

# Rapporto Tecnico

Giorgio Gemignani<sup>1</sup> and Errico Balestrieri<sup>2</sup>

<sup>1</sup> DSA - University of Naples Parthenope  
Centro Direzionale, Isola C/4, 80143 Naples, Italy

<sup>2</sup> Canale8  
Via Depretis 111, 80131 Naples, Italy

## 1 Contenuto del Documento

Questo documento descrive i risultati dell'attività di prototipazione svolta nell'ambito del progetto Amiav che ha portato all'integrazione del software di riferimento per la realizzazione di un sistema automatico per il monitoraggio del traffico veicolare, all'interno della piattaforma hardware progettata. In particolare verranno presentati i risultati preliminari raggiunti dalle attività di testing degli algoritmi utilizzati e successivamente si descriveranno la fase di testing dell'intero sistema di videosorveglianza.

## 2 Introduzione

All'interno del progetto AMIAV, al fine di massimizzare la qualità del sistema ed al tempo stesso ridurre i tempi di sviluppo, è stato adottato un sviluppo basato su approccio di tipo incrementale, caratterizzato da una separazione iniziale tra lo sviluppo della piattaforma hardware e l'implementazione del software di visione artificiale che consentisse al sistema di stimare la densità veicolare.

Seguendo questa metodologia di progettazione nella prima iterazione del ciclo di sviluppo, partendo dalle specifiche attese per il sistema sono state analizzate le tecniche di computer vision utili alla realizzazione di un sistema di Traffic Density Estimation. Tale analisi è stata sviluppata indipendentemente dai mezzi di acquisizione video, al fine di:

- individuare le tecniche più performanti a prescindere dalle caratteristiche degli strumenti di acquisizione video a nostra disposizione.
- dividere l'attività di prototipazione in due task paralleli e indipendenti assegnati a due gruppi di lavoro.

In questo modo è stato possibile suddividere l'attività di sviluppo e testing in due fasi indipendenti.

In un primo momento sono stati analizzati e testati le tecniche di visione artificiale presenti in letteratura, su riprese video benchmark rese disponibili dalla computer vision community e riprese registrate durante le campagne di acquisizione dati, al fine di selezionare le soluzioni più efficaci ed efficienti al problema del Traffic Density estimation.

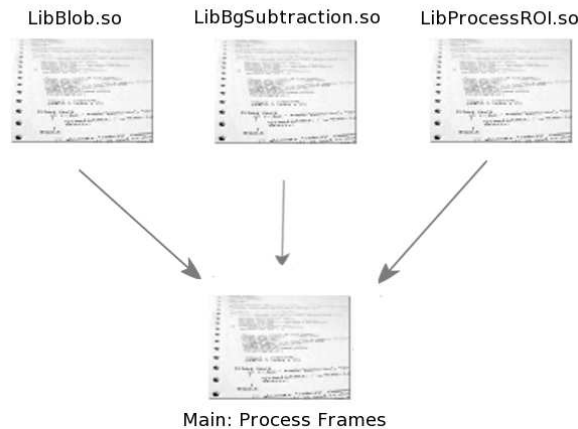
In una seconda fase si è proceduto all'integrazione della pipeline di processing sviluppata nella fase precedente all'interno framework ZoneMinder ed al testing dell'infrastruttura costituente l'intero sistema di videosorveglianza sullo streaming video acquisito direttamente dalle telecamere installate nel VideoServer, al fine di analizzare le prestazioni del sistema in condizioni di carico reale.

### 3 Progettazione del software

Come previsto dal progetto AMIAV, sono stati studiati e sviluppati algoritmi di visione artificiale sia per l'analisi del livello di occupazione della superficie stradale sia per la stima dei flussi veicolari.

Gli algoritmi prototipali sono stati sviluppati in linguaggio c++, utilizzando la libreria opensource OpenCV [7] sviluppata da Intel, integrata alle api OpenMP [6], utilizzate per sfruttare a pieno la potenza di calcolo di tutte le cpu presenti all'interno del processore multicore installato nella piattaforma hardware utilizzata.

Seguendo un approccio di sviluppo modulare il codice è stato organizzato in un insieme di librerie dinamiche caricate dalla routine principale **PROCESS FRAMES** che si occupa di recuperare il flusso video e processarlo attraverso le tecniche di visione artificiale adottate. In Figura 1 è rappresentata l'organizzazione del codice sviluppato. In particolare, possiamo distinguere le seguenti



**Fig. 1.** Organizzazione Sorgenti

dynamic linked libraries:

- **LibBgSubtraction.so**: libreria che contiene le routine di Background subtraction necessarie alla modellazione dello sfondo ed all'individuazione dei pixel di foreground presenti nell'immagine.

- **LibBlob.so**: libreria contenente l'algoritmo per la ricerca delle componenti connesse necessario a clusterizzare i pixel di foreground, comunemente definiti Blob.
- **LibProcessRoi.so**: libreria che contiene le routine necessarie alla selezione delle regioni di interesse (**ROI**) all'interno dell'immagine osservata e ed il calcolo della trasformazione prospettica da applicare alle ROI

La funzione **PROCESS FRAMES** (schematizzata in pseudo-codice) contiene il nucleo di elaborazione delle immagini a regime.

L'algoritmo si basa su un approccio di background subtraction per evidenziare i cambiamenti nelle regioni di interesse, rispetto ad un modello di background di riferimento generato offline attraverso una rete neurale autororganizzante SOM secondo l'approccio proposto in [5].)

### PROCESS FRAMES

Input: pixel  $\mathbf{x}$  in sequence frame  $I_t, t = 0, \dots, \text{LastFrame}$

Output:

$Count_t$ , numero veicoli in movimento nella ROI

$Density_t$ , stima della densità del traffico veicolare

0.  $ROI = SelectROI(I_0)$
1.  $ROI_{perspective} = PerspectiveTransform(ROI)$
2.  $InitializeModels(B_0(\mathbf{x}))$
3. **for**  $t=1, Kinit$
4.  $CalibrateModels(B_t(\mathbf{x}))$
5. **for**  $t=CurrentFrame$
6.  $(F(\mathbf{x}) = UpdateModels(B_t(\mathbf{x})), I_t(\mathbf{x}))$
7.  $Blobs_t = GetConnectedComponents(F(\mathbf{x}))$
8.  $(Count_t, Density_t) = CountBlob(Blobs_t, ROI_w)$

Per vantaggi computazionali, l'intero processo di elaborazione viene eseguita solo su una o più regioni di interesse (**ROI**) predeterminate, corrispondenti ai tratti stradali che si desidera monitorare. A tale **ROI** è applicata un'operazione di preprocessing che consente di rendere l'area da analizzare nelle fasi successive invariante alla prospettiva.

Questa operazione, é una trasformazione prospettica che manterrá fisse le dimensioni dei veicolo in transito che si avvicinano o si allontanano dalla telecamera, come riportato nelle figure 3 e 8. Per ciascuna regione viene accumulata la somma dei valori assoluti delle differenze pixel a pixel, opportunamente corrette per ottenere maggiore robustezza rispetto ai cambiamenti nella illuminazione ambientale.

A tal fine vengono impiegate versioni normalizzate dell'immagine corrente rispetto a parametri globali di illuminazione (media e varianza della immagine corrente e di riferimento).

La somma delle differenze viene scalata rispetto all'area della regione di interesse ed infine archiviata in un buffer. Vengono quindi eseguiti controlli per verificare se il valore archiviato rappresenta un cambiamento nell' immagine. In particolare, confrontando il valore archiviato rispetto allo storico vengono filtrati eventuali valori sopra la media dovuti alla presenza di ombre.

Allo stesso modo, il valore archiviato viene confrontato con una soglia per individuare eventuali cambiamenti rispetto al modello di riferimento. L'algoritmo giunge cosí ad individuare i pixel in movimento all'interno dell' immagine producendo una maschera binaria rappresentativa degli oggetti in movimento nei frame.

Studiando le componenti connesse, vengono individuati dei blob di interesse. Infine si verifica se i blob individuati possono essere riferiti ad oggetti in moto attraverso una ROI predefinita e si procede al loro conteggio (funzionalità aggiuntiva ed ancora in fase di sviluppo) ed alla stima della densità del traffico veicolare misurata come percentuale dell' area totale **ROI** occupata da oggetti in movimento.

## 4 Attività di Testing

Gli algoritmi prototipali sopra discussi sono stati testati sulle sequenze di test descritte nelle Sezioni 5 e 6, seguendo modalità di valutazione descritta di seguito. Gli eventi di interesse per la stima della densità veicolare sono rappresentati da:

- Numero di veicoli in transito lungo l'area di interesse.
- Percentuale dell'area di interesse occupata da un veicolo in transito.

Al fine di produrre delle misure di precisione ed accuratezza della stima del livello di traffico veicolare prodotta dal sistema è stato necessario produrre delle tavole verità che misurassero nel tempo i livelli di traffico presenti nelle **ROI**.

A tal scopo sono state generate delle ground truth da osservatore umano che ha rivisto le sequenze di test offline ed ha annotato in una tabella frame per frame il numero esatto di veicoli in transito lungo la **ROI** ed il livello di traffico.

Nello specifico si è scelto di impostare a 3 il numero di livelli di densità veicolare etichettando le entry della tavola di verità con le seguenti label:

- **0** corrispondente a condizioni di **Traffico Basso**.
- **1** corrispondente a condizioni di **Traffico Medio**.
- **2** corrispondente a condizioni di **Traffico Elevato**.

Il sistema sarà dunque in grado di classificare il livello di traffico presente in un frame della sequenza video analizzata come appartenente ad una degli eventi sopra elencati. Il numero di veicoli in transito nella **ROI** corrispondente alla livello stimato dal sistema dipenderà dalle caratteristiche dello scenario monitorato, ovvero dalla dimensione dell'area di interesse selezionata e dal numero massimo di veicoli che essa potrà contenere.

La bontà della classificazione prodotta dal sistema sarà quindi misurata ricercando il giusto tradeoff tra i *falsi negativi* ed i *falsi positivi* che il sistema potrà rilevare per ciascuna delle condizioni di traffico sopra definite.

Attraverso la matrice di confusione  $M$  è possibile confrontare i risultati ottenuti dal sistema sviluppato in rapporto alle classificazioni corrette. La tabella 1 mostra la configurazione di tale matrice in relazione allo specifico problema trattato  $Fmeasure$ .

Livello Reale \ Livello Stimato	Traffico Basso (0)	Traffico Medio (1)	Traffico Elevato(2)
Traffico Basso (0)	$T_p$	$F_n$	$F_n$
Traffico Medio (1)	$F_p$	$T_p$	$F_n$
Traffico Elevato (2)	$F_p$	$F_p$	$T_p$

**Table 1.** Risultati

I valori di  $M$  possono assumere le seguenti configurazioni:

$M(i, j) = T_p$ : *veri positivi* per la classe  $i$ -esima, ovvero il numero di eventi  $i$  correttamente classificati dal sistema.

$M(i, j) = F_p$ : *falsi positivi* per la classe  $i$ -esima, ovvero il numero di eventi delle classi  $i$  erroneamente classificati dal sistema come classe  $j$ .

$M(i, j) = F_n$ : *falsi negativi* per la classe  $i$ -esima, ovvero il numero di eventi delle classi  $j$  erroneamente classificati dal sistema come classe  $i$ .

I risultati ottenuti in termini di *veri positivi*, *falsi positivi* e *falsi negativi* per tutte le classi definite nel contesto applicativo, sono confrontati in termini di *true positive rate* e *false positive rate* mediante la rappresentazione della curva **ROC ( Receiver Operating Characteristic)** secondo il criterio **One versus All**

I parametri che definiscono i punti nello spazio *ROC* sono i valori delle soglie  $s_1, s_2$  in base ai quali il livello di traffico (percentuale della ROI occupata da veicoli in transito ) viene classificato negli eventi **0,1,2**. Variando le i valori delle coppie  $s_1, s_2$  é stato possibile stabilire la configurazione ottimale di tali parametri per massimizzare le prestazioni del sistema in termini di *True positive rate* e *False positive rate*. Tale configurazione é stata individuata attraverso la *Fmeasure*.

In particolare per ogni evento di Traffico stabilito si definiscono le seguenti misure di precisione:

**True positive rate** , la capacità del sistema di individuare correttamente i valori positivi, ovvero la percentuale delle classificazioni positive correttamente effettuate dal sistema rispetto a tutte le classificazioni positive effettuate dall' annotatore:

$$TP_{rate} = \frac{T_p}{P}$$

**False positive rate, (Recall)** la misura di quanto il sistema sbaglia nel classificare i valori positivi, ovvero la percentuale delle classificazioni positive erroneamente effettuate dal sistema rispetto a tutte le classificazioni negative effettuate dall' annotatore,:

$$FP_{rate} = R = \frac{F_p}{N}$$

**Precision**, percentuale delle classificazioni positive correttamente effettuate dal sistema rispetto a tutte le classificazioni positive effettuate dal sistema

$$P = \frac{T_p}{T_p + F_p}$$

**Fmeasure**, misura che combina Precision e Recall attraverso media armonica nota in letteratura come *balanced F-score* o *F1*:

$$Fmeasure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Nelle sezioni ?? e ?? sono valutate le prestazioni del sistema secondo i suddetti parametri.

## 5 Test Preliminari

Al fine di velocizzare la fase di analisi e selezione delle tecniche di computer vision più efficienti per il problema del Traffic Density Estimation si è deciso di analizzare le prestazioni degli algoritmi prototipali sui datasets di benchmark per *i*-LIDS 2007 (Image Library for Intelligent Detection System) [9], competizione sul rilevamento di Veicoli in sosta vietata. Tali datasets sono liberamente scaricabili all'indirizzo <ftp://motinas.elec.qmul.ac.uk/pub/iLids>.

Essi rappresentano un tipico scenario cittadino, caratterizzato da una principale ed un attraversamento pedonale dove i veicoli che transitano regolarmente sono costretti a fermarsi per consentire il passaggio dei pedoni. Le sequenze **Easy**, **Medium** e **Hard** in analisi illustrate in figura 2 sono riprese effettuate in diversi momenti della giornata su una strada di Londra, caratterizzate da una difficoltà crescente relativamente al task di rilevamento automatico di sosta vietata. La crescente difficoltà è caratterizzata dall'aumento della densità veicolare presente nelle tre sequenze e dall'aumentare della distanza dalla telecamera della posizione cui il veicolo sosta illegalmente.

Tale contesto rappresenta uno scenario ideale di strada trafficata per la sperimentazione preliminare degli algoritmi prototipali sviluppati, costituenti il nucleo operativo del sistema proposto.



**Fig. 2.** Dataset ILIDS 2007

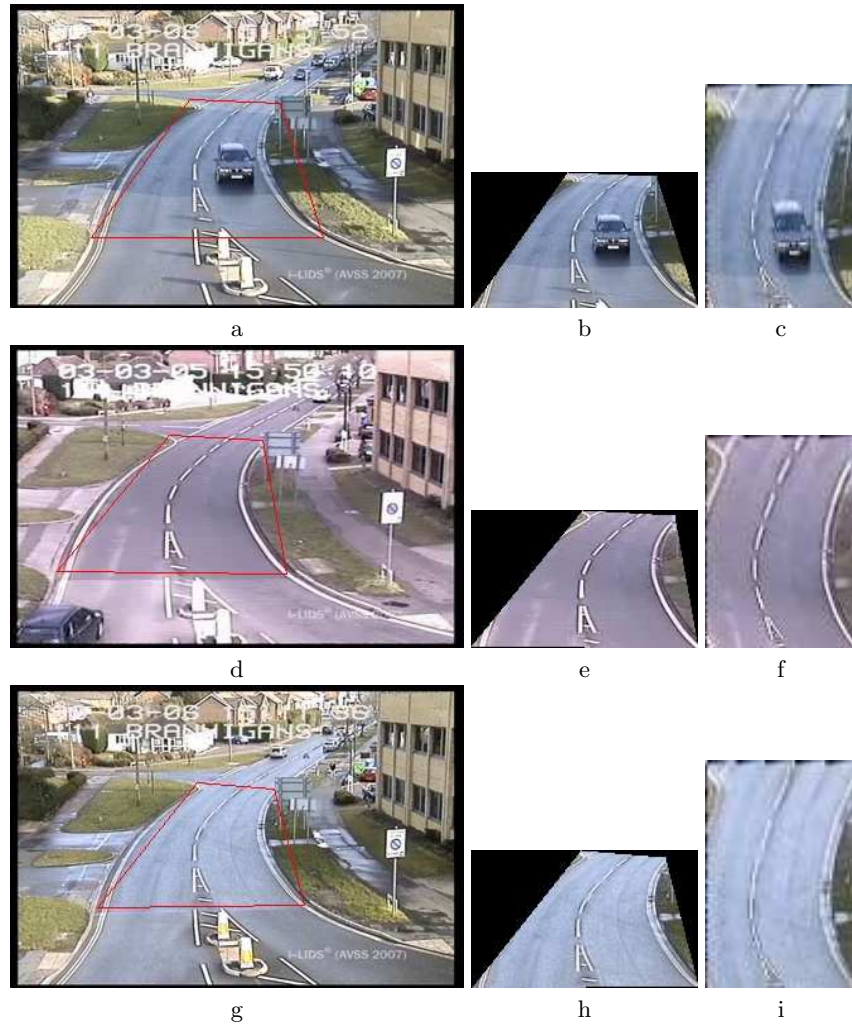
In Tabella 2 sono riassunti alcuni dettagli tecnici delle sequenze di acquisizione.

**Table 2.** Dettagli sequenze di Test ILIDS 2007

<i>Video</i>	<i>Size</i>	<i>Color Type</i>	<i>Frame Rate (fps)</i>	<i>Durata</i>
Sequenza Easy	$360 \times 240$	RGB	25	02 : 30
Sequenza Medium	$360 \times 240$	RGB	25	02 : 22
Sequenza Hard	$360 \times 240$	RGB	25	02 : 54



Per ciascuna sequenza é stata selezionata una regione di interesse sulla quale stimare la densità veicolare come illustrato in figura .



**Fig. 3.** Sequenze Analizzate e relative ROI (a,b,c) Easy, (d,e,f) Medium, (g,h,i) Hard

Nella prima colonna sono illustrate le scene monitorate e le relative **ROI** evidenziate in rosso. Nella seconda e terza colonna é illustrato il risultato della fase di preprocessing che consente di estrarre la **ROI** dalla scena (colonna 2) ed applicare la trasformazione prospettica (colonna 3).

Per ciascuna sequenza é stato estratto uno spezzone di 1 minuto corrispondente a 1500 frames, sufficientemente variabile nei livelli di traffico in modo da poter misurare le prestazioni della classificazione in tutte le condizioni di traffico definite nella fase preliminare di progettazione. In tabella 3 é riportata la ground truth delle sequenze appena descritte.

**Table 3.** Ground truth

<i>Video</i>	<i>Traffico Basso</i>	<i>Traffico Medio</i>	<i>Traffico Alto</i>	<i>Numero Veicoli Passanti</i>
Sequenza Easy	447	770	283	28
Sequenza Medium	281	697	522	21
Sequenza Hard	603	819	78	17

## 5.1 Risultati

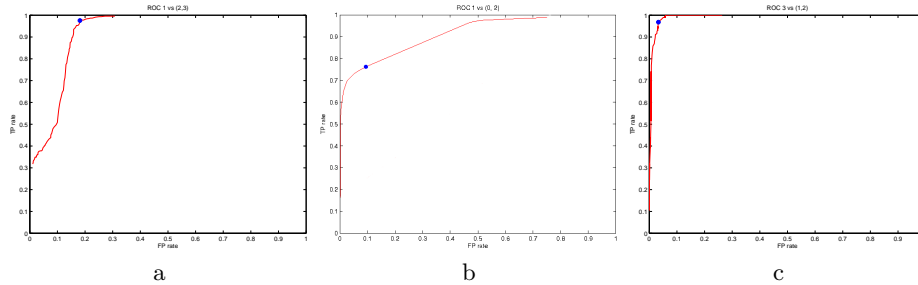
In questa sezione sono analizzati i risultati ottenuti sulle sequenze **Easy**, **Medium** e **Hard** del dataset. I test effettuati hanno prodotto risultati incoraggianti evidenziando come la scelta dei valori ottimali delle soglie  $s_1$  e  $s_2$  rappresenti un problema essenziale per il corretto rilevamento dei livelli di traffico definiti, di cui é riportato un esempio in figura 4.



**Fig. 4.** Esempio di esecuzione sulla sequenza Easy

É possibile notare come il sistema proposto sia in grado di stimare il livello del traffico veicolare ed allo stesso tempo valutare il numero di veicoli presenti nella scena osservata.

In figura 5 sono riportate le curve ROC per la sequenza Easy mentre in tabella 4 sono riportate, rispettivamente, le matrici di confusione, i valori delle soglie  $s_1, s_2$  che massimizzano la  $Fmeasure$  ed i corrispettivi valori di *True positive rate*, *False positive rate* e *Precision*.

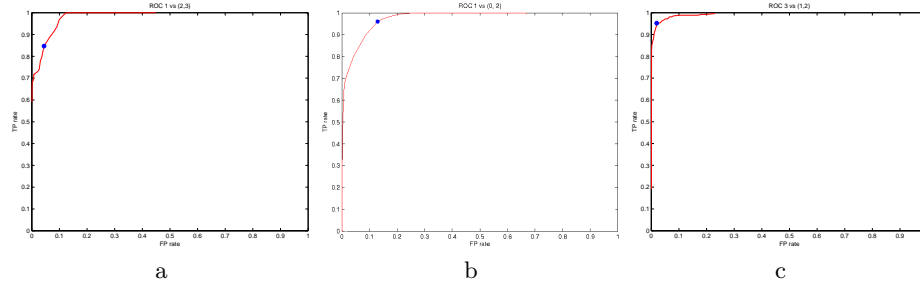


**Fig. 5.** Sequenza Easy : (a) Roc 0 vs (1,2). (b) Roc 1 vs (0,2). (c) Roc 2 vs (0,1)

L.Reale \ L.Stimato	<b>0</b>	<b>1</b>	<b>2</b>	$Tp_{rate}$	$Fp_{rate}$	$Fmeasure$	$Precision$	s1	s2	Tipo ROC
Traffico Basso ( <b>0</b> )	436	11	0	0.98	0.18	0.81	0.70			
Traffico Medio ( <b>1</b> )	191	258	321	0.34	0.18	0.50	0.96	0.16	0.20	ROC 1 vs (2,3)
Traffico Elevato ( <b>2</b> )	0	0	283	1.00	0.26	0.81	0.47			
Traffico Basso ( <b>0</b> )	418	29	0	0.94	0.16	0.81	0.71			
Traffico Medio ( <b>1</b> )	167	587	16	0.76	0.16	0.82	0.89	0.15	0.37	ROC 2 vs (1,3)
Traffico Elevato ( <b>2</b> )	0	40	243	0.86	0.01	0.81	0.94			
Traffico Basso ( <b>0</b> )	142	305	0	0.32	0.01	0.47	0.93			
Traffico Medio ( <b>1</b> )	11	719	40	0.93	0.01	0.80	0.70	0.05	0.33	ROC 3 vs (1,2)
Traffico Elevato ( <b>2</b> )	0	9	274	0.97	0.03	0.47	0.87			

**Table 4.** Matrici di Confusione Sequenza Easy

In figura 6 sono riportate la curve ROC per la sequenza Easy mentre nella tabella 5 sono riportate rispettivamente le matrici di confusione per ogni singola classe, relative ai punti  $(s_1, s_2)$  che massimizzano la  $Fmeasure$  ed i corrispettivi valori di *True positive rate (Recall)*, *False positive rate* e *Precision*.

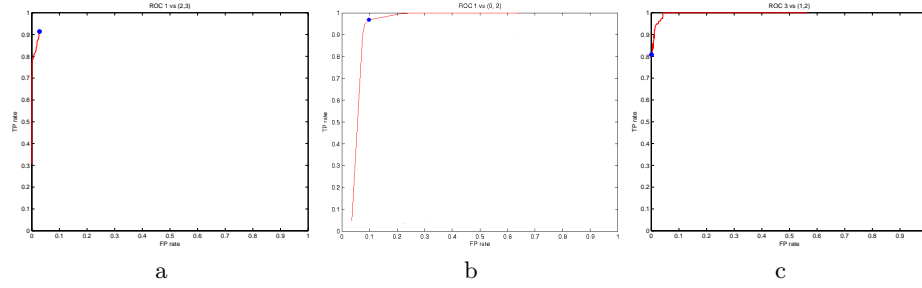


**Fig. 6.** Sequenza Medium : (a) Roc 0 vs (1,2). (b) Roc 1 vs (0,2). (b) Roc 2 vs (0,1)

L.Reale \ L.Stimato	0	1	2	$Tp_{rate}$	$Fp_{rate}$	$Fmeasure$	$Precision$	s1	s2	Tipo ROC
Traffico Basso ( <b>0</b> )	238	43	0	0.85	0.04	0.83	0.82			
Traffico Medio ( <b>1</b> )	54	418	225	0.60	0.04	0.72	0.90	0.09	0.18	ROC 1 vs (2,3)
Traffico Elevato ( <b>2</b> )	0	2	520	1.00	0.23	0.83	0.70			
Traffico Basso ( <b>0</b> )	202	79	0	0.72	0.01	0.82	0.96			
Traffico Medio ( <b>1</b> )	9	669	19	0.96	0.01	0.91	0.87	0.12	0.30	ROC 2 vs (1,3)
Traffico Elevato ( <b>2</b> )	0	25	497	0.95	0.02	0.82	0.96			
Traffico Basso ( <b>0</b> )	166	115	0	0.59	0.00	0.74	0.99			
Traffico Medio ( <b>1</b> )	1	677	19	0.97	0.00	0.89	0.83	0.05	0.27	ROC 3 vs (1,2)
Traffico Elevato ( <b>2</b> )	0	25	497	0.95	0.02	0.74	0.96			

**Table 5.** Matrici di Confusione Sequenza Medium

In figura 7 sono riportate la curve ROC per la sequenza Easy mentre nella tabella 6 sono riportate rispettivamente le matrici di confusione per ogni singola classe, relative ai punti  $(s_1, s_2)$  che massimizzano la  $Fmeasure$  ed i corrispettivi valori di *True positive rate (Recall)*, *False positive rate* e *Precision*.



**Fig. 7.** Sequenza Hard : (a) Roc 0 vs (1,2). (b) Roc 1 vs (0,2). (b) Roc 2 vs (0,1)

L.Reale \ L.Stimato	<b>0</b>	<b>1</b>	<b>2</b>	$Tp_{rate}$	$Fp_{rate}$	$Fmeasure$	$Precision$	s1	s2	Tipo ROC
Traffico Basso ( <b>0</b> )	551	25	27	0.91	0.03	0.93	0.96			
Traffico Medio ( <b>1</b> )	25	21	773	0.03	0.03	0.05	0.46	0.18	0.20	ROC 1 vs (2,3)
Traffico Elevato ( <b>2</b> )	0	0	78	1.00	0.56	0.93	0.09			
Traffico Basso ( <b>0</b> )	551	52	0	0.91	0.03	0.93	0.96			
Traffico Medio ( <b>1</b> )	25	793	1	0.97	0.03	0.94	0.92	0.18	0.45	ROC 2 vs (1,3)
Traffico Elevato ( <b>2</b> )	0	15	63	0.81	0.00	0.93	0.98			
Traffico Basso ( <b>0</b> )	184	419	0	0.31	0.00	0.47	1.00			
Traffico Medio ( <b>1</b> )	0	818	1	1.00	0.00	0.79	0.65	0.05	0.45	ROC 3 vs (1,2)
Traffico Elevato ( <b>2</b> )	0	15	63	0.81	0.00	0.47	0.98			

**Table 6.** Matrici di Confusione Sequenza Hard



Analizzando i risultati riportati nelle tabelle 5,6,7 é possibile notare la variabilit  della classificazione in funzione della scelta delle soglie  $s_1$  e  $s_2$ . In particolare un intervallo  $[s_1 s_2]$  stretto consente di aumentare l' accuratezza sulle classi Traffico Basso e Traffico Alto, mentre un intervallo pi  ampio determina l'incremento dell' accuratezza sulla classe Traffico Medio. Dai test effettuati sulle 3 sequenze il migliore compromesso in termini di prestazioni é stato ottenuto utilizzando i seguenti valori di  $(s_1, s_2)$

- $s_1 = 0.15$  e  $s_2 = 0.37$  per la sequenza Easy.
- $s_1 = 0.12$  e  $s_2 = 0.30$  per la sequenza Medium.
- $s_1 = 0.18$  e  $s_2 = 0.45$  per la sequenza Hard.

Tali valori consentono di raggiungere un elevato valore della  $Fmeasure$  per le 3 classi, e confermano la naturale intuizione che valori uniformi degli intervalli entro i quali la percentuale di area occupata dei pixel di foreground viene classificato in uno dei livelli definiti produgano risultati pi  accurati, rispetto ad altre configurazioni che invece tendono a privilegiare l'accuratezza dei 2 livelli Alto e Basso diminuendo l'accuratezza sul livello Medio.

## 6 Test su dati acquisiti

La seconda fase dell'attività di testing è stata effettuata su dati acquisiti direttamente da uno scenario cittadino. Per tale motivo è stata effettuata nella giornata del 15 Aprile 2010 una campagna di acquisizione volta a generare dei dati di traffico veicolare.

L'attività di acquisizione è stata svolta per l'intera giornata e ci ha consentito di registrare un insieme di dati eterogeneo che rappresentasse lo scenario in analisi nelle diverse ore della giornata.

In tale modo è stato possibile catturare le variazioni di illuminazione cui è sottoposto lo scenario nell'arco della giornata (variazioni di illuminazione, ombre, le variazioni del traffico veicolare nelle ore di punta), nonché valutare le prestazioni degli algoritmi sui dati catturati dai dispositivi di acquisizione che sono stati installati nel sistema completo.

Lo scenario di acquisizione è una strada comunale di Cercola, (Napoli) illustrata in figura ?? dove sono visualizzate le 2 sequenze acquisite ed utilizzate nel Testing.



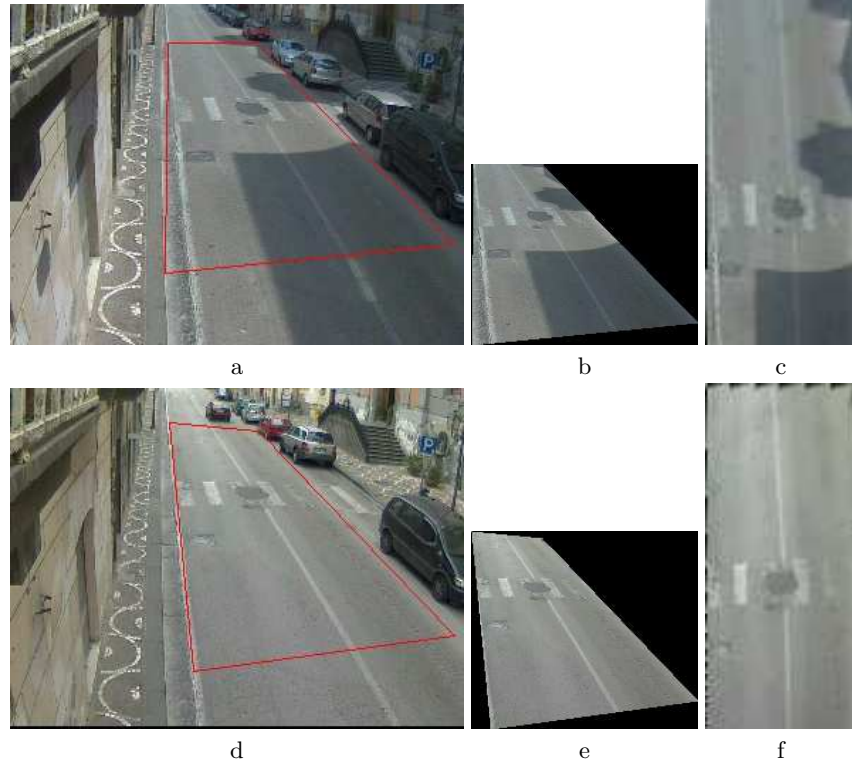
**Fig. 8.** Dataset Cercola 2010

In Tabella 7 si riassumono alcuni dettagli delle sequenze di acquisizione.

**Table 7.** Dettagli video sequenze Cercola

<i>Video</i>	<i>Size</i>	<i>Color Type</i>	<i>Frame Rate (fps)</i>	<i>Durata</i>	<i>Orario Acquisizione</i>
Sequenza Mattina	$360 \times 240$	RGB	25	12 : 30	12 : 30
Sequenza Pomeriggio	$360 \times 240$	RGB	25	15 : 22	17 : 50

Per ciascuna sequenza é stata selezionata la ROI sulla quale stimare la la densità veicolare come illustrato in figura 9.



**Fig. 9.** Sequenze Analizzate (a) Mattina, (d) Pomeriggio

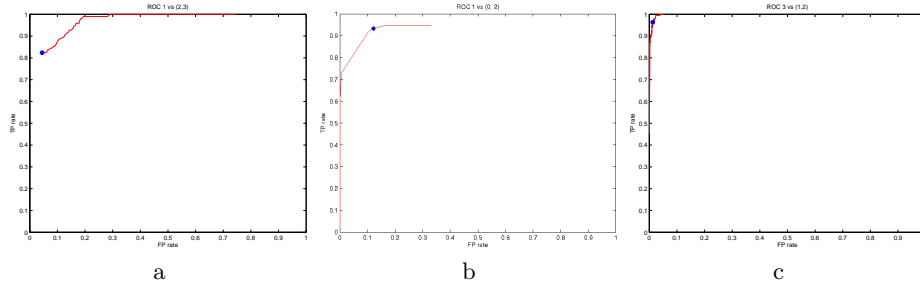
Per ciascuna sequenza é stata estatto uno spezzone di 1 minuto (1500 frames)sufficientemente variabile nei livelli di Traffico osservati ed é stata generata la ground truth riportata in tabella 8

**Table 8.** Ground truth

<i>Video</i>	<i>Traffico Basso</i>	<i>Traffico Medio</i>	<i>Traffico Alto</i>	<i>Numero Veicoli Passanti</i>
Sequenza Mattina	289	993	218	11
Sequenza Pomeriggio	1041	241	218	17

## 6.1 Risultati

Nelle sezione successiva sono riportati i risultati sulle sequenze **Mattina** e **Pomeriggio**. In figura 10 sono riportate la curve ROC per la sequenza **Mattina** mentre in tabella 9 sono riportate, rispettivamente, le matrici di confusione, i valori delle soglie  $s_1, s_2$  che massimizzano la *Fmeasure* ed i corrispettivi valori di *True positive rate*, *False positive rate* e *Precision*.

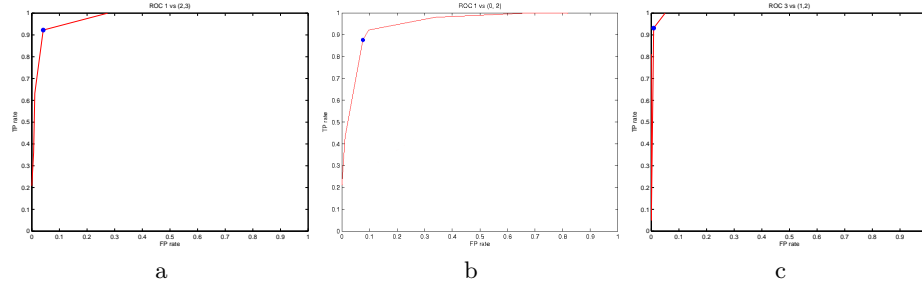


**Fig. 10.** Sequenza Mattina : (a) Roc 0 vs (1,2). (b) Roc 1 vs (0,2). (b) Roc 2 vs (0,1)

L.Reale \ L.Stimato	<b>0</b>	<b>1</b>	<b>2</b>	$Tp_{rate}$	$Fp_{rate}$	$Fmeasure$	$Precision$	s1	s2	Tipo ROC
Traffico Basso ( <b>0</b> )	238	51	0	0.82	0.04	0.82	0.82			
Traffico Medio ( <b>1</b> )	54	868	71	0.87	0.04	0.91	0.94	0.12	0.31	ROC 1 vs (2,3)
Traffico Elevato ( <b>2</b> )	0	0	218	1.00	0.06	0.82	0.75			
Traffico Basso ( <b>0</b> )	238	51	0	0.82	0.04	0.82	0.82			
Traffico Medio ( <b>1</b> )	54	926	13	0.93	0.04	0.93	0.94	0.15	0.38	ROC 2 vs (1,3)
Traffico Elevato ( <b>2</b> )	0	11	207	0.95	0.01	0.82	0.94			
Traffico Basso ( <b>0</b> )	238	51	0	0.82	0.04	0.82	0.82			
Traffico Medio ( <b>1</b> )	54	923	16	0.93	0.04	0.93	0.94	0.25	0.38	ROC 3 vs (1,2)
Traffico Elevato ( <b>2</b> )	0	8	210	0.96	0.01	0.82	0.93			

**Table 9.** Matrici di Confusione Sequenza Mattina

In figura 11 sono riportate la curve ROC per la sequenza Pomeriggio mentre in tabella 10 sono riportate, rispettivamente, le matrici di confusione, i valori delle soglie  $s_1, s_2$  che massimizzano la  $Fmeasure$  ed i corrispettivi valori di  $True\ positive\ rate, False\ positive\ rate$  e  $Precision$ .



**Fig. 11.** Sequenza Pomeriggio : (a) Roc 0 vs (1,2). (b) Roc 1 vs (0,2). (b) Roc 2 vs (0,1)

L.Reale \ L.Stimato	<b>0</b>	<b>1</b>	<b>2</b>	$Tp_{rate}$	$Fp_{rate}$	$Fmeasure$	$Precision$	s1	s2	Tipo ROC
Traffico Basso ( <b>0</b> )	960	81	0	0.92	0.04	0.95	0.98			
Traffico Medio ( <b>1</b> )	19	158	64	0.66	0.04	0.66	0.66	0.21	0.32	ROC 1 vs (2,3)
Traffico Elevato ( <b>2</b> )	0	0	218	1.00	0.05	0.95	0.77			
Traffico Basso ( <b>0</b> )	960	81	0	0.92	0.04	0.95	0.98			
Traffico Medio ( <b>1</b> )	19	211	11	0.88	0.04	0.77	0.69	0.21	0.39	ROC 2 vs (1,3)
Traffico Elevato ( <b>2</b> )	0	15	203	0.93	0.01	0.95	0.95			
Traffico Basso ( <b>0</b> )	217	824	0	0.21	0.00	0.34	1.00			
Traffico Medio ( <b>1</b> )	0	230	11	0.95	0.00	0.35	0.22	0.01	0.39	ROC 3 vs (1,2)
Traffico Elevato ( <b>2</b> )	0	15	203	0.93	0.01	0.34	0.95			

**Table 10.** Matrici di Confusione Sequenza Pomeriggio

I test effettuati hanno confermato i risultati preliminari evidenziati nella prima fase di testing ed hanno evidenziato l'efficacia del metodo proposto nel riconoscere i livelli di traffico.

In particolare é possibile notare come i livelli "Traffico Basso" e "Traffico Alto" sono riconosciuti con un' elevata accuratezza mentre il livello "Traffico Medio" é caratterizzato da una maggiore variabilitá, giustificata dall' imprecisione intrinseca nella definizione della classe corrispondente al suddetto livello di traffico.



## 7 Test Conclusivi

L'ultima fase dell'attività di prototipazione e testing effettuata ha riguardato l'integrazione del software sviluppato all'interno del servizio videosever **Zone-minder** [2] e l'analisi delle prestazioni del sistema in condizioni di carico. Seguono una breve descrizione del processo di integrazione e successivamente una descrizione dei test di carico effettuati

### 7.1 Integrazione in Zoneminder

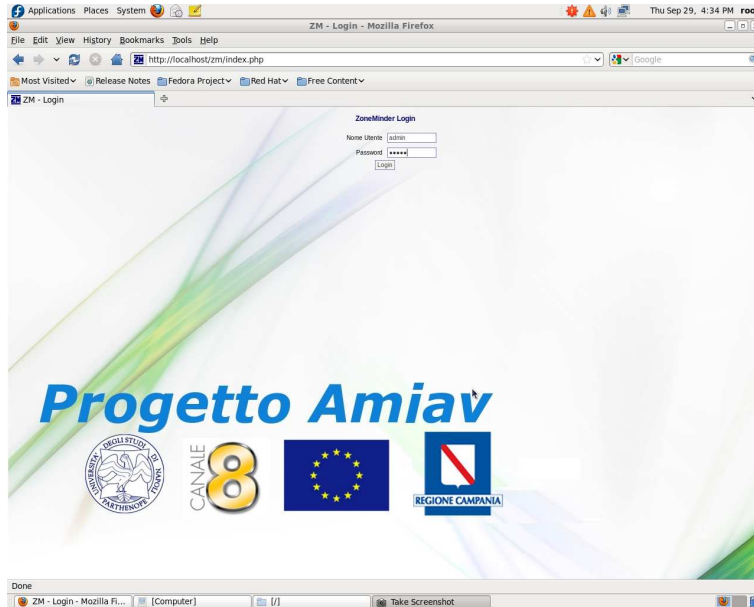
Per un'integrazione ottimale del software sviluppato all'interno di **Zoneminder** è stata effettuata un'attenta analisi dell'intero framework al fine di individuare i punti dell'applicazione dove andare ad integrare la pipeline di processing realizzata. ZoneMinder non è una singola applicazione monolitica, ma è formata da diversi componenti che comprendono essenzialmente eseguibili binari compilati che eseguono il task principale di elaborazione video, script perl che di solito eseguono helper e / o attività di interfaccia esterna e script php web che sono utilizzati per l'interfaccia web. I principali componenti analizzati sono:

- **zmc**: è il demone Capture di ZoneMinder incaricato di effettuare il video grabbing dai dispositivi video installati nel sistema. Tale processo dovrebbe funzionare alla velocità più o meno costante di **25 fps**.
- **zma**: è il demone Analisi di ZoneMinder attraverso il quale i frame acquisiti e vengono processati per il rilevamento di movimento e la generazione di allarmi o eventi.
- **zmf**: è il demone Frame di ZoneMinder incaricato di effettuare lo storing dei dati acquisiti dal demone zmc.
- **zms**: è lo Stream server di ZoneMinder cui l'utente si collega per visualizzare i flussi video.
- **zmu**: è una piccola utility di ZoneMinder attraverso la quale è possibile gestire da riga di comando l'intero sistema ed effettuare operazioni di debugging.

Obbiettivi dell'attività di integrazione è stato quello di interfacciare il software di rilevamento densità veicolare con il demone zmc in modo da dirigere lo stream acquisito dal videosever verso la nostra pipeline di processing. Tale operazione è stata effettuata integrando la soluzione sviluppata all'interno del codice sorgente costituente il progetto zoneminder.

## 7.2 Interfaccia Grafica

L'interfaccia utente grafica è il punto di accesso dell'utente al sistema di rilevamento traffico ed il mezzo che gli consente di interagire con il sistema stesso. Le Figure 12,13 mostrano le **Web GUI** realizzate, rispettivamente, per l'accesso e la gestione del sistema.

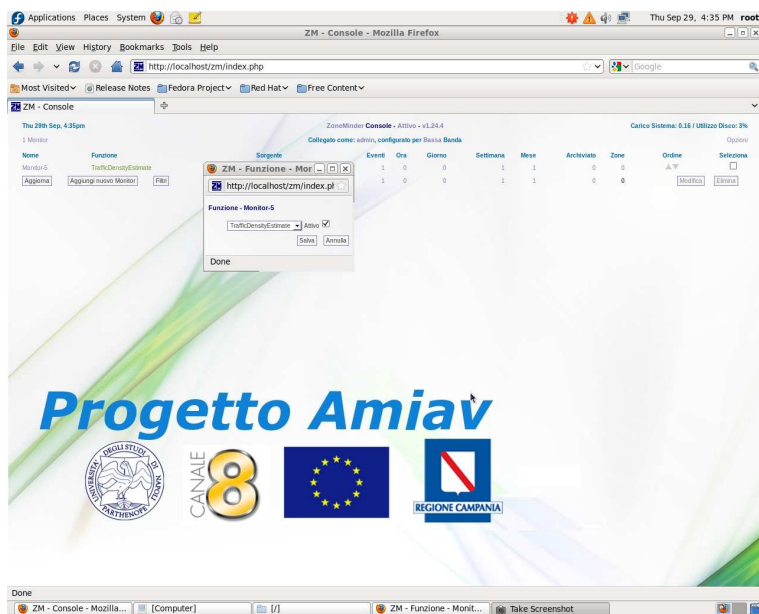


**Fig. 12.** Procedura di Login al sistema

L'interfaccia di gestione delle Telecamere è caratterizzata dalla lista di telecamere configurate all'interno del sistema per ciascuna delle quali sono visualizzate le seguenti informazioni:

- Nome della Telecamera.
- Funzione abilitata sulla Telecamera (come riportato nel documento )
- sorgente hardware da cui viene recuperato il flusso video.
- Informazioni riguardanti gli allarmi intercettati sul flusso video associato alla telecamera(data,orario, etc.)

In basso a sinistra è presente il menù per la configurazione di nuove telecamere nel sistema.



**Fig. 13.** Selezione della funzione associata alla telecamera

Cliccando sul Nome della telecamera é possibile visualizzare i risultati dell'algoritmo di stima del livello di traffico veicolare sul flusso video catturato dalla telecamera, come riportato in figura 14

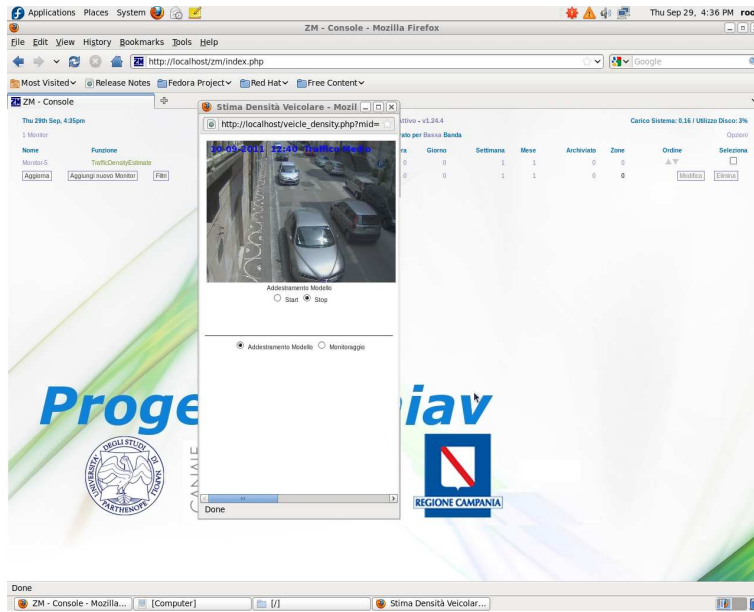


Fig. 14. Visualizzazione risultati della Stima Densit veicolare

### 7.3 Test di Carico

L'analisi del sistema in condizione di carico é stata oggetto dell'ultima fase della prototipazione.

Tale analisi ha consentito di valutare le prestazioni dell'intero sistema di videosorveglianza quando uno o piú utenti loggati al sistema utilizzano contemporaneamente diverse telecamere installate per la stima della densit  del traffico veicolare, causando inevitabilmente lo stress della macchina. Tali situazioni hanno consentito di verificare le prestazioni del sistema in termini di framerate garantito per il processamento degli stream video che si desidera monitorare simultaneamente. In tabella 11 sono riportati i tempi di esecuzione ottenuti testando il funzionamento del sistema fino ad un massimo di 3 telecamere che eseguono la stima del traffico veicolare concorrentemente. Per tempo di processamento é inteso il tempo trascorso dall'inizio di acquisizione del flusso video alla restituzione del valore stimato per il livello del traffico veicolare.

Table 11. Esecuzione con camere multiple

Telecamere	Tempo medio Esecuzione per Frame (sec.)	Fps
1 Network Camera	0.19	5.2
2 Network Camera	0.22	4.54
2 Network + 1 USB Camere	0.25	4

Osservando i valori riportati in tabella é possibile notare come il processamento di 3 flussi video concorrenti non introduca significativo overhead, mantenendo dello stesso ordine di grandezza il tempo di elaborazione dei flussi video. Considerando che il flusso video viaggia attraverso la rete, i frame rates ottenuti rappresentano sicuramente un ottimo risultato , consentendo ad un operatore di avere una visione istantanea dello stato del traffico presente sulle strade monitorate

## 8 Conclusioni

La prototipazione di un sistema di videosorveglianza capace di stimare la densità del traffico veicolare ha messo in evidenza la dipendenza di questo metodo dal modello di background e dai valori delle soglie per determinare il livello di traffico. L'impiego di intervalli uniformi ha fatto registrare le migliori risposte dell'algoritmo al rilevamento dei livelli di traffico prestabiliti.

Alla luce di quanto esposto e dai risultati sperimentali riportati, è possibile affermare che l'algoritmo utilizzato in questo sistema restituisce degli ottimi risultati se considerato nell'ottica di uno strumento di ausilio all'operatore. Infatti l'approccio robusto alla complessità degli scenari, tipico dei sistemi di rilevamento basato sull'elaborazione pixel-per-pixel, consente di evidenziare con buona accuratezza le diverse situazioni di traffico. In questo modo l'attenzione dell'operatore è rivolta solo agli eventi suggeriti dal sistema. Infatti è facile immaginare come questa circostanza possa essere d'aiuto in situazioni tipiche delle Sale Operative, dove il numero di monitor da tenere in osservazione è notevole.

## 9 Sviluppi futuri

Nell'ottica di un futuro miglioramento del sistema presentato è senza dubbio vantaggioso affiancare all'algoritmo di rilevamento un classificatore di veicoli che sia in grado di migliorare la precisione nell'individuazione dei livelli di Traffico. Questo porterebbe a filtrare i falsi allarmi provocati da pixel di foreground corrispondenti a forti variazioni di illuminazione.

## References

1. R.T. Collins et al., A System for Video Surveillance and Monitoring, The Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-00-12, 2000.
2. ZONEMINDER reference, <http://www.zoneminder.com>
3. J.M. Ferryman (Ed.): Proceedings of the 10th IEEE International Workshop on PETS, Rio de Janeiro, Brazil, October 14, 2007.
4. E. Herrero-Jaraba et al., Detected Motion Classification with a Double-background and a Neighborhood-based Difference, *Patt. Recogn. Lett.* 24, 2079–2092, 2003.
5. L. Maddalena and A. Petrosino, A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications, *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. July, 2008.
6. OpenMP, OpenMP guide, <http://www.openmp.org>
7. OpenCV, OpenCV guide, <http://opencv.willowgarage.com>
8. F. Porikli, Y. Ivanov, T. Haga, Robust Abandoned Object Detection Using Dual Foregrounds, *EURASIP Journal on Advances in Signal Processing*, 2008.
9. Proc. of Fourth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2007), IEEE Computer Society, 2007.