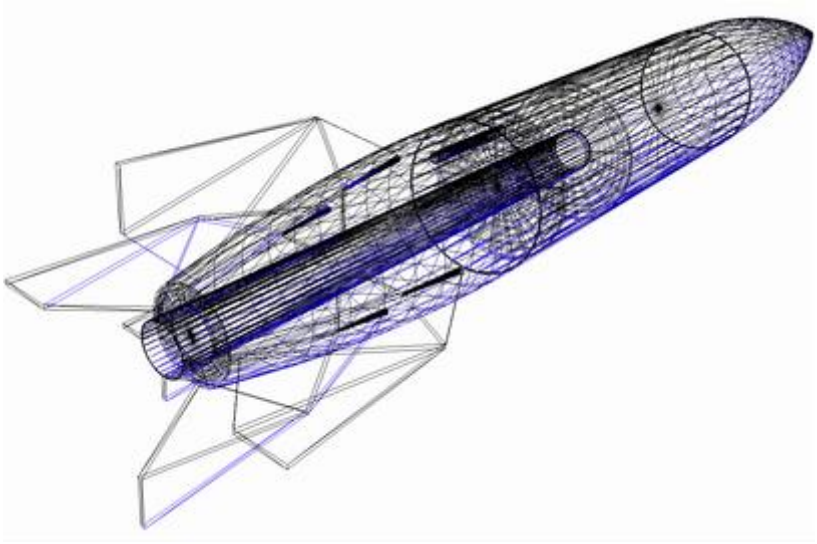


Figura 3.5: esempio di modello Wireframe

Al momento del test, generano prove per i vertici applicando densamente i rilevatori. Zia et al. estendono questo lavoro utilizzando direttamente modelli CAD 3D dettagliati nella loro formulazione, combinandoli con rappresentazioni esplicite di probabili schemi di occlusione. Inoltre, un piano terra viene stimato congiuntamente per stabilizzare il processo di stima della rappresentazione. Questa estensione mostra i vantaggi del ragionamento nel vero spazio metrico 3D.

Sebbene queste rappresentazioni 3D forniscano descrizioni più espressive degli oggetti, non possono ancora competere con i rilevatori all'avanguardia che utilizzano immagini 2D. Per superare questo problema, Pepik et al. [61] propone un'estensione 3D del modello delle parti deformabili [62] che combina la rappresentazione geometrica 3D con una robusta corrispondenza alle immagini del mondo reale. Aggiungono inoltre informazioni CAD 3D alla classe di oggetti di interesse come spunto geometrico per arricchire l’aspetto del modello.

Kundu et al. [63] addestrano una CNN a mappare le proposte di oggetti 2D in una struttura 3D completa. Aggiungono sottoreti regionali (“region-based”) per la forma 3D, mentre la previsione della struttura 3D viene aggiunta ad un'architettura Faster-RCNN / Network-on-Convolution.

Per semplificare il problema, apprendono uno spazio a bassa dimensione dai modelli CAD e lo usano come forma iniziale. La stima della forma 3D viene quindi formulata come un problema di previsione di un insieme di parametri di forma a bassa dimensione. Con una funzione obiettivo di rendering e confronto differenziabili (Render-and-Compare loss), sono in grado di apprendere la forma e la struttura 3D dalla supervisione 2D (segmentazione o profondità dell'istanza).

Al contrario, Ku et al. [64] suggeriscono un approccio più flessibile utilizzando nuvole di punti LiDAR come supervisione per evitare la dipendenza da set di dati (dataset) derivati dai modelli CAD. Usano i rilevamenti 2D di MS-CNN [65] e apprendono un modello basato su Faster-RCNN regredendolo a rettangoli di delimitazione 3D amodali orientati.

3.3.6 Rilevamento di oggetti 3D da nuvole di punti 3D

A differenza delle telecamere, i sensori di portata laser forniscono direttamente informazioni 3D accurate, il che semplifica l'estrazione di oggetti candidati e può essere utile per l'attività di classificazione poiché fornisce informazioni sulle forme 3D. Li et al. [66] sfrutta una rete neurale convoluzionale per rilevare veicoli dai dati relativi alla distanza. Viene usata una rappresentazione 2D del range di dati 3D analoghi alle immagini cilindriche, con i canali che codificano la posizione 3D dei punti. Data questa rappresentazione, prevedono simultaneamente una misura di confidenza e un riquadro di delimitazione utilizzando una singola CNN 2D.

Al contrario, Wang e Posner [67] propongono uno schema efficiente per applicare il comune approccio di rilevamento della finestra scorrevole 2D ai dati 3D. Più specificamente, discretizzano lo spazio in una griglia voxel (volumetric picture element) 3D e sfruttano la natura sparsa del problema con uno schema di voto sopra un classificatore lineare, che si dimostra essere equivalente alle convoluzioni sulla nuvola di punti 3D completa.

Engelcke et al. [68] estendono questo schema di voto incentrato sulle caratteristiche implementando un nuovo strato convoluzionale per applicare convoluzioni sparse attraverso la nuvola di punti 3D. Inoltre, migliorano la scarsità nella rappresentazione intermedia utilizzando “ReLU non-linearities” e “L1 penalty”.

Fare affidamento solo sui dati provenienti dai sensori laser rende il compito di rilevamento impegnativo a causa della densità limitata delle scansioni laser e della mancanza di informazioni sull'aspetto. Pertanto, gli approcci esistenti basati su LiDAR hanno prestazioni inferiori alle loro controparti basate su immagini 2D. Tuttavia, di recente, è stato dimostrato che la fusione di LiDAR e delle informazioni provenienti dalla telecamera consente di ridurre il divario e alla fine persino di superare le prestazioni dei rivelatori 2D all'avanguardia [69].

3.4 Dataset e parametri

I set di dati più popolari per il rilevamento di oggetti sono ImageNet [70], PASCAL VOC [71], Microsoft COCO [72], KITTI [73] e Caltech Pedestrian Detection [74]. Mentre ImageNet, PASCAL VOC e Microsoft COCO considerano il problema di rilevamento generale, il benchmark KITTI e Caltech Pedestrian Detection si concentra sulle classi rilevanti per il contesto di guida autonoma.

KITTI fornisce benchmark separati per il rilevamento 2D e 3D di automobili, pedoni e ciclisti con modalità di input 2D e 3D per entrambi i benchmark. Al contrario, il benchmark Caltech Detection si concentra solo sul problema del rilevamento dei pedoni. Recentemente, è stato presentato EuroCity Persons [75], un nuovo parametro di riferimento su larga scala per il rilevamento dei pedoni. Inoltre, diverse società, ad esempio ApolloScape, NuScenes e Berkeley DeepDrive, hanno presentato nuovi set di dati disponibili pubblicamente per il rilevamento di oggetti nelle scene di strada.

Analogamente a KITTI, ApolloScape fornisce annotazioni per il rilevamento di auto 3D ma non prende in considerazione altre classi oltre alle auto. Il set di dati Berkeley DeepDrive fornisce anche classi aggiuntive (semaforo, segnale stradale, treno) per il problema del rilevamento di oggetti stradali. Tuttavia, questi set di dati e benchmark non sono ancora stati stabilizzati nel campo.

I parametri più popolari per le prestazioni dei sistemi di rilevamento di oggetti sono la precisione media (AP) e il richiamo medio (AR)]. Inoltre, la curva di richiamo di precisione viene solitamente utilizzata per valutare i metodi. Per il calcolo della precisione e del richiamo, i rilevamenti sono classificati in veri positivi, falsi positivi e falsi negativi. A tal fine, viene considerata l'intersezione sull'unione (IOU) tra i riquadri di delimitazione rilevati e i riquadri di delimitazione corretti. Una soglia popolare per i veri positivi è un IOU di almeno il 50%.

L'AP con un IOU del 50% è noto come metrica PASCAL VOC e viene utilizzata in diversi benchmark. All’interno del benchmark KITTI La performance è valutata per tre livelli di difficoltà (facile, moderata, difficile). Mentre la metrica PASCAL VOC (IOU del 50%) viene utilizzata per pedoni e ciclisti, la metrica utilizzata per le auto è più rigorosa e richiede una sovrapposizione del 70%. Gli esempi facili hanno un'altezza minima del riquadro di delimitazione di 40 px e sono completamente visibili, mentre gli esempi moderati hanno un'altezza minima di 25 px e includono occlusioni parziali. Gli esempi difficili hanno la stessa altezza minima ma includono consistenti livelli di occlusione.

Capitolo 4

Metodi di apprendimento end-to-end per la guida autonoma

4.1 Introduzione

La guida autonoma consiste nel fornire ai veicoli degli strumenti per comprendere l'ambiente circostante in modo di essere in grado di prendere decisioni opportune e quindi compiere azioni, tenendo in considerazione molteplici fattori esterni.

Rendere possibile a un veicolo una comprensione così approfondita del mondo attorno a lui è molto complesso, poiché sono presenti numerose problematiche, tra le quali ad esempio individuare con precisione i segnali stradali, gli altri veicoli in strada e i passanti, ottenere informazioni relative agli edifici e alle linee che identificano la carreggiata, passaggi pedonali o segnali di arresto. Si devono inoltre predisporre metodi affidabili per stimare la velocità di movimento, per prevedere gli spostamenti delle altre auto e dei pedoni. Solitamente è possibile affrontare queste sfide seguendo due approcci diversi:

- il primo approccio consiste nel suddividere i problemi da risolvere in più moduli, al fine di gestirli singolarmente in modo più semplice ma ottenendo delle soluzioni che non sono complessivamente ottimali, poiché in questo caso i risultati ottenuti per ogni modulo saranno indipendenti l'uno dall'altro. Seguendo questo metodo, non si tengono inoltre in considerazione alcuni aspetti fondamentali, come la sicurezza, il tempo impiegato a raggiungere la destinazione e il comfort dei passeggeri.
- come alternativa all'approccio appena descritto si può considerare l'apprendimento della guida autonoma come un problema di apprendimento end-to-end. Seguendo questa metodologia la

gestione di percezione, pianificazione e controllo vengono combinate e viene addestrato un unico modello utilizzando reti neurali profonde, chiamate in inglese Deep Neural Network (DNN). In questo modo le varie tematiche per le quali il sistema deve essere addestrato al fine di gestirle in modo ottimale, vengono trattate come un unico problema, rendendo la soluzione più articolata ma più adatta a questo tipo di ambito.

Nelle sezioni seguenti si vanno a presentare i principali metodi di apprendimento end-to-end utilizzati oggi.

4.2 Metodi utilizzati

L'addestramento dei modelli per l'apprendimento end-to-end della guida autonoma può essere definito come una funzione che mappa l'input proveniente dai sensori, costituito ad esempio dalle immagini catturate dalla videocamera frontale dell'auto, a un comando che verrà inviato al veicolo e quindi all'azione che verrà compiuta. Il fine è quello di ottenere un sistema in grado di prendere decisioni coerenti con l'ambiente circostante e con le regole stradali.

Tipicamente sono due i metodi di apprendimento che vengono applicati nell'ambito della guida autonoma, ovvero l'apprendimento tramite l'imitazione di un comportamento tratto solitamente da esempi compiuti da esseri umani, chiamato behavior cloning, oppure l'apprendimento chiamato reinforcement learning, che permette al sistema di imparare per mezzo di tentativi effettuati in delle simulazioni cercando di massimizzare le azioni corrette e di minimizzare gli errori, tramite un approccio denominato "trial and error". Chiaramente è possibile combinare i due approcci appena descritti in modo da ottenere delle soluzioni intermedie, che vanno a sfruttare i vantaggi di entrambi i metodi.

Si vanno ora a descrivere nel dettaglio i metodi sopracitati.

4.2.1 Behavior Cloning

L'apprendimento tramite behavior cloning consiste nell'imitare il comportamento umano per un determinato compito. Nell'ambito della guida autonoma l'osservazione fornisce al sistema le informazioni necessarie per mappare le informazioni tratte dai dati dei sensori al comportamento di guida desiderato. Questo approccio ricade nella categoria di tecniche di apprendimento supervisionato, poiché i dati utilizzati per l'addestramento sono etichettati, ovvero che l'output desiderato è già noto in fase di addestramento.

Il primo successo di un veicolo a guida autonoma tramite l'utilizzo di tecniche di apprendimento di behavior cloning è avvenuto nel 1989. Dean Pomerleau, professore alla Carnegie Mellon University, università privata di Pittsburgh, in Pennsylvania, propose ALVINN [76], acronimo che sta per Autonomous Land Vehicle In a Neural Network. Questo fu il primo sistema funzionante di guida autonoma basata su una rete neurale a tre livelli pensata per quello che può sembrare il banale task di seguire la carreggiata in una strada. I test effettuati hanno mostrato infatti che se la carreggiata avesse rispettato alcune caratteristiche, il sistema sarebbe stato in grado di capire, interpretare l'ambiente e prendere le decisioni corrette a seconda della situazione che doveva affrontare. Sempre Pomerleau [77] nel 1995 disse che molto probabilmente in cinque o dieci anni al massimo i veicoli autonomi sarebbero stati disponibili in commercio.

Addestrare modelli con l'approccio behavior cloning però presenta molte più difficoltà rispetto a quelle previste da Pomerleau, poiché anche se la capacità di seguire da carreggiata è un requisito necessario non è assolutamente sufficiente per risolvere tutte le complessità della guida autonoma.

Per prima cosa è necessario raccogliere i dati che saranno fondamentali per addestrare il sistema, ovvero quelli che costituiranno il training set.

Come detto in precedenza il training set fornisce informazioni che permettono al sistema di apprendere dall'input ottenuto dai sensori un determinato output che costituisce l'effettiva azione che l'auto deve intraprendere.

I dati da raccogliere riguarderanno informazioni sulle leggi stradali e sulle possibili traiettorie che l'auto potrà intraprendere. Per quanto poco pratico, è possibile collezionare i dati per tutte le regole della strada, in combinazione con tutte le situazioni in cui il veicolo si può trovare. Più problematico è invece il fatto che non si possono prevedere tutte le possibili traiettorie dell'auto, rendendo in questo senso il training set incompleto.

Conseguentemente, durante le fasi di testing, il modello non si muove esattamente nella direzione prevista dal training set. Avviene un fenomeno chiamato covariate shift, che consiste in uno spostamento che diverge da quello previsto, portando il modello a degli stati diversi rispetto alla fase di addestramento. L'errore che il sistema commette aumenta ad ogni passo, portandolo ad affrontare nuove situazioni per le quali non è stato addestrato, rendendo la possibilità di incorrere in decisioni erronee molto più alta. Essendo nell'ambito della guida di veicoli, questo errore non è ammissibile nel caso di utilizzo nel mondo reale. Si può vedere una rappresentazione grafica del covariate shift nella figura 4.1.

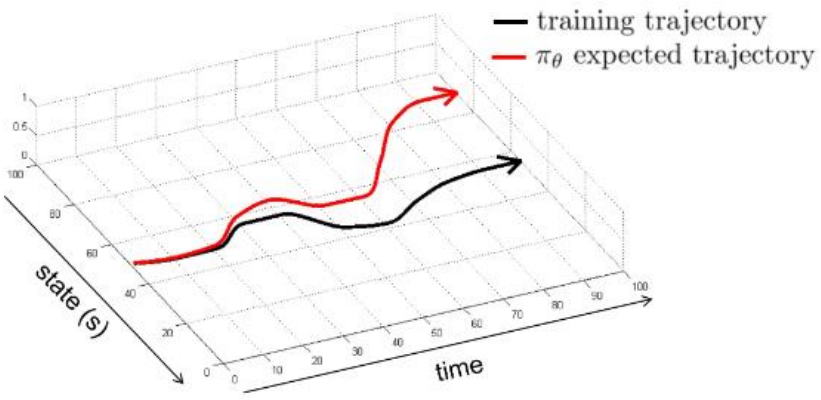


Figura 4.1: Rappresentazione grafica del covariate shift, che mostra che la traiettoria tra la fase di addestramento e quella di test è diversa e come questa divergenza aumenti nel tempo.

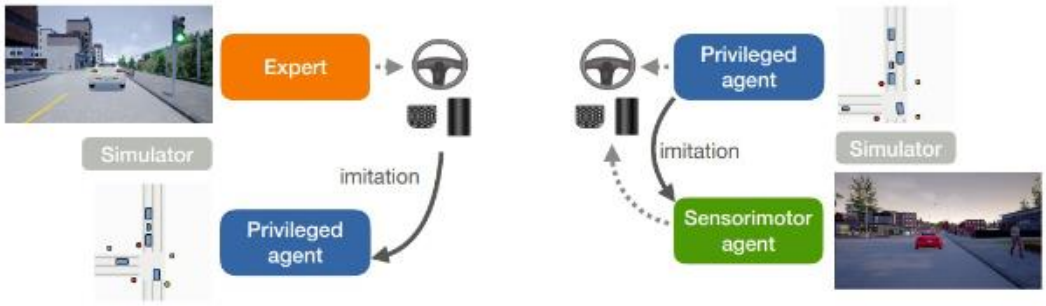
Al fine di ridurre questo scostamento Ross e Bagnell proposero DAgger [78], che va a correggere le azioni andando a collezionare iterativamente dati che modificano le traiettorie errate, coerentemente sempre alle leggi stradali. Il covariate shift non è però l'unica limitazione che l'apprendimento tramite behavior cloning presenta.

Utilizzare dei sensori per comprendere l'ambiente circostante non sempre è sufficiente per far prendere al veicolo delle decisioni ottimali. Ci sono situazioni in cui le informazioni fornite dai sensori mostrano che più di una possibilità è valida, come nel caso degli incroci stradali, in cui più direzioni costituiscono un'azione corretta e sicura, ma non tutte sono adeguate in una determinata situazione. Se non si specificano condizionamenti di alcun tipo relativi all'obiettivo finale, il sistema si troverà in una condizione in cui non è in grado di decidere quale direzione intraprendere senza l'intervento umano.
L'idea è quella di andare a delegare questo compito a dei sistemi di navigazione di alto livello, che rappresentano l'intenzione del passeggero, senza che debba però esprimere una preferenza durante la guida, quindi preservando il concetto di guida autonoma.

Codevilla [79] esplorò ulteriori limitazioni del metodo di apprendimento behavior cloning nell'ambito della guida autonoma soprattutto relative alle performance riguardanti la generalizzazione dei dati. Codevilla ha osservato infatti che rispetto ai tipici metodi di apprendimento supervisionato, la generalizzazione data dal behavior cloning non scala con l'aumento delle dimensioni dei dati nel training set. Come ulteriore limitazione ha identificato una variazione significativa andando a modificare l'ordine dei dati a cui il sistema veniva sottoposto o anche andando a modificare l'inizializzazione del modello.

Il metodi di behavior cloning può essere semplificato andando a decomporlo in due fasi, come introdotto da Chen [80] tramite quello che è stato denominato "Learning by cheating", ovvero apprendimento tramite imbroglio, di cui si può vedere una rappresentazione nella figura 4.2. Quello che si intende con questo termine è il fatto che durante la prima fase si va ad addestrare un agente fornendogli l'accesso a informazioni privilegiate, che gli permettono di "imbrogliare" e quindi di ottenere dati effettivi sull'ambiente circostante senza che debba essere obbligato ad ottenerli tramite sensori. Questo agente, quindi, ottiene l'accesso diretto ai dati del mondo reale, compresa anche la posizione di tutti gli elementi presenti, come altre automobili e pedoni. La seconda fase consiste nell'utilizzo dell'agente privilegiato come insegnante durante l'addestramento del sistema che si basa solo sui dati ricavati dai sensori, ottenendo quindi un agente che non ha accesso a nessuna informazione privilegiata e che quindi non "imbroglia" in alcun modo. Un sistema addestrato in questo modo è in grado di ottenere buoni risultati, superando allo stato dell'arte il benchmark del simulatore CARLA, di cui si parlerà successivamente.

Figura 4.2: Rappresentazione grafica del metodo "Learning by cheating": a sinistra (a) mostra l'agente privilegiato che accede alle informazioni effettive



(a) Privileged agent imitates the expert (b) Sensorimotor agent imitates the privileged agent

dell'ambiente esterno, mentre a destra (b) è presente l'agente che apprende per mezzo dell'agente privilegiato.

4.2.2 Reinforcement Learning

Il reinforcement learning è un approccio che permette a un sistema di apprendere un comportamento tramite ripetute interazioni di tipo "trial and error", ovvero facendo tentativi e incorrendo in errori [81]. Inizialmente l'agente conosce l'obiettivo dell'apprendimento ma non sa come raggiungerlo, dato che il training set utilizzato durante l'addestramento non è etichettato e non ha nessuna conoscenza pregressa. Per questa ragione nel reinforcement learning l'agente deve imparare dall'esperienza. Ogni azione che viene compiuta, fornisce all'agente un feedback: la qualità di un'azione è data da un valore numerico chiamato reward, ovvero una ricompensa, che ha lo scopo di incoraggiare i comportamenti corretti e penalizzare invece quelli errati [82].

Nell'ambito della guida autonoma il sistema apprende le azioni che deve compiere cercando di massimizzare i reward mentre interagisce con l'ambiente circostante. Prendendo decisioni corrette il modello ottiene un

reward positivo che gli permetterà di comprendere che quella compiuta era effettivamente l'azione ottima. Il reward nella guida autonoma è coerente con l'obiettivo che si sta perseguendo, come ad esempio la metà che si vuole raggiungere, oltre ovviamente cercando di evitare situazioni pericolose addestrandolo perché sia in grado di rispettare le leggi stradali.

Essendo un apprendimento basato come detto in precedenza su iterazioni di tipo "trial and error", è necessario addestrare il sistema per mezzo di simulazioni.

Un simulatore tipicamente utilizzato per l'apprendimento tramite per la guida autonoma è CARLA, introdotto da A. Dosovitskiy [83].

Figura 4.3: Una strada presente in una città virtuale presente sul simulatore CARLA, vista nelle quattro possibili condizioni metereologiche [80].

CARLA è un simulatore open-source che mette a disposizione assetti urbani che sono stati creati proprio con lo scopo di creare un ambiente virtuale nel quale addestrare sistemi per l'apprendimento della guida

autonoma, senza vincolare però l'approccio da utilizzare. Un esempio degli scenari che sono presenti nel simulatore CARLA è mostrato in figura 4.3.

L'utilizzo di CARLA ha permesso a Dosovitskiy di osservare il diverso comportamento di un sistema addestrato tramite diverse metodologie. In generale i modelli addestrati tramite reinforcement learning hanno delle performance significativamente peggiori rispetto a modelli addestrati tramite metodologie di behavior cloning, nonostante nel primo caso venissero utilizzati training set con un maggior numero di dati.

Si possono scegliere due approcci per l'apprendimento tramite reinforcement learning:

- model-free: il problema di un apprendimento senza un modello è principalmente il fatto che è molto inefficiente, in quanto il sistema ha bisogno di una maggiore quantità di dati e un maggior numero di tentativi perché l'apprendimento avvenga con successo.
- model-based: l'agente si basa sulla comprensione che ha dell'ambiente e del modello che crea per rappresentarlo, di modo da utilizzarlo successivamente per l'apprendimento delle leggi stradali. Rispetto agli approcci model-free l'apprendimento è più veloce e meno costoso. Nonostante i vantaggi, i metodi model-base presentato una problematica che deve essere considerata: basandosi appunto su un modello, il rischio è quello che il sistema incorra in errori appena non vengono più utilizzati i dati di addestramento.

  Un modo per risolvere questa sfida che può portare a situazioni pericolose è stato proposto da Henaff, il quale ha introdotto un controllo che incoraggia il sistema a prendere solo le decisioni di cui ha una forte convinzione che siano corrette, minimizzando quindi l'errore.

4.2.3 Metodi combinati

Usare metodi di behavior cloning rende i modelli facili da addestrare, ma richiede ingenti quantità di dati. I metodi di reinforcement learning invece sono più flessibili dal punto di vista dell'apprendimento, in quanto non hanno necessità di un training set etichettato, ma è sufficiente un'esplorazione dell'ambiente in cui si trovano. Questa complessità li rende però molto meno efficienti e poco pratici, in quanto il loro utilizzo è del tutto sicuro in simulazioni, ma non è detto che in una situazione reale siano utilizzabili.

X. Liang [84] introdusse un modo per ridurre la poca efficienza di esplorazione di un ambiente di grandi dimensioni del reinforcement learning. Per prima cosa viene addestrato un modello tramite behavior cloning e successivamente vengono utilizzati i pesi della rete neurale appena addestrata per inizializzare la rete che verrà utilizzata per il reinforcement learning.

G. Li [85] ha osservato che gli approcci tipici tendono a essere molto sensibili agli errori degli insegnanti da cui apprendono e per questo ha proposto OIL, sigla che sta per Observational Imitation Learning. Questo approccio è una variante del behavior cloning che consiste nel selezionare il comportamento ottimale andando a osservare molteplici insegnanti imperfetti.
Questo non solo rende l'apprendimento più veloce, in quanto gli insegnanti sono diversi, ma permette di garantire un'esplorazione dell'ambiente molto più accurata.

Un altro intervento che può essere fatto è quello di non mappare direttamente i pixel delle immagini a delle azioni. Chen [86] presentò un approccio il cui obiettivo è quello di stimare dalle immagini una ridotta quantità di misurazioni interpretabili dagli esseri umani, come l'angolazione del veicolo rispetto alla strada, la distanza dalle altre automobili o dalle linee presenti sulla carreggiata. Queste informazioni ottenute verranno poi interpretate dal sistema, che prenderà una decisione che causerà la conseguente azione compiuta dall'auto. Il vantaggio nell'utilizzo di rappresentazioni intermedie è che la rete neurale può essere addestrata e

validata prima dell'effettivo deployment. Inoltre il modello è più facile da addestrare mediante rappresentazioni dei dati intermedie di questo tipo, rendendo le soluzioni trovate anche più interpretabili dagli esseri umani rispetto che nel caso in cui vengano utilizzati metodi tradizionali. Questo approccio è stato tentato da Chen nel simulatore di gare automobilistiche TORCS, del quale si parlerà più nel dettaglio nella sezione 4.3, mentre è stato utilizzato sul simulatore CARLA da Sauer [87], per testare questo approccio sotto condizioni più complicate. In questo caso, infatti, l'agente dovrà anche rispettare le leggi stradali, che comprendono i limiti di velocità, i semafori ed evitare la collisione con i vari elementi che possono essere presenti.

4.3 Dalla simulazione al mondo reale

Una importante limitazione che il reinforcement learning discusso in precedenza presenta è la necessità che l'apprendimento del modello per la guida autonoma avvenga in delle simulazioni.

In fase di training quindi i modelli vengono addestrati su dati che simulano le situazioni stradali del mondo reale, ma è evidente che una simulazione non possa fornire tutte le casistiche possibili perché il modello venga correttamente addestrato. Il rischio è quindi quello che, nonostante buoni risultati in fase di training, i dati non vengano generalizzati a sufficienza e che quindi sia impossibile poi trasportare il sistema nel mondo reale.

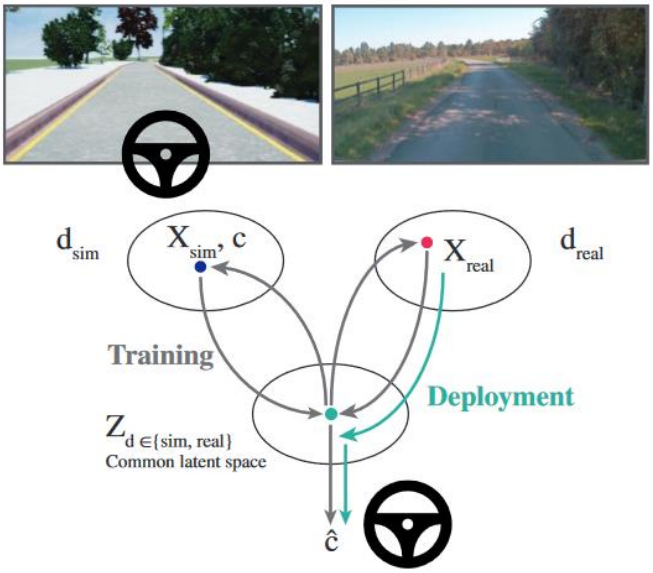
I modelli devono avere quindi la capacità di tradurre le immagini simulate in immagini appartenenti al mondo reale. Questo viene fatto andando prima a segmentare le immagini virtuali del simulatore e poi traducendo queste immagini segmentate in quelle reali.

Bewley [88] introdusse l'addestramento basato sulla traduzione image-to-image, che consiste nel trasportare le leggi stradali dalla simulazione al mondo reale. Bewley ha addestrato una rete neurale profonda per renderla in grado di guidare in un ambiente simulato, dove è possibile dare una conoscenza completa del mondo, e poi l'ha adattata per fare in modo di

riconoscere la variazione delle immagini che avrebbe sperimentato nel mondo reale.

Figura 4.4: Mostra il modello di apprendimento di Bewley, in cui le leggi stradali vengono apprese dallo spazio latente comune Z, indipendente dalla fonte da cui arriva l'immagine.

Il punto chiave di questo addestramento non è la sola traduzione image-to-image, ma è il fatto di assicurarsi che le leggi stradali siano invariate a prescindere dal fatto che l'immagine sottoposta al sistema sia reale o



simulata. Per questa ragione in fase di training le informazioni relative alle leggi stradali e al modo corretto di agire vengono tratte da uno spazio comune latente, come rappresentato nella figura 4.4.

4.4 Dataset utilizzati

L'utilizzo di metodi di apprendimento end-to-end per sistemi relativamente alla guida autonoma sta diventando sempre più diffuso e per questa ragione sono necessari dati da utilizzare ai fini di addestrare correttamente i modelli.

Un dataset molto utilizzato è comma.ai [89], che mette a disposizione più di 7 ore di video di guida in autostrada. Non sono contenuti solo i video di guida ma anche dati di alcuni sensori, come quello della velocità, i dati del GPS e quelli del giroscopio. Il problema principale di questo dataset è però che i video catturati rappresentano solo scenari autostradali, quindi non sono sufficienti per addestrare un modello nel caso in cui si voglia utilizzati in ambienti complicati, come nel caso di guida in città.

Un altro dataset, molto più completo di comma.ai, è Berkeley DeepDrive Video [90], il quale continente circa 10.000 video di guida, per un totale di 1.100 ore, in città, zone rurali e autostrade, in momenti diversi della giornata e con differenti condizioni atmosferiche. Sono anche inclusi dati relativi alla posizione GPS e timestamp. Si possono trarre da questo dataset numerose informazioni, come l'identificazione degli oggetti per strada e individuazione delle aree in cui si può effettivamente guidare. Inoltre, sono presenti molteplici tipi di annotazioni che identificano le linee stradali, che permettono di individuare i passaggi pedonali, le righe delle carreggiate, indicazioni di stop e così via.

Per quanto riguarda i simulatori open-source possiamo identificare i due principali:

- TORCS (The Open Racing Car Simulator) [91]: è uno dei primi simulatori open-source di gare automobilistiche è stato utilizzato per gli addestramenti di sistemi per la guida autonoma con l'obiettivo di

seguire la carreggiata. Purtroppo, è un simulatore che utilizza un ambiente semplificato, nel quale non sono presenti pedoni, incroci di alcun tipo. Per questa ragione TORCS è inadatto per l'addestramento completo di sistemi per la guida autonoma nel mondo reale.

- CARLA: questo simulatore open-source, rispetto a quello trattato in precedenza, mette a disposizione un ambiente più realistico e complesso, che rende possibile l'addestramento di veicoli per la guida autonoma in condizioni urbane. CARLA fornisce immagine di alta qualità e al fine di rendere l'ambiente il più simile possibile a quello reale, presenta zone urbane con diversi layout, comprendendo traffico, edifici, pedoni e segnali stradali. Il simulatore permette anche di addestrare il modello utilizzando differenti condizioni atmosferiche e in diversi momenti della giornata, di modo da avere tutte le possibili luminosità.

Purtroppo, l'utilizzo di dataset o di simulatori per l'addestramento di veicoli per la guida autonoma non forniscono tutte le informazioni necessarie per rendere i sistemi completamente sicuri. Non si riescono infatti a considerare tutte le possibilità e di conseguenza si rischia di incorrere in errori in situazioni del mondo reale, poiché si potrebbero verificare eventi rari per i quali il modello non è stato addestrato.

Il sistema di guida introdotto da Tesla [92] permette di memorizzare passivamente le informazioni che vengono utilizzate nel caso in cui si verifichino rare condizioni di errore, di modo che il modello sia in grado di evitare tali errori. Purtroppo, non ci sono dettagli su come questa tecnologia sia stata sviluppata, poiché i dati sono di proprietà di Tesla e non sono stati rilasciati.

Le informazioni rilasciate da Tesla riguardano il fatto che le reti neurali utilizzati da Tesla vengono addestrate anche su situazioni complicate e considerando i diversi scenari che possono essere incontrati nel mondo,

rielaborando i dati ottenuti da circa 1 milione di veicoli catturati in tempo reale.

Un sistema completamente addestrato coinvolge 48 reti che impiegano 70,000 ore di GPU per essere addestrate, il che significa che con l'utilizzo di una singola GPU l'addestramento sarebbe completato dopo 70,000 ore. Ad ogni passo vengono fatte circa 1.000 predizioni [93].

4.5 Metriche di valutazione

La valutazione di sistemi di apprendimento end-to-end per la guida autonoma non è standardizzata, quindi non esistono dei benchmark utilizzabili per confrontare diversi modelli addestrati.

Quello che si può fare è utilizzare un benchmark come quello fornito da CARLA per valutare sistemi addestrati su un determinato dataset. Questo benchmark si basa sulla misurazione di due metriche: la prima valuta la percentuale di episodi superati con successo, considerando le possibili quattro condizioni fornite da CARLA. La seconda metrica invece misura la distanza media percorsa tra due azioni errate, che includono guidare nella carreggiata sbagliata, impattare con altri veicoli o pedoni e colpire oggetti statici.

Capitolo 5

Conclusione e sviluppi futuri

In questo lavoro sono state illustrate le principali tecnologie e tecniche utilizzate all'interno del settore della guida autonoma, mostrandone i principali limiti e difficoltà. L'identificazione di oggetti in un'immagine è un problema aperto, che vede proporre sempre nuovi metodi e scenari. In particolare, esso è un punto chiave di tecniche più sofisticate che effettuano l'analisi delle azioni e l'associazione di persone alla loro identità. Riuscire ad ottenere algoritmi veloci ed affidabili è un passo imprescindibile per un'analisi più avanzata. È stato organizzato il lavoro in tre capitoli. Il primo ha avuto lo scopo di determinare il contesto, nel secondo si è osservato come viene gestito il problema del riconoscimento degli oggetti mentre nel terzo sono state visionate le più moderne tecniche di visione artificiale all'interno della guida autonoma.

I veicoli a guida autonoma rappresentano una delle tecnologie destinate a rivoluzionare il settore della mobilità a livello globale.

Lo shock causato dalla diffusione del COVID-19 e la conseguente crisi economica globale potrebbero portare ad un momentaneo rallentamento degli investimenti e dello sviluppo di questo tipo di tecnologie, ma nel lungo periodo i veicoli a guida autonoma potrebbero assumere un ruolo molto importante proprio in risposta alla pandemia.

In prospettiva, infatti, i veicoli senza conducente potrebbero avere un importante ruolo per rispondere a nuove esigenze per lo spostamento di persone e merci, derivante dalla pandemia COVID-19. Il problema dell'affollamento dei trasporti pubblici, ad esempio, potrebbe essere parzialmente risolto attraverso l'utilizzo di minibus a guida autonoma prenotabili su richiesta per promuovere il distanziamento sociale, mentre i veicoli a guida autonoma utilizzati per effettuare le spedizioni potrebbero essere utili per soddisfare la domanda di consegna evitando il contatto

fisico. Inoltre, il cambiamento dei bisogni degli utenti e la crescente attenzione all’impatto ambientale nel settore dei trasporti potrebbe accelerare l’adozione di queste tecnologie a livello globale. Si tratta, quindi, di un settore con ampi margini di sviluppo e differenti tipologie di applicazioni, che avrà impatti importanti in diversi ambiti.

Negli ultimi anni sono stati fatti importanti progressi nel campo dei veicoli a guida autonoma. In primo luogo, si è cercato di migliorare il grado di sicurezza ed efficacia, attraverso test su larga scala. In molti paesi, oggi, esistono leggi ad hoc e regolamenti specifici per consentire l'utilizzo delle tecnologie AV. Si stanno compiendo importanti progressi sulle difficili sfide dell'attuazione, compresa la realizzazione di infrastrutture, la definizione di politiche e protocolli relativi ai dati e la definizione di politiche in materia di licenze e assicurazioni.

Le tecnologie per i veicoli a guida autonoma trovano applicazione sia in campo privato (automobili private, servizi di taxi, trasporto condiviso), sia in ambito pubblico (autobus). Vi sono, inoltre, ulteriori opportunità per espandere l'utilizzo dei veicoli a guida autonoma per il trasporto merci e in ambienti chiusi come aree industriali, portuali e minerarie.

Tuttavia, il vero salto di qualità verrà effettuato quando i veicoli senza conducente entreranno concretamente nella nostra quotidianità, migliorando la qualità della nostra vita.

Capitolo 6

Bibliografia

[1] D. Milakis, B. Van Arem and B. Van Wee Journal of Intelligent Transportation Systems, 21 (4) (2017), pp. 324-348 “Policy and society related implications of automated driving: a review of literature and directions for future research”

[2] J.H. Kim, G. Lee, J.Y. Park, J. Hong and J. Park Energy Pol., 132 (2019), pp. 736-743 “Consumer intentions to purchase battery electric vehicles in Korea”

[3] F. Liu, F. Zhao, Z. Liu and H. Hao Energy Pol., 132 (2019), pp. 462-473 “Can autonomous vehicle reduce greenhouse gas emissions? A country-level evaluation”

[4] NSC, 2017. 2017 “Estimates Show Vehicle Fatalities Topped 40,000 for Second Straight Year”. Retrieved from. National Safety Council

[5] “Transition to manual: driver behaviour when resuming control from a highly automated vehicle” N. Merat, A.H. Jamson, F.C. Lai, M. Daly and O.M. Carsten Transp. Res. Part F Traffic Psychol. Behav., 27 (2014), pp. 274-282

[6] “50th anniversary invited article—autonomous vehicles and connected vehicle systems: flow and operations considerations” H.S. Mahmassani Transp. Sci., 50 (4) (2016), pp. 1140-1162

[7] “Autonomous Vehicle Implementation Predictions” T. Litman Victoria Transport Policy Institute., Canada (2017)

[8] J.H. Andrew - 2019 “Waymo’s Driverless Cars Hit a New Milestone: 10 Million Miles on Public Roads”

https://www.theverge.com/2018/10/10/17958276/waymo-self-driving-cars-10-million-miles-challenges

[9] K. Freund - 2019 “NVIDIA VR + AI = Billions of Miles of Virtual Driving. Forbes”

https://www.forbes.com/sites/moorinsights/2018/03/27/nvidia-vr-ai-billions-of-miles-of-virtual-driving/#25a1f42d438a

[10] W. Pavia – 2018 “Driverless Uber Car ‘not to Blame’ for Woman’s Death”

https://www.thetimes.co.uk/article/driverless-uber-car-not-to-blame-for-woman-s-death-klkbt7vf0

[11] Tesla – 2016 “A Tragic Loss”

https://www.tesla.com/blog/tragicloss?utm_campaign=Blog_063016&utm_source=Twitter&utm_medium=social

[12] California Department of Motor Vehicles – 2018 “Autonomous Vehicle Disengagement Reports”

[13] C. Lv, D. Cao, Y. Zhao, D.J. Auger, M. Sullman, H. Wang and A. Mouzakitis IEEE/CAA J. Autom. Sin., 5 (1) (2017), pp. 58-68 “Analysis of autopilot disengagements occurring during autonomous vehicle testing”

[14] Sean Campbell, Niall O’ Mahony, Lenka Krpalcova, Daniel Riordan, Joseph Walsh “Sensor Technology in Autonomous Vehicles” IMaR Technology Gateway Institute of Technology Tralee Tralee, Ireland; Biocomputing and Developmental Systems Research Group University of Limerick Limerick, Ireland

[15] Joel Janai, Fatma Guney, Aseem Behl, Andreas Geiger. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art”

Autonomous Vision Group, MPI for Intelligent Systems and University of Tubingen, Germany; College of Engineering, Koc University, Turkey (December 18, 2019) (pag 23)

[16] “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art” (vedi [15] per dettagli) pag (11-12)

[17] C. Ilas, "Perception in autonomous ground vehicles", International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2013.

[18] Jri Lee, Yi-An Li, Meng-Hsiung Hung, Shih-Jou Huang, "A FullyIntegrated 77-GHz FMCW Radar Transceiver in 65-nm CMOS Technology", IEEE Journal of Solid-State Circuits 2010.

[19] H. Rohling and C. Moller, “Radar waveform for automotive radar systems and applications,” in 2008 IEEE Radar Conference, 2008, pp. 1–4.

[20] L. Xiaoming, Qin Tian, Chen Wanchun, and Y. Xingliang, “Real-time distance measurement using a modified camera”, in 2010 IEEE Sensors Applications Symposium (SAS), 2010, pp. 54-58

[21] W. Rahiman, Z. Zainal, "An Overview of Development GPS Navigation for Autonomous Car", 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2013.

[22] D. Xu, A. Jain, and D. Anguelov, “PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation, Proc. of CVPR, 2018.

[23] F. Iandola and K. Keutzer, "Keynote: small neural nets are beautiful: enabling embedded systems with small deep-neuralnetwork architectures," 2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Seoul, 2017, pp. 1-10.

[24] S. Oh and H. Kang, "Fast Occupancy Grid Filtering Using Grid Cell Clusters From LIDAR and Stereo Vision Sensor Data," in IEEE Sensors Journal, vol. 16, no. 19, pp. 7258-7266, Oct.1, 2016

[25] R. O. Chavez-Garcia and O. Aycard, "Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2, pp. 525-534, Feb. 2016.

[26] F. Iandola and K. Keutzer, "Keynote: small neural nets are beautiful: enabling embedded systems with small deep-neuralnetwork architectures," 2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Seoul, 2017, pp. 1-10.

[27] Qilong Zhang and R. Pless. “Extrinsic calibration of a camera and

laser range finder". In: Proc. IEEE International Conf. on Intelligent

Robots and Systems (IROS). 2004.

[28] Janne Heikkila and Olli Silven. “A Four-step Camera Calibration Procedure with Implicit Image Correction". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 1997.

[29] Henrik Andreasson and Achim J. Lilienthal. \6D scan registration using depth-interpolated local image features". In: Robotics and Autonomous Systems (RAS). 2010.

[30] Lionel Heng, Bo Li, and Marc Pollefeys. “CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry". In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS). 2013.

[31] Daniel Scharstein and Richard Szeliski. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: International Journal of Computer Vision (IJCV) 47 (2002), pp. 7-42.

[32] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge". In: International Journal of Computer Vision (IJCV) 88.2 (2010), pp. 303-338.

[33] “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art” (vedi [15] per dettagli) pag (29-30)

[34] Frederic Suard, Alain Rakotomamonjy, Abdelaziz Bensrhair, and Alberto Broggi. “Pedestrian detection using infrared images and histograms of oriented gradients". In: Proc. IEEE Intelligent Vehicles Symposium (IV). IEEE. 2006, pp. 206-212.

[35] “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art” (vedi [15] per dettagli) pag (45-46)

[36] P. A. Viola, M. J. Jones, and D. Snow. “Detecting pedestrians using patterns of motion and appearance". In: International Journal of Computer Vision (IJCV) 63(2) (2005), pp. 153-161.

[37] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2005.

[38] M. Enzweiler and D. M. Gavrila. “Monocular Pedestrian Detection: Survey and Experiments". In: IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 31 (2009), pp. 2179-2195.

[39] Rodrigo Benenson, Mohamed Omran, Jan Hendrik Hosang, and Bernt Schiele. “Ten Years of Pedestrian Detection, What HaveWe Learned?" In: Proc. of the European Conf. on Computer Vision (ECCV). 2014.

[40] Pedro F. Felzenszwalb, David A. McAllester, and Deva Ramanan. “A discriminatively trained, multiscale, deformable part model". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2008.

[41] B. Leibe, A. Leonardis, and B. Schiele. “Robust Object Detection with Interleaved Categorization and Segmentation". In: International Journal of Computer Vision (IJCV) 77.1-3 (2008), pp. 259-289.

[42] Rodrigo Benenson, Mohamed Omran, Jan Hendrik Hosang, and Bernt Schiele. “Ten Years of Pedestrian Detection, What HaveWe Learned?" In: Proc. of the European Conf. on Computer Vision (ECCV). 2014.

[43] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. “Pedestrian Detection with Unsupervised Multi-stage Feature Learning". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2013.

[44] Ross Girshick, Je Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2014.

[45] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. “Selective search for object recognition". In: International Journal of Computer Vision (IJCV) 104.2 (2013), pp. 154-171.

[46] Ross B. Girshick. “Fast R-CNN". In: Proc. of the IEEE International Conf. on Computer Vision (ICCV). 2015.

[47] Pierre Sermanet, David Eigen, Xiang Zhang, Micha•el Mathieu, Rob Fergus, and Yann LeCun. “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks". In: Proc. of the International Conf. on Learning Representations (ICLR). 2014.

[48] M.D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks”, Proceedings of theEuropean Conference on Computer Vision 2014.

[49] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, No. 1, pp. 142-158, Jan. 1 2016.

[50] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,”IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2016.2577031.

[51] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. “The German Traffic Sign Recognition Benchmark: A multi-class classification competition". In: International Joint Conference on Neural Networks (IJCNN). 2011, pp. 1453-1460.

[52] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. “Detection of traffic signs in real-world images: The German traffic sign detection benchmark". In: International Joint Conference on Neural Networks (IJCNN). 2013, pp. 1-8.

[53] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. “Traffic-sign detection and classification in the wild". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2016, pp. 2110-2118.

[54] Dan C Ciresan, Ueli Meier, Jonathan Masci, and Jourgen Schmidhuber. “A committee of neural networks for traffic sign classification". In: International Joint Conference on Neural Networks (IJCNN). 2011, pp. 1918-1921.

[55] Pierre Sermanet and Yann LeCun. “Traffic sign recognition with multiscale Convolutional Networks." In: International Joint Conference on Neural Networks (IJCNN). 2011, pp. 2809-2813.

[56] Hamed Habibi Aghdam, Elnaz Jahani Heravi, and Domenec Puig. “A practical approach for detection and classification of traffic signs using Convolutional Neural Networks". In: Robotics and Autonomous Systems (RAS) 84 (2016), pp. 97-112.

[57] Alvaro Arcos Garcia, Juan Antonio Alvarez-Garcia, and Luis Miguel Soria-Morillo. “Evaluation of deep neural networks for traffic sign detection systems". In: Neurocomputing 316 (2018), pp. 332-344.

[58] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: Advances in Neural Information Processing Systems (NIPS). 2015.

[59] Sergey Io e and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: Proc. of the International Conf. on Machine learning (ICML). 2015.

[60] M.Z. Zia, M. Stark, B. Schiele, and K. Schindler. “Detailed 3D Representations for Object Recognition and Modeling". In: IEEE Trans. On Pattern Analysis and Machine Intelligence (PAMI) 35.11 (Nov. 2013), pp. 2608-2623.

[61] Bojan Pepik, Michael Stark, Peter V. Gehler, and Bernt Schiele. “Multi View and 3D Deformable Part Models". In: IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 37.11 (2015), pp. 2232-2245.

[62] Pedro F. Felzenszwalb, David A. McAllester, and Deva Ramanan. “A discriminatively trained, multiscale, deformable part model". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2008.

[63] Abhijit Kundu, Yin Li, and James M. Rehg. \3D-RCNN: Instance-Level 3D Object Reconstruction via Render-and-Compare". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2018, pp. 3559-3568.

[64] Jason Ku, Alex D. Pon, and Steven L. Waslander. “Monocular 3D Object Detection Leveraging Accurate Proposals and Shape

Reconstruction". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019).

[65] Zhaowei Cai, Quanfu Fan, Rogerio Schmidt Feris, and Nuno Vasconcelos. “A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection". In: Proc. of the European Conf. on Computer Vision (ECCV). 2016.

[66] Bo Li, Tianlei Zhang, and Tian Xia. “Vehicle Detection from 3D Lidar Using Fully Convolutional Network". In: Proc. Robotics: Science and Systems (RSS). 2016.

[67] Dominic Zeng Wang and Ingmar Posner. “Voting for Voting in Online Point Cloud Object Detection". In: Proc. Robotics: Science and Systems (RSS). 2015.

[68] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. \Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks". In: Proc. IEEE International Conf. on Robotics and Automation (ICRA). 2017, pp. 1355-1361.

[69] Jason Ku, Melissa Mozian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. “Joint 3D Proposal Generation and Object Detection from View Aggregation". In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS). 2018, pp. 1-8.

[70] Jia Deng, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Li Fei-fei. “Imagenet: A large-scale hierarchical image database". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2009.

[71] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge". In: International Journal of Computer Vision (IJCV) 88.2 (2010), pp. 303-338.

[72] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. “Microsoft

COCO: Common Objects in Context". In: Proc. of the European Conf. On Computer Vision (ECCV). 2014.

[73] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2012.

[74] Piotr Dollar, ChristianWojek, Bernt Schiele, and Pietro Perona. “Pedestrian Detection: An Evaluation of the State of the Art". In: IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 34.4 (2012), pp. 743-761.

[75] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu M. Gavrila. “The EuroCity Persons Dataset: A Novel Benchmark for Object Detection". In: IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) (2019).

[76] Dean Pomerleau. “ALVINN: An Autonomous Land Vehicle in a Neu-ral Network”. In:Advances in Neural Information Processing Systems(NIPS). 1988, pp. 305–313.

[77] Dean Pomerleau and Todd Jochem.Look, Ma, No Hands. https://www.cmu.edu/news/stories/archives/2015/july/look-ma-no-hand.html. Online: accessed 18-June-2019. 2015

[78] Stephane Ross and Drew Bagnell. “Efficient Reductions for Imita-tion Learning”. In:Conference on Artificial Intelligence and Statistics(AISTATS).

[79] Felipe Codevilla, Eder Santana, Antonio M. L ´opez, and Adrien Gaidon.“Exploring the Limitations of Behavior Cloning for Autonomous Driv-ing”. In:arXiv.orgabs/1904.08980 (2019).

[80] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Kr ¨ahenb ¨uhl. “Learning by Cheating”. In:Proc. Conf. on Robot Learning (CoRL).2019.

[81] Mathworks“ Reinforcement Learning”. https://it.mathworks.com/discovery/reinforcement-learning.html

[82] Andrea Minini. “L’apprendimento con rinforzo”. http://www.andreaminini.com/ai/machine-learning/apprendimento-con rinforzo

[83] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez,and Vladlen Koltun. “CARLA: An Open Urban Driving Simulator”.In:Proc. Conf. on Robot Learning (CoRL). 2017.

[84] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. “CIRL:Controllable Imitative Reinforcement Learning for Vision-based Self-driving”. In:arXiv.orgabs/1807.03776 (2018).

[85] Guohao Li, Matthias Mueller, Vincent Casser, Neil Smith, Dominik L.Michels, and Bernard Ghanem. “Teaching UAVs to Race With Obser-vational Imitation Learning”. In:arXiv.orgabs/1803.01129 (2018).

[86] Chenyi Chen, Ari Seff, Alain L. Kornhauser, and Jianxiong Xiao.“DeepDriving: Learning Affordance for Direct Perception in AutonomousDriving”. In:Proc. of the IEEE International Conf. on Computer Vi-sion (ICCV). 2015, pp. 2722–2730.

[87] Axel Sauer, Nikolay Savinov, and Andreas Geiger. “Conditional Affor-dance Learning for Driving in Urban Environments”. In:Proc. Conf.on Robot Learning (CoRL). 2018.

[88] Alex Bewley, Jessica Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen,Vinh Dieu Lam, and Alex Kendall. “Learning to Drive from Simulationwithout Real World Labels”. In:arXiv.orgabs/1812.03823 (2018).

[89] Commaai. “Comma.ai research”. https://research.comma.ai/

[90] Berkeley DeepDrive. https://bdd-data.berkeley.edu/

[91] SourceForge. “TORCS - The Open Racing Car Simulator”. https://sourceforge.net/projects/torcs/

[92] Tesla. “Introducing Software Version 9.0”. https://www.tesla.com/blog/introducing-software-version-9?redirect=no

[93] Tesla. “Tesla Autopilot AI”. https://www.tesla.com/autopilotAI?redirect=no