

Progetto SmartPark@Lucca

D4. - Progettazione e realizzazione software di riconoscimento visuale parcheggi

Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle
Ricerche (ISTI-CNR)
Via G. Moruzzi, 1
56124 Pisa

Autori: Giuseppe Amato, Paolo Bolettieri, Fabio Carrara, Luca Ciampi,
Claudio Gennaro, Giuseppe Riccardo Leone, Davide Moroni, Gabriele
Pieri, Claudio Vairo

Sommario

In questo deliverable sono descritte le attività eseguite all'interno del WP2, in particolare relative al Task 2.3 - Realizzazione SW.

Indice

1 Introduzione	4
2 Classificazione singoli stalli tramite Smart Camera con reti neurali	5
2.1 Realizzazione dataset	5
2.2 Realizzazione rete neurale	6
2.3 Implementazione algoritmo	7
3 Classificazione singoli stalli tramite Smart Embedded Cameras	9
4 Conteggio parcheggi globale	12
4.1 Introduzione e motivazioni	12
4.2 Realizzazione della rete neurale	13
4.3 Implementazione dell'algoritmo per il conteggio automatico	14

1 Introduzione

In questo documento descriviamo la progettazione e la realizzazione del software per il monitoraggio visuale del parcheggio.

A tale scopo, abbiamo progettato e realizzato due approcci leggermente diversi, di cui uno è stato implementato utilizzando due soluzioni hardware e software complementari. Più precisamente, il primo approccio si basa sulla classificazione dello stato di occupazione dei singoli stalli del parcheggio e aggrega queste informazioni per ottenere lo stato di occupazione globale del parcheggio. Il secondo approccio invece si basa direttamente sul conteggio globale delle auto presenti nel parcheggio analizzando l'immagine nel suo insieme.

Del primo approccio presentiamo due diverse implementazioni: una basata su smart cameras dotate di sufficienti risorse hardware e sufficiente capacità di calcolo da eseguire algoritmi che sfruttano reti neurali, e una basata su embedded smart cameras dalle risorse e capacità di calcolo più limitate che sfrutta l'analisi del segnale dell'immagine catturata dalla camera per determinare lo stato di occupazione dei singoli stalli.

Tutte le soluzioni proposte sono state progettate e implementate per essere eseguite completamente a bordo delle smart cameras. All'esterno della rete viene trasmessa solo l'informazione testuale sullo stato di occupazione dei parcheggi per fini di visualizzazione.

2 Classificazione singoli stalli tramite Smart Camera con reti neurali

In questa sezione presentiamo l'approccio usato per realizzare il software di monitoraggio dello stato di occupazione degli stalli del parcheggio installato all'interno delle smart cameras e basato su reti neurali.

In particolare, presentiamo prima il dataset che abbiamo costruito al fine di addestrare la rete neurale, successivamente descriviamo la rete neurale che abbiamo realizzato per riconoscere lo stato di occupazione degli stalli e infine descriviamo l'algoritmo che viene eseguito a tempo di esecuzione per classificare i singoli stalli del parcheggio.

2.1 Realizzazione dataset

Al fine di addestrare la nostra rete neurale, abbiamo costruito un dataset di immagini del parcheggio dell'area di ricerca del CNR di Pisa, composto da 164 posti auto. Questo dataset, chiamato CNRPark-EXT [1], contiene circa 150.000 immagini etichettate di posti auto vuoti e occupati.

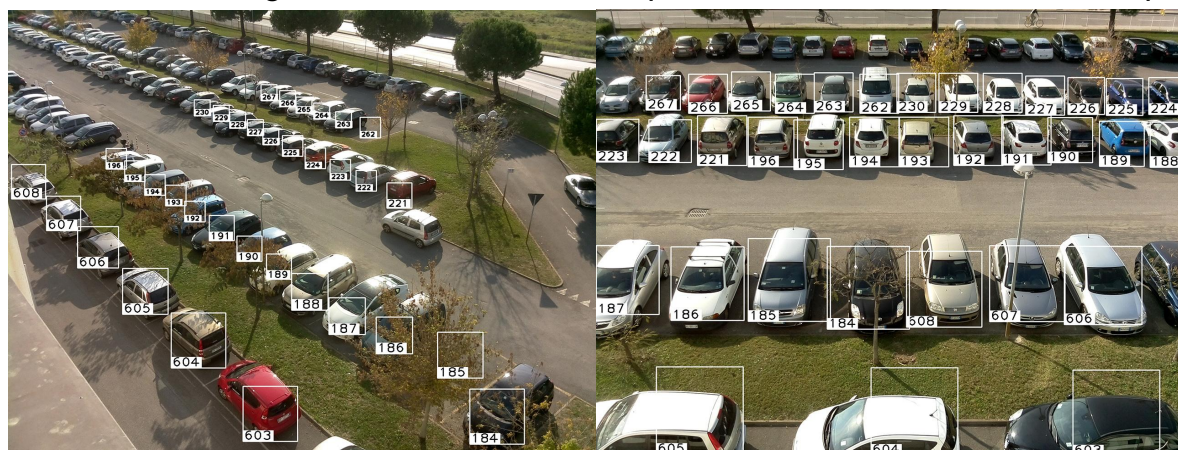


Figura 1: Esempi di immagini del dataset.

CNRPark-EXT contiene immagini raccolte in diversi giorni e in varie condizioni climatiche (soleggiato, nuvoloso e piovoso) da 9 telecamere con diverse prospettive e angoli di vista (vedere Figura 1 per alcuni esempi).

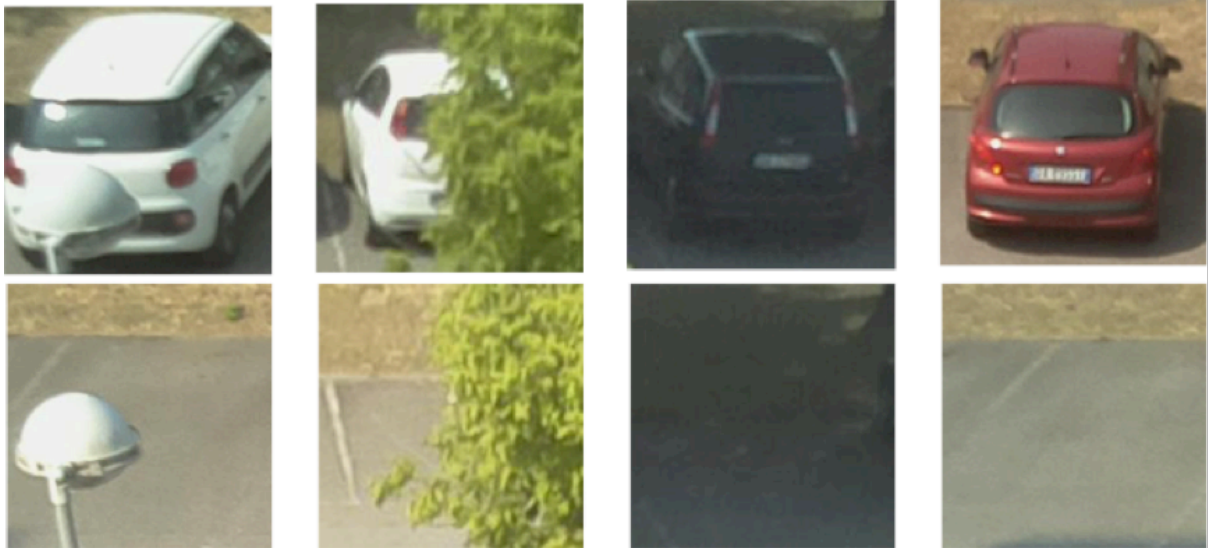


Figura 2: Esempi di patch contenenti stalli singoli del parcheggio. Si possono notare i problemi di occlusione e ombra presenti nel dataset.

CNRPark-EXT comprende situazioni con diverse condizioni luminose e include problematiche di occlusione parziale a causa di ostacoli (alberi, lampioni, altre auto) e auto parzialmente o globalmente ombreggiate (vedere Figura 2). Ciò consente di addestrare un classificatore in grado di distinguere la maggior parte delle situazioni difficili che si possono trovare in uno scenario reale.

Per ogni smart camera, abbiamo creato manualmente delle maschere, cioè delle Regioni di Interesse (ROI), che consentono di ritagliare le immagini riprese dalla fotocamera in immagini più piccole, ciascuna contenente un unico spazio di parcheggio. Nel resto del documento ci riferiamo a queste immagini più piccole come *patch* (alcuni esempi sono riportati in Figura 2). Una patch è un quadrato di dimensioni proporzionale alla distanza dalla fotocamera, le patch più vicine sono più grandi di quelle più lontane. Infine, abbiamo etichettato manualmente tutte le patch in base allo stato di occupazione del parcheggio corrispondente, 0 per vacante e 1 per occupato.

2.2 Realizzazione rete neurale

Per realizzare la nostra rete neurale che classifica lo stato di occupazione di uno stallo del parcheggio, abbiamo usato la famosa Convolutional Neural Network (CNN) AlexNet [2], e ne abbiamo creato una versione più compatta e leggera che possa essere installata ed eseguita su un dispositivo con risorse limitate come le nostre smart cameras. L'abbiamo chiamata mAlexNet.

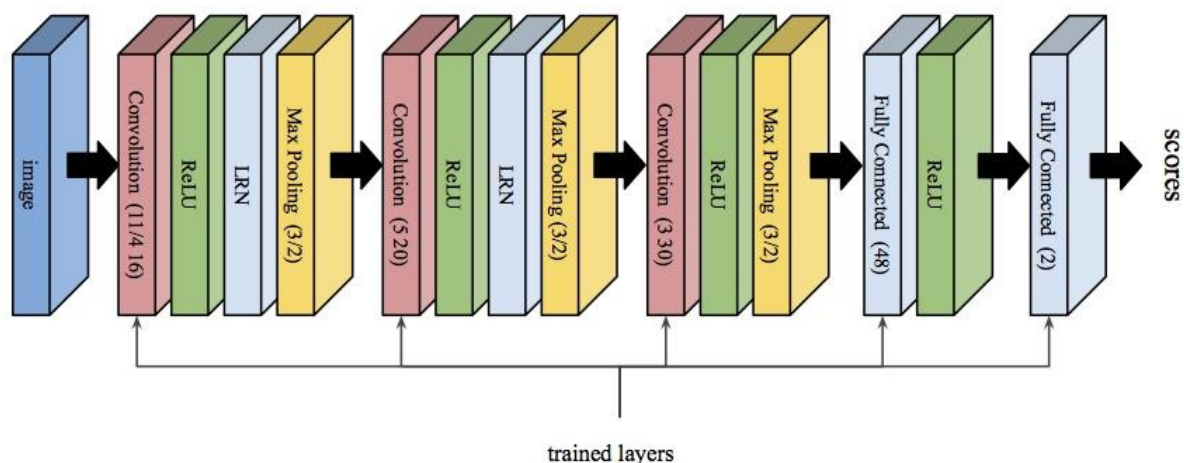


Figura 3: struttura della rete mAlexNet.

La mAlexNet è composta da tre livelli convoluzionali e due livelli completamente connessi (vedi Figura 3), ed è stata addestrata utilizzando il dataset di immagini del parcheggio descritto nella Sezione 2.1. Anche la dimensione dei parametri per ciascun livello convoluzionale è stata notevolmente ridotta rispetto alla AlexNet originale. L'addestramento della rete è stato eseguito su un server dotato di una scheda video Nvidia K40 e ha richiesto alcune ore di training. Una volta allenata tuttavia, la rete è molto veloce a classificare un singolo stalli. Infatti su un Raspberry Pi 2 model B impiega circa 15 secondi a classificare 54 stalli, che è il maggior numero di stalli inclusi nell'angolo di visione di una singola camera.

2.3 Implementazione algoritmo

Il sistema implementato per il riconoscimento dello stato di occupazione degli stalli del parcheggio sfrutta una rete di smart cameras in cui ciascuna camera monitora l'insieme di stalli inclusi nella porzione di parcheggio vista dalla telecamera. Il sistema comprende anche un server che riceve le informazioni sull'occupazione degli stalli del parcheggio e visualizza queste informazioni.

Per ogni smart camera, abbiamo definito la regione di interesse (ROI) per tutti gli stalli inclusi nel campo visivo di quella telecamera. Questa operazione viene eseguita manualmente una volta prima della distribuzione del sistema. In fase di esecuzione, utilizziamo queste ROI per ritagliare l'immagine acquisita dalla smart camera ed ottenere un insieme di patches che comprende i singoli stalli del parcheggio. Ogni patch viene quindi fornita in input alla nostra rete mAlexNet che la elabora e fornisce lo stato di occupazione dello stalli analizzato, associato al valore di confidenza della

classificazione effettuata. Ogni smart camera quindi invia periodicamente al server centrale la lista degli id degli stalli monitorati con lo stato di occupazione rilevato e il valore di confidenza corrispondente.

In alcuni casi, più telecamere monitorano lo stesso stallo del parcheggio. Questa ridondanza viene sfruttata per risolvere eventuali problemi di occlusione (alberi, altre macchine, ostacoli). In particolare, abbiamo assegnato un peso a ciascuna coppia <stallo, smart camera>, che rappresenta quanto è buona la vista di quello stallo visto da quella smart camera. In fase di esecuzione, la confidenza della previsione restituita dalle smart cameras che monitorano lo stesso stallo, è pesata con questo valore e in caso di previsioni contrastanti (una libera e una occupata), viene presa per buona la previsione corrispondente alla confidenza pesata più elevata.

Tutta l'elaborazione viene eseguita a bordo della smart camera. Nessuna immagine viene inviata all'esterno. L'unica informazione trasmessa al server è lo stato di occupazione per ogni parcheggio. Questo preserva la privacy delle targhe e riduce considerevolmente la larghezza di banda necessaria per l'esecuzione del software di monitoraggio visuale del parcheggio.

3 Classificazione singoli stalli tramite Smart Embedded Cameras

Come descritto nel Deliverable 3 paragrafo 3.5 anche in zone urbane la connettività elettrica a volte può essere del tutto assente o particolarmente laboriosa. In questi casi si può prevedere l'installazione contestuale nei nodi di monitoraggio di un pacco batteria ed un pannello fotovoltaico che garantiscono il funzionamento completamente autonomo dei nodi stessi.

Per garantire una estrema compattezza ed una elevata autonomia questo secondo tipo di telecamera intelligente utilizza una scheda di elaborazione a basso consumo energetico completamente progettata e realizzata dal laboratorio Segnali ed Immagini dello ISTI-CNR di Pisa.

Questa scheda non possiede le caratteristiche di potenza e memoria necessarie per l'utilizzo del metodo presentato nel paragrafo precedente: abbiamo dunque sviluppato algoritmi di visione artificiale specifici per la problematica in esame.

Per ridurre le zone da analizzare si prendono in considerazione solo determinate regioni di interesse (in inglese RoI) che corrispondono agli slot di parcheggio.

Come riferimento si utilizza un modello dello sfondo dinamico costruito attraverso procedure che richiedono poca elaborazione [5,6] ma sono robuste ai cambiamenti di illuminazione delle scene in esterno. L'immagine di riferimento è aggiornata in continuazione utilizzando l'ultima immagine acquisita.

Un parcheggio libero si presenta come una zona di asfalto uniforme senza nulla all'interno: partendo da questa ipotesi si utilizzano due indici, risultato di due differenti analisi dell'immagine, per ottenere una predizione sicura dello stato di ogni parcheggio.

Come prima operazione si effettua la "rilevazione dell'asfalto": utilizzando come riferimento delle piccole zone rettangolari di strada (che si prendono nell'immagine di sfondo in modo che sicuramente nessun veicolo in movimento sia coinvolto) si filtra l'immagine corrente in base ai valori cromatici di questi campioni.

Per ogni Regione di Interesse R_k viene calcolato l'indice $a_k(t)$ ovviamente dipendente dal frame in oggetto (al tempo t).

Questo indice è proporzionale al rapporto tra il numero di "pixel asfalto" e il numero totale dei pixel nella Regione di Interesse. Il risultato di questo filtro è mostrato in figura 4-C).

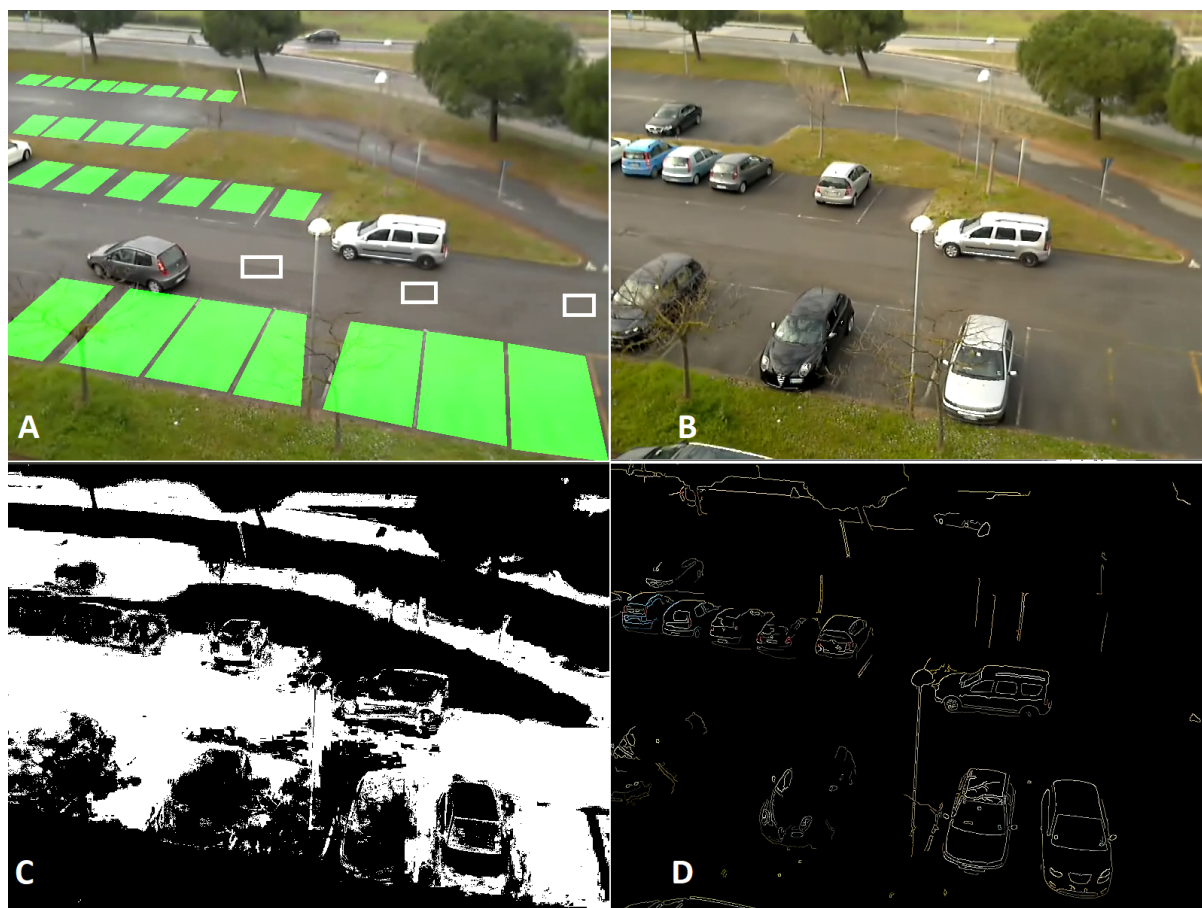


Figura 4 A) Le Regioni di Interesse (RoI) sono disegnate manualmente con l'ausilio di un tool grafico. Piccoli rettangoli sulla strada definiscono i campioni per la rilevazione dell'asfalto. B) L'immagine di sfondo di riferimento. C) Le zone bianche rappresentano la presenza di asfalto. D) I bordi delle auto sono rilevati mediante l'operatore di Canny.

La seconda elaborazione che viene effettuata è un rilevamento dei bordi delle automobili mediante l'operatore di Canny. Il risultato può essere osservato in figura 4-D). Come nel precedente caso viene calcolato un secondo indice $e_k(t)$ proporzionale al rapporto tra i pixel di contorno rilevati e il numero totale dei pixel nella Regione di Interesse R_k .

Questi due indici vengono combinati mediante prodotto (in modo che l'assenza di uno dei due pesi molto sul risultato finale)

La probabilità che un parcheggio associato alla RoI k sia occupato al tempo t è data dunque dalla formula:

$$P(t) = e(t) * [1 - a(t)]$$

La figura 5 mostra lo stato di occupazione degli stalli: verde è disponibile, rosso è occupato e blu incerto. Lo stato di un parcheggio diventa occupato solo dopo che è stato osservato tale per un numero di frame consecutivi

predefinito (per evitare falsi positivi dovuti a passaggi temporanei nelle zone di parcheggio). Ad esempio si può osservare come lo stato del parcheggio in basso a destra sia ancora blu (incerto) in quanto la macchina bianca è appena giunta e non sono ancora passati il numero di frame previsti per un'occupazione certa.



Figura 5 - Lo stato di occupazione degli stalli: verde è disponibile, rosso è occupato e blu incerto (in tal caso si mantiene lo stato precedente)

4 Conteggio parcheggi globale

In questa sezione presentiamo un ulteriore approccio usato per realizzare un software in grado di effettuare una stima del numero di veicoli presenti all'interno di un parcheggio monitorato da una smart camera. Anche questa soluzione, come quella descritta nel paragrafo 2, utilizza delle smart cameras con reti neurali.

In particolare, per prima cosa introduciamo le motivazioni a monte della scelta di sviluppare un secondo approccio alternativo ai precedenti, quindi descriviamo la rete neurale e la procedura utilizzata, e infine descriviamo l'algoritmo che viene eseguito per effettuare il conteggio delle macchine a tempo di esecuzione.

4.1 Introduzione e motivazioni

Entrambi gli approcci per il monitoraggio dello stato di occupazione degli stalli di un parcheggio descritti nei precedenti paragrafi, necessitano, per poter funzionare correttamente, di conoscere la mappatura degli stalli.

Infatti, come abbiamo visto, il loro funzionamento si basa sull'elaborazione di un singolo stallo alla volta, e sull'aggregazione di questi singoli risultati.

Pertanto, data un'immagine del parcheggio acquisita da una smart camera, l'operazione preliminare che quest'ultima deve obbligatoriamente fare è ritagliare l'immagine stessa in immagini più piccole (*patch*), ciascuna contenente un unico spazio di parcheggio, seguendo la mappatura degli stalli memorizzata al suo interno.

Si può quindi facilmente notare che, ogni qualvolta il parcheggio monitorato varia (cioè ogni volta che, per qualsiasi motivo, la disposizione degli stalli che formano il parcheggio cambia), il sistema deve essere nuovamente configurato inserendo una nuova e consistente mappatura degli stalli.

Quanto descritto può rappresentare una limitazione per questo approccio, in quanto in questi casi è necessario un intervento manuale atto a segmentare l'immagine e a memorizzare la nuova mappatura degli stalli nella smart camera. Questa operazione può essere anche dispendioso in termini di tempo, in maniera variabile a seconda della complessità del parcheggio monitorato.

Con lo scopo di ottenere una soluzione ancora più flessibile rispetto alle due precedentemente proposte, che funzioni a prescindere dalla conoscenza preliminare della mappatura degli stalli, abbiamo progettato e sviluppato un

nuovo software avente come obiettivo principale quello di effettuare una stima del numero di macchine presenti nell'immagine catturata dalla smart camera, con una conseguente valutazione del grado di occupazione del parcheggio monitorato.

Spesso, infatti, anche la sola conoscenza di quest'ultima informazione può essere di grande aiuto, anche senza avere la conoscenza della precisa localizzazione dei singoli stalli liberi e occupati: un parcheggio che risulta essere pieno può infatti essere immediatamente scartato da un automobilista, con un conseguente risparmio di tempo e carburante.

Utilizzando questo approccio, l'unica informazione di cui si ha bisogno oltre all'immagine del parcheggio acquisita dalla smart camera risulta essere il numero totale degli stalli disponibili. Questo dato è molto più facile da reperire e da memorizzare all'interno del sistema, e ciò rende questa soluzione molto più flessibile e più facilmente applicabile in nuovi contesti o comunque in situazioni che cambiano e si evolvono.

4.2 Realizzazione della rete neurale

Per sviluppare questo approccio abbiamo utilizzato Mask R-CNN [3], una rete convoluzionale avente come scopo originale quello di individuare le istanze degli oggetti presenti in una immagine. In particolare, gli oggetti trovati vengono identificati mediante delle maschere: l'immagine viene cioè segmentata e le istanze degli oggetti trovati sono individuate da dei poligoni, le cui coordinate rappresentano l'uscita della rete. Conteggiando le istanze afferenti ad una certa categoria (nel nostro caso la categoria 'automobile') possiamo quindi effettuare una stima dei veicoli presenti nella scena.

La versione base di Mask R-CNN da cui siamo partiti è addestrata utilizzando il dataset Microsoft COCO [4], una collezione di immagini rappresentanti oggetti generici. Abbiamo quindi raffinato questa rete neurale affinché riconoscesse in maniera migliore le automobili, poichè il nostro scopo è appunto il riconoscimento e la localizzazione delle stesse. Per farlo, abbiamo nuovamente addestrato Mask R-CNN con il dataset CNRPark-EXT, descritto nel paragrafo 2.2 .

4.3 Implementazione dell'algoritmo per il conteggio automatico

In questa sezione descriviamo l'algoritmo usato dal sistema per effettuare una stima del grado di occupazione del parcheggio monitorato, che utilizza la rete neurale convoluzionale descritta nel paragrafo precedente.

Anche in questo caso, come nella precedente soluzione che utilizzava una rete neurale per classificare i parcheggi occupati e vuoti, tutta l'elaborazione viene effettuata a bordo della smart camera. Il sistema comprende quindi anche un server che riceve le informazioni elaborate dalla telecamera intelligente e che, eventualmente, le visualizza.

Con questo nuovo approccio, a differenza dei precedenti, è sufficiente memorizzare su ogni smart camera il numero totale degli stalli del parcheggio che si vuole monitorare. In fase di esecuzione, la rete neurale elabora direttamente l'immagine intera acquisita dalla cam, fornendo in uscita una stima del numero dei veicoli presenti nel parcheggio. Confrontando questa grandezza con il numero totale degli stalli, il software è in grado di effettuare facilmente una stima del grado di occupazione del parcheggio stesso. La Figura 6 di seguito mostra due immagini acquisite da due smart camera che monitorano due differenti parcheggi su cui abbiamo applicato il software descritto.

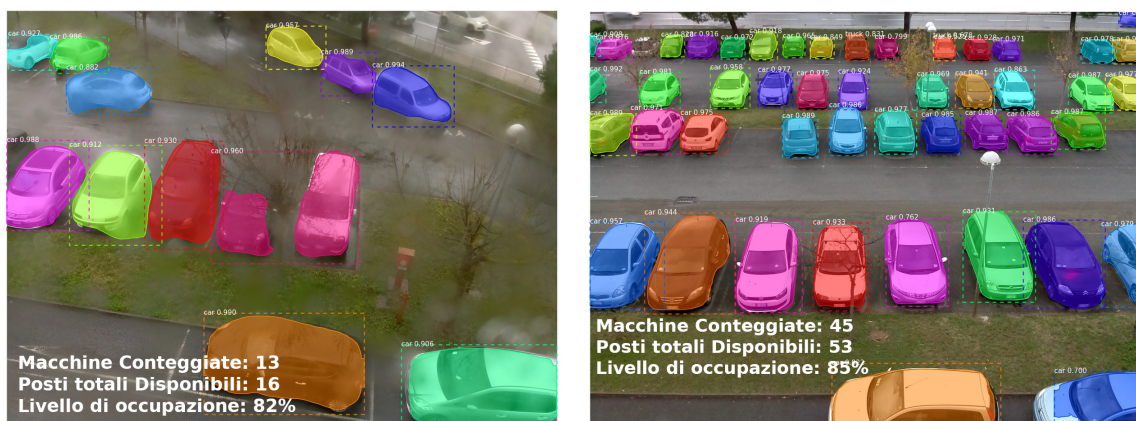


Figura 6: Due esempi di immagini acquisite da due smart camera diverse, con le stime del conteggio di veicoli presenti calcolate dal nostro software.

Si noti che, anche in questo caso, l'unica informazione trasmessa al server è un numero che indica il grado di occupazione del parcheggio, e che nessuna immagine viene inviata all'esterno, garantendo la privacy delle targhe e riducendo considerevolmente la larghezza di banda necessaria.

Si noti infine che, all'occorrenza, questa soluzione può facilmente essere estesa e modificata per risolvere il precedente obiettivo del monitoraggio dello stato di occupazione degli stalli. Questa rete neurale, infatti, fornisce in uscita anche le coordinate dei poligoni che individuano le istanze dei veicoli trovati. Confrontando questi dati con la mappatura degli stalli, si possono quindi facilmente individuare gli stalli liberi e gli stalli occupati. Ovviamente, anche in questo caso si presenta nuovamente il problema iniziale, poiché è necessario fornire preliminarmente la segmentazione del parcheggio alla smart camera.

1. Amato G, Carrara F, Falchi F, Gennaro C, Meghini C, Vairo C. Deep learning for decentralized parking lot occupancy detection. *Expert Syst Appl.* 2017;72: 327–334.
2. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM.* 2017;60: 84–90.
3. Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN
4. Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. Microsoft COCO: Common Objects in Context.
5. Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L., "Background modeling and subtraction by codebook construction," In: *IEEE Int. Conf. on Image Processing*, 2004.
6. Stauffer, C., and Grimson, W. E., "Adaptive background mixture models for real-time tracking," In: *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp.246-252, 1999.