ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Progetto SmartPark@Lucca

D5. - Integrazione e sperimentazione sul campo

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 WP3, in particolare relative al Task 3.1 - Integrazione e al Task 3.2 - Sperimentazione sul campo.

Indice

1 Introduzione

In questo documento descriviamo gli esperimenti eseguiti per il riconoscimento dello stato di occupazione degli stalli di un parcheggio nell'area urbana di Lucca e gli esperimenti per il conteggio globale delle auto nello stesso parcheggio.
A tale scopo abbiamo condotto una raccolta di immagini del parcheggio di Santa Maria a Lucca in una intera giornata e abbiamo usato tale dataset per eseguire gli esperimenti. Nella Sezione 2 presentiamo il dataset raccolto.
Nelle sezioni 3 e 4 presentiamo i due approcci utilizzati per la classificazione dei singoli stalli del parcheggio, uno tramite l'uso di smart cameras con reti neurali (Sezione 3) e uno tramite l'utilizzo di smart embedded cameras (Sezione 4).
Nella sezione 5 descriviamo l'approccio del conteggio globale delle auto nel parcheggio analizzato e riportiamo i relativi risultati.

2 Acquisizione immagini parcheggio Santa Maria e realizzazione dataset

In questa sezione presentiamo il dataset che abbiamo costruito al fine di testare in un contesto applicativo reale gli approcci descritti nel "*Deliverable D4: Progettazione e realizzazione software di riconoscimento visuale parcheggi*".

2.1 Realizzazione dataset

Con lo scopo di testare il software per il monitoraggio visuale dei parcheggi descritto nei Deliverable precedenti, abbiamo costruito un dataset che descrive un tipico contesto applicativo reale.

Questo dataset, che abbiamo chiamato SmartPark-Lucca, contiene circa 700 immagini acquisite da uno dei parcheggi suggeriti da Metro e oggetto di sopralluogo del personale del CNR, descritto nel documento "*Relazione attività primo anno*". In particolare, il parcheggio scelto per la sperimentazione è quello sito nella Zona Piazza Santa Maria di Lucca.
Vista l'impossibilità di installare telecamere fisse nei punti stabiliti durante il sopralluogo, sono state utilizzate due telecamere poste su due cavalletti mobili posizionati su Porta Santa Maria, che inquadravano da due differenti prospettive uno dei due parcheggi principali precedentemente individuati.
Una visuale dall'alto dell'area di parcheggio monitorata è riportata in Figura 1, dove sono anche indicati i campi di visualizzazione delle due telecamere utilizzate.

Il numero totale di stalli monitorati all'interno del parcheggio è 38. Come accennato, le due telecamere monitorano la stessa area di parcheggio da due differenti prospettive: questa ridondanza di dati può eventualmente essere sfruttata per monitorare in maniera ottimale quegli stalli che, da una visuale risultano essere difficilmente identificabili (perché coperti da altre macchine, da alberi, o da altri ostacoli), ma che da una differente prospettiva sono invece più facilmente individuabili. La mappatura degli stalli è riportata in Figura 2a per quanto riguarda la telecamera 1, e in Figura 2b per quanto riguarda la telecamera 2. Mentre in Figura 3 è riportata lo schema planimetrico

complessivo degli stalli monitorati, con evidenziati in rosso gli stalli particolari (es. Parcheggio invalidi, carico/scarico, etc.).

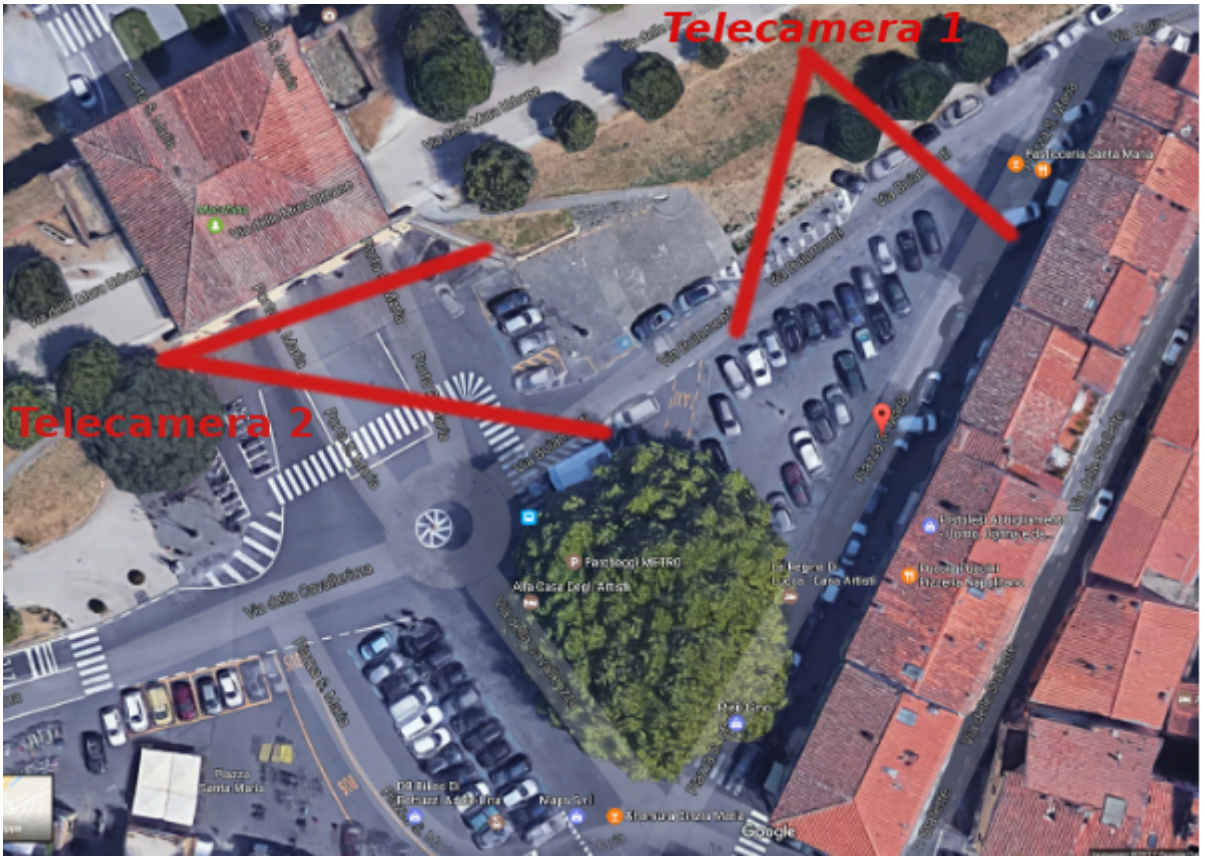

Fig.1 - Visuale dall'alto dell'area di parcheggio monitorata sita nella Zona Piazza Santa Maria di Lucca: in rosso sono evidenziate le due telecamere utilizzate e i rispettivi campi di copertura.

Fig. 2a - Mappatura degli stalli della camera 1.

Fig. 2b - Mappatura degli stalli della camera 2.

Abbiamo suddiviso questo dataset in due parti: una parte composta da circa 600 immagini da utilizzare per addestramenti futuri di ulteriori algoritmi, e un'altra parte composta da circa 100 immagini che abbiamo invece utilizzato per testare il software per il monitoraggio visuale dei parcheggi in oggetto in questi deliverable. Queste immagini di test sono state opportunamente etichettate per essere utilizzate sia mediante l'approccio per la classificazione dello stato di occupazione degli stalli, sia mediante l'approccio per il conteggio globale delle auto presenti nel parcheggio.
Si noti che la telecamera 1 copre un totale di 31 stalli, mentre la telecamera 2 copre tutti i 38 stalli monitorati. Si noti inoltre che abbiamo considerato solo gli stalli ufficialmente localizzati dalle strisce blu, e alcuni stalli aventi scopi particolari ed individuati da strisce gialle.

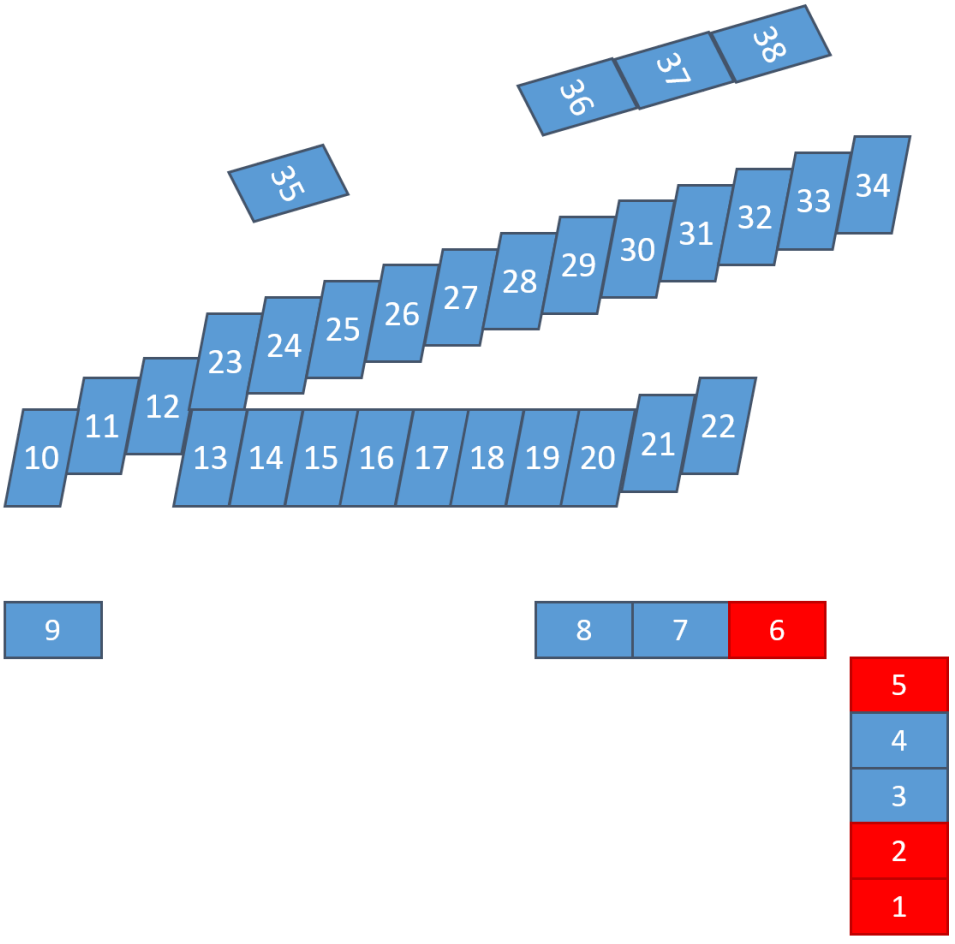

Fig. 3 - Schema con planimetria totale degli stalli, rispettive posizioni, e suddivisione tra stalli "blu" e stalli riservati (rossi).

3 Classificazione singoli stalli tramite Smart Cameras con reti neurali

In questa sezione presentiamo gli esperimenti eseguiti sfruttando le smart cameras con reti neurali sul dataset raccolto nel parcheggio di Santa Maria a Lucca. In particolare, abbiamo usato un test set composto da 100 immagini scelte in modo casuale dal dataset completo.
Abbiamo creato manualmente delle maschere per ciascuna delle due telecamere che abbiamo usato per ritagliare le immagini dei singoli stalli. Ogni maschera corrisponde a un singolo stallo. La rete neurale, infatti, è in grado di classificare un singolo stallo in libero o occupato, per cui è necessario interrogare la rete neurale passandogli in input l'immagine di un singolo stallo. Le maschere utilizzate sono mostrate in Figura 4a e 4b.

Fig. 4a - Maschere dei singoli stalli della camera 1

Fig. 4b - Maschere dei singoli stalli della camera 2

Una volta ritagliate le immagini dei singoli stalli del test set, abbiamo etichettato manualmente ogni singolo stallo in libero o occupato. Questo passo è necessario per creare il ground truth che permette poi di calcolare quanto è precisa la predizione della rete neurale.
Alcuni esempi di stalli liberi e occupati per le due camere sono riportati nelle Figure 5a e 5b.

Fig. 5a - Esempi di stalli liberi (prima fila) e occupati (seconda fila) della camera 1

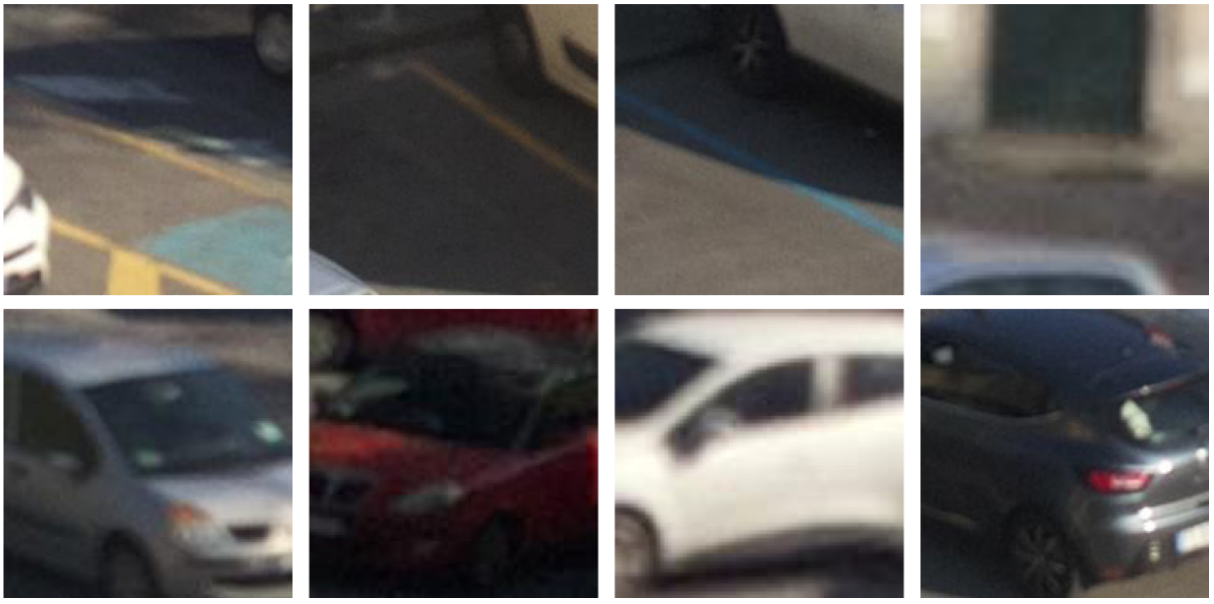

Fig. 5b - Esempi di stalli liberi (prima fila) e occupati (seconda fila) della camera 2

3.1 Risultati

Abbiamo quindi usato la rete neurale addestrata a classificare i singoli stalli per determinare se gli stalli ottenuti dal test set fossero liberi o occupati e abbiamo confrontato la predizione restituita dalla rete neurale con il ground truth che abbiamo precedentemente calcolato.

L’accuracy ottenuta è del 96% per la camera 1 e del 96,5% per la camera 2. Questo è un risultato molto buono, considerando che la rete neurale è stata

addestrata, come descritto nel D4 “*Progettazione e realizzazione software di riconoscimento visuale parcheggi*”, sul dataset CNRPark-EXT [1] Questo dataset è stato raccolto nel parcheggio dell’area di ricerca di del CNR di Pisa, ed è diverso da quello di Santa Maria a Lucca. Quindi la rete neurale non ha mai processato, prima di questi test, le immagini del parcheggio di Lucca. Questo conferma che un tale approccio è sufficientemente generico, da poter essere utilizzato anche in altri parcheggi, senza rendere necessario un ri-addestramento della rete neurale.

Le Figure 6a e 6b riportano un esempio di output della classificazione dei singoli stalli della rete neurale.

Fig. 6a - Esempio di classificazione dei singoli stalli per la camera 1

Fig. 6b - Esempio di classificazione dei singoli stalli per la camera 2

4 Classificazione singoli stalli tramite Smart Embedded Cameras

Questo tipo di telecamere sono sensori a bassissimo consumo energetico molto versatili perché caratterizzati dalla peculiarità di poter funzionare in modo autonomo alimentati da un pannello solare ed un modulo batteria. Per una descrizione dettagliata del dispositivo e delle sue applicazioni si faccia riferimento a [2], [3] e [4]. La potenza elaborativa a disposizione è limitata ed in particolare gli algoritmi di analisi delle immagini non possono essere svolti su risoluzioni elevate. Il primo passo della sperimentazione è stato dedicato ad un preprocessamento atto a esaudire le caratteristiche richieste.

4.1 Preprocessing delle immagini

Il dataset acquisito nella campagna di acquisizione svolta a Lucca è composto da immagini ad alta risoluzione (3280x2464) salvate ad intervalli regolari di 6 secondi (un tempo adeguato all'osservazione dei cambiamenti che avvengono in un parcheggio).
Nel tempo totale di 6 ore sono state raccolte 3690 immagini temporalmente sequenziali per una dimensione totale di 20.6 Gigabyte.
Le embedded camera, per questioni soprattutto legate alla limitata memoria di lavoro, supportano una risoluzione massima molto inferiore.
Il primo passo del pre-processamento è stata la riduzione della risoluzione (*resize)* mantenendo lo stesso rapporto larghezza/altezza per non introdurre distorsione; abbiamo ottenuto immagini con risoluzione 1280x962.
Si è reso necessario un secondo passo per ritagliare l'immagine (*crop)* eliminando una parte senza informazioni significative per ottenere la risoluzione desiderata di 1280x720. Queste operazioni riducono la dimensione dei dati di un fattore 10 (2.5 Gigabyte).
Infine tutte le immagini a risoluzioni più bassa sono utilizzate per creare artificialmente un video in formato h264. La grandissima ridondanza tra le immagini (sono delle riprese a postazione fissa con cambiamenti relativamente lenti) permette una elevata compressione dei dati: con soli 92 Megabyte tutti i frame sono codificati con una piccolissima perdita di qualità.

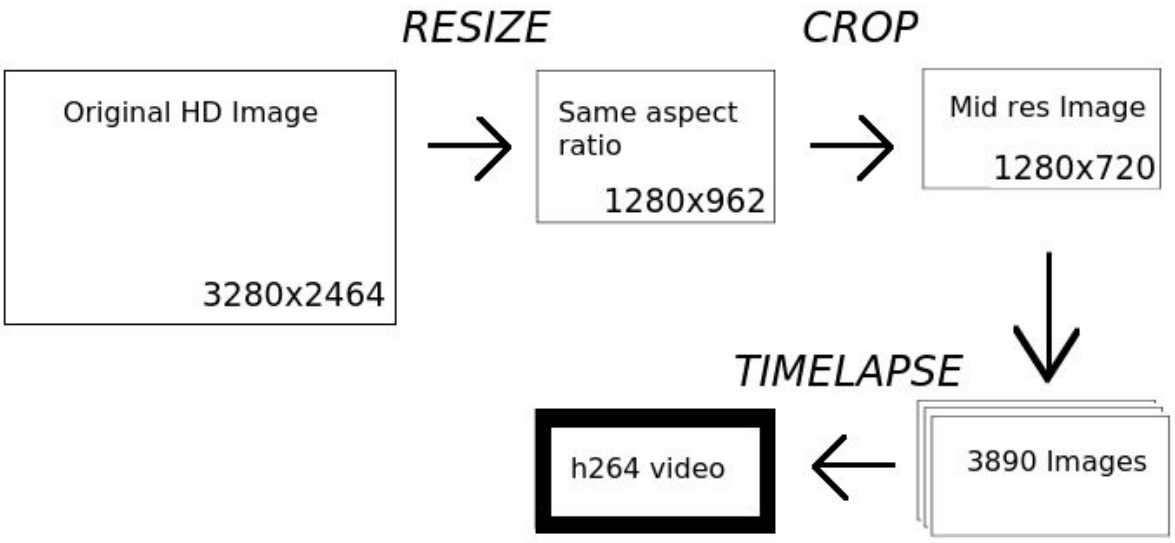

Fig. 7 - Lo schema del pre-processamento delle immagini ad alta risoluzione necessario per le embedded cameras. Tale procedura è inoltre molto utile per ridurre la dimensione dei dati da archiviare.

La procedura descritta è indispensabile per le esigenze delle embedded cameras, ma può essere adottata per un'archiviazione a lungo termine di sequenze time-lapse con grande risparmio di risorse in confronto alle originali immagini ad alta risoluzione.

4.2 Elaborazione

Le embedded cameras sono dispositivi a bassa potenza elaborativa. Per questo motivo non è possibile utilizzare algoritmi che fanno uso intenso di CPU o di memoria come le deep neural network presentate nel capitolo precedente. Bisogna ricorrere a tecniche di computer vision "leggere" che unite ad un efficiente utilizzo della conoscenza a priori del contesto permettono di raggiungere risultati soddisfacenti e comparabili con quelli ottenuti mediante mezzi elaborativi più performanti.
Un esempio di utilizzo efficiente della conoscenza a priori riguarda la definizione delle ROI (Regions Of Interest) ovvero le parti di immagini che rappresentano i parcheggi; non ci si riferisce alle strisce blu sull'asfalto, bensì a dei poligoni liberamente disegnati che permettono di ridurre al massimo possibili occlusioni dovute alla prospettiva o ad oggetti fissi della scena come alberi o pali. Nelle figure che seguono, in modo intuitivo, lo stato occupato è indicato dal colore rosso e lo stato disponibile da quello verde. Come si può vedere in figura 8a le regioni di riferimento devono essere tali da evitare sovrapposizioni ed infatti il posto in seconda fila sulla destra è correttamente

individuato disponibile anche se parzialmente occluso dalla macchina in prima fila; d'altro canto le ROI devono essere abbastanza grandi per contenere informazioni sufficienti, altrimenti si rischiano falsi negativi come in figura 8b dove il posto sulla destra è erroneamente indicato disponibile (cosa che succede quasi sempre nelle posizioni più lontane in assenza di sufficiente illuminazione).

Fig. 8a Le ROI sono disegnate per ridurre le occlusioni

Fig. 8b Un esempio di falso negativo dovuto alla scarsa illuminazione

Per determinare lo stato di un determinato slot di parcheggio si cerca di inferire se in quello spazio di asfalto è presente qualcosa di sufficientemente grosso per un tempo adeguato (bisogna evitare di segnalare occupati i posti in presenza di pedoni che camminano).
Un primo indicatore è fornito dall'algoritmo di *edge detection* (rilevamento dei bordi), il cui output si può osservare in figura 9; esso si basa sul gradiente di intensità locale e ci fornisce una stima della presenza o meno

Fig. 9 - Il rilevamento dei bordi (edge-detection) è una elaborazione che permette di ottenere informazioni utili per determinare la presenza di elementi in una zona dell'immagine.

di oggetti in una determinata zona: i pixel bianchi indicano la presenza di un bordo mentre i pixel neri una zona di immagine omogenea. Nel primo caso possiamo essere abbastanza sicuri della presenza di oggetti nella ROI, mentre nel secondo caso potrebbe essere asfalto uniforme, ma anche la fiancata di un furgone o, come in figura 8b, una parte di immagine non abbastanza illuminata.
Il secondo indicatore è relativo alla informazione colore contenuta nella zona di interesse. Andando ad analizzare la distribuzione dei colori rappresentati con il modello HSV (intensità, tonalità e saturazione) siamo in grado di distinguere una zona uniforme come l'asfalto (che presenta una distribuzione molto stretta attorno ad un valore di intensità) da un'altra dove sono presenti più elementi (ruote, carrozzeria, parabrezza); come possiamo osservare in figura 10 è molto facile individuare veicoli a tinte chiare o colorate perché presentano una distribuzione delle intensità molto ampia, mentre per auto scure che si avvicinano al colore dell'asfalto la differenza è meno netta ma pur sempre significativa.

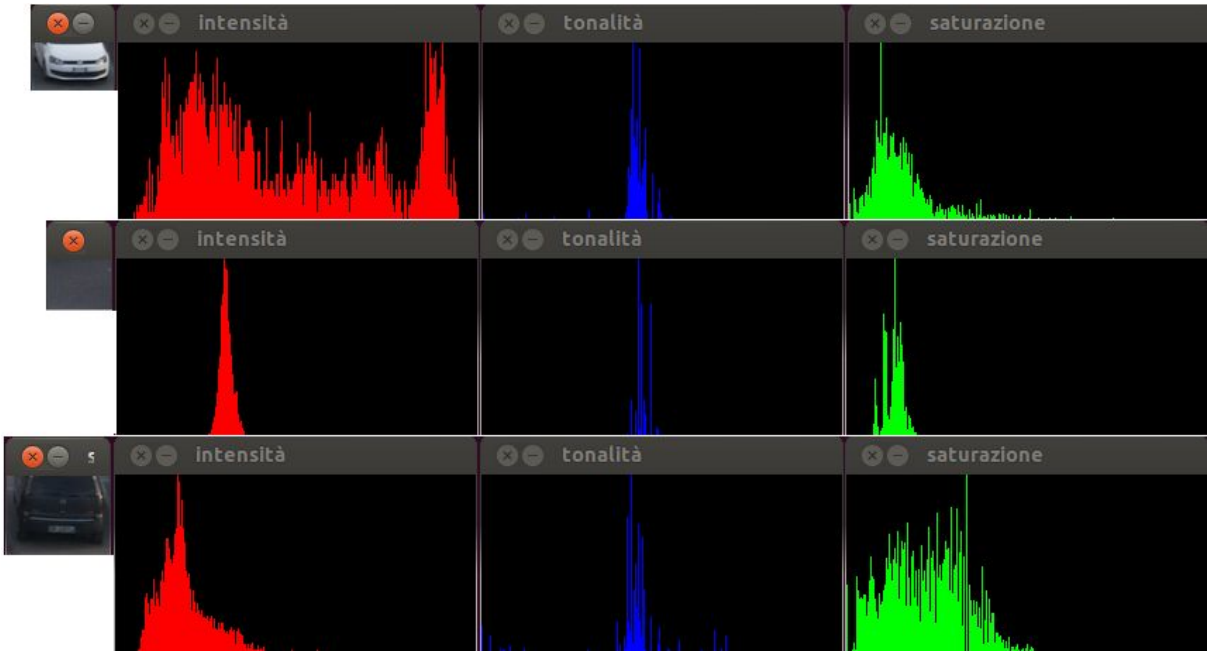

Fig. 10 - L'analisi della componente colore della ROI mostra come solo con il parametro intensità è facile distinguere una zona uniforme come l'asfalto (riga centrale) da una che ha all'interno degli autoveicoli;

Dall'utilizzo congiunto di questi due indicatori viene calcolato un indice globale *g=f(edge,varianza istogramma)* che normalizzato nell'intervallo (0,1) esprime la "probabilità" di occupazione dello slot; un valore reale che tiene conto anche dell'incertezza dovuta alle situazioni meno chiare osservate in precedenza.

4.1 Risultati

Per comparare i risultati delle diverse tecnologie implementate nel progetto abbiamo scelto in modo casuale 100 immagini dai 3890 che compongono il set iniziale. Tra queste 100 ne sono state selezionate 15 che fossero rappresentative di varie situazioni (slot liberi, occlusioni, differente illuminazione solare). Per ogni frame sono stati considerati tutti gli errori di valutazione, sia i falsi positivi (posti che vengono segnalati occupati ma in realtà sono liberi) che i falsi negativi (posti che vengono segnalati liberi, ma in realtà sono occupati).

Il primo conteggio è stato effettuato considerando tutti i posti visibili dal sensore numero 1, anche quelli lontani e in posizione disagiata.
Questo permette il monitoraggio di 30 stalli dei 38 totali. La tabella riassuntiva è la seguente:

Telecamera n.1 - 30 posti monitorati

Frame	FP+FN	Err Rate
Cam1_1125	1+4	5/30 = 0.167
Cam1_1150	1+2	3/30 = 0.100
Cam1_1473	1+3	4/30 = 0.133
Cam1_2163	0+2	2/30 = 0.067
Cam1_2166	0+1	1/30 = 0.033
Cam1_2170	0+1	1/30 = 0.033
Cam1_2177	0+1	1/30 = 0.033
Cam1_2180	1+1	2/30 = 0.067
Cam1_2262	1+1	2/30 = 0.067
Cam1_2362	0+2	2/30 = 0.067
Cam1_2556	0+5	5/30 = 0.167
Cam1_2713	0+6	6/30 = 0.2
Cam1_3024	0+4	4/30 = 0.133
Cam1_3211	0+0	0/30 = 0
Cam1_3591	0+0	0/30 = 0

Errore medio 0.084 = 8.4% ovvero una accuratezza del 91.6%

Si osserva immediatamente che la stragrande maggioranza degli errori sono dovuti ai falsi negativi ovvero percezione di posto libero in presenza di macchina. Come si può osservare in figura 11a questo accade maggiormente nei posti più lontani dalla telecamera. Le embedded camera infatti lavorano

con immagini ad medio-bassa risoluzione e quindi al crescere della distanza dal sensore diminuisce la quantità di immagine analizzata e dunque aumenta la percentuale di errore.

Fig. 11a - Le embedded camera lavorano con immagini a risoluzione medio-bassa e l'accuratezza scende molto al crescere della distanza dal sensore.

Per ottenere risultati migliori bisogna limitare il campo di pertinenza del sensore ad un raggio di 20-25 metri ed affidarsi ad un altro sensore per le zone oltre questa distanza.

Fig. 11b - Gli stalli monitorati dal sensore numero 2

E’ quello che abbiamo fatto con il sensore numero 2, considerando solo i 19 stalli più vicini alla telecamera,come visibile in figura 11b. Non essendo presenti occlusioni problematiche i risultati migliorano sensibilmente come mostrato dalla tabella riassuntiva.

Telecamera n.2 - 19 posti monitorati

Frame	FP+FN	Err Rate
Cam1_1125	0+0	0/19 = 0
Cam1_1150	0+0	0/19 = 0
Cam1_1473	1+0	1/19 = 0.053
Cam1_2163	0+0	0/19 = 0
Cam1_2166	0+0	0/19 = 0
Cam1_2170	0+0	0/19 = 0
Cam1_2177	0+0	0/19 = 0
Cam1_2180	0+0	0/19 = 0
Cam1_2262	0+0	0/19 = 0
Cam1_2362	1+1	2/19 = 0.105
Cam1_2556	0+0	0/19 = 0
Cam1_2713	0+0	0/19 = 0
Cam1_3024	0+0	0/19 = 0
Cam1_3211	0+0	0/19 = 0
Cam1_3591	0+0	0/19 = 0

Errore medio 0.011 = 1.1% ovvero una accuratezza del 98.9%

A questo punto è interessante andare a vedere l’accuratezza totale del “sistema cooperativo” ovvero considerando tutti i 38 stalli ma scegliendo il risultato migliore dei posti monitorati da entrambe i sensori (in base ad un parametro “confidenza” stabilito sperimentalmente dal confronto dei risultati con il ground truth).

Telecamera n1 + telecamera n.2 - 38 posti monitorati

Frame	FP+FN	Err Rate
Cam1_1125	1+2	3/38 = 0.079
Cam1_1150	1+2	3/38 = 0.079
Cam1_1473	2+3	5/38 = 0.132
Cam1_2163	0+0	0/38 = 0
Cam1_2166	0+0	0/38 = 0
Cam1_2170	0+0	0/38 = 0
Cam1_2177	0+0	0/38 = 0

Cam1_2180	0+0	0/38 = 0
Cam1_2262	1+0	1/38 = 0.026
Cam1_2362	1+1	2/38 = 0.053
Cam1_2556	1+1	2/38 = 0.053
Cam1_2713	1+3	4/38 = 0.105
Cam1_3024	1+2	3/38 = 0.079
Cam1_3211	0+0	0/38 = 0
Cam1_3591	0+0	0/38 = 0

Errore medio 0.04 = 4% ovvero una accuratezza del 96%

Come ci si poteva aspettare il risultato del sistema complessivo è influenzato positivamente dalle ottime prestazioni del sensore numero 2 che gode di visuale migliore proprio rispetto a quelle postazioni che nel sensore numero 1 generano molti falsi negativi.

L'insegnamento è abbastanza intuitivo: per ottenere una accuratezza soddisfacente è fondamentale disporre di un numero adeguato di telecamere rispetto all'area da monitorare e che vanno disposte in modo tale che ogni posto sia visibile senza ostruzioni da almeno uno dei sensori.

5 Conteggio parcheggi globale

In questa sezione presentiamo i risultati ottenuti applicando sullo scenario applicativo precedentemente descritto l'approccio per effettuare la stima del numero globale di veicoli presenti nel parcheggio monitorato, descritto nella sezione 4 del "*Deliverable D4: Progettazione e realizzazione software di riconoscimento visuale parcheggi*".

5.1 Risultati

Come abbiamo visto, l'approccio basato sul conteggio globale delle macchine presenti nel parcheggio risulta essere più flessibile di quello che monitora lo stato di occupazione degli stalli, in quanto non è necessaria la conoscenza della mappatura degli stalli stessi.
Abbiamo quindi testato la rete neurale sviluppata in precedenza per stimare il numero di autovetture presenti nelle immagini afferenti al dataset SmartPark-Lucca.
L'algoritmo che abbiamo utilizzato è quello allenato con il dataset CNRPark-EXT. Nessun ulteriore addestramento è stato effettuato utilizzando il nuovo dataset SmartPark-Lucca: quest'ultimo è stato impiegato esclusivamente per testare il software di conteggio automatico in un contesto applicativo reale, e per verificare che quest'ultimo fosse in grado di generalizzare le prestazioni su nuovi scenari. In Figura 12a e 12b sono riportati due esempi del nostro approccio su due immagini acquisite dalle due differenti telecamere.
Si noti che le uniche informazioni che abbiamo dato all'algoritmo, oltre ovviamente alle immagini da analizzare, sono state il numero totale dei posti disponibili (31 nel caso della telecamera 1 e 38 nel caso della telecamera 2) e una indicazione approssimativa delle regioni dell'immagine da considerare, in modo da scartare le porzioni del parcheggio che non si volevano monitorare.

Fig. 12a - Esempio di conteggio dei veicoli dalla Telecamera 1

Fig. 12b - Esempio di conteggio dei veicoli dalla Telecamera 2

E’ stata inoltre effettuata un’analisi quantitativa dei risultati ottenuti. La metrica comunemente utilizzata in letteratura per il conteggio di oggetti contenuti in immagini si chiama Errore Assoluto Medio (EAM). Essa è semplicemente una media degli errori totali di conteggio commessi nelle singole immagini.
Abbiamo ottenuto un valore pari a 3.6: in media, quindi, l’errore di valutazione che commettiamo su una immagine è inferiore alle 4 macchine conteggiate.

Riferimenti

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. M. Alam, D. Moroni, G. Pieri, M. Tampucci, M. Gomes, J. Fonseca, J. Ferreira and G.R. Leone *Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities*. Journal of Advanced Transportation, vol. 2018, Article ID 1485652, 13 pages, 2018.
3. G.R.Leone, M.Magrini, D.Moroni, G.Pieri, O.Salvetti and M.Tampucci *A smart device for monitoring railway tracks in remote areas*. In proc. of IEEE Int. Workshop on Computational Intelligence for Multimedia Understanding, Reggio Calabria(IT), 27-28 Ottobre 2016.
4. D. Moroni, G. Pieri, G. Palazzese, G. R. Leone, M. Magrini, and O. Salvetti *Computer Vision on Embedded Sensors for Traffic Flow Monitoring*. In proc. of 18th IEEE Int. Conf. On Intelligent Transportation Systems, Las Palmas de Gran Canaria(ES), 15-18 Settembre 2015.