Politecnico di Torino
Corso di Laurea Magistrale in Ingegneria Informatica

Progetto Computer Vision

Conteggio persone in una stanza

Montanaro Teodoro (188924)

Federico Ianne (188283)

Sommario

Obiettivo

L'obiettivo della presente tesina consiste nel realizzare un programma che tramite l'interfacciamento con una telecamera, sia in grado di contare le persone presenti in un'aula universitaria, evitando di includere nel conteggio gli altri oggetti che tipicamente si trovano in questi luoghi (ad esempio i computer, le sedie ecc.).

Telecamera in dotazione

La telecamera fornita è la videocamera di rete Samsung SND-3080C.
Essa viene fornita con annesso il software da utilizzare per l'identificazione e la configurazione di tutti i parametri di rete necessari per il corretto funzionamento: IP installer.
Tale software è utilizzabile solo in ambiente Microsoft Windows, tuttavia è possibile configurare i parametri di rete anche attraverso l'interfaccia web alla quale normalmente si viene rimandati dal software IP Installer, ma che è accessibile facilmente conoscendo l'indirizzo IP assegnato alla camera dal DHCP.
Di seguito vengono riportati alcuni dati tecnici relativi alla videocamera utili per il progetto:

Immagine :
- Dispositivo :
  - 1/3" CCD PS Super-HAD
- Pixel:
  - Totali : 811x508 (NTSC), 795x596 (PAL)
  - Effettivi : 768x494 (NTSC), 752x582 (PAL)

Scansione:
- Frequenza orizzontale : 15,734 Hz (NTSC), 15,625 Hz (PAL)
- Frequenza verticale : 59,94 Hz (NTSC), 50 Hz (PAL)

Risoluzione :
- Orizzontale : 600 linee TV
- Verticale : 350 linee TV

Video:
- Compressione :
  - Codec multiplo H.264 / MPEG4 / MJPEG
  - (H.264/MPEG4 selezionabili)
  - Doppio streaming simultaneo
- Risoluzione :
  - 4CIF : 704x480 (NTSC), 704x576 (PAL)
  - VGA : 640x480
  - CIF : 352x240 (NTSC), 352x288 (PAL)
- Frequenza Fotogrammi : 30,15,8,3,1fps (NTSC), 25,13,6,3,1fps (PAL)

INDIRIZZO IP ASSEGNATO STATICAMENTE: 192.168.1.128

Ambiente di sviluppo

La tesina è stata realizzata in ambiente Qt, attraverso il linguaggio di programmazione C++ e con l'ausilio della libreria OpenCV.

OpenCV è una libreria orientata alla Computer Vision, originariamente sviluppata da Intel, ma attualmente disponibile sotto licenza open source BSD.
È una libreria multipiattaforma ed è quindi compilabile su molti sistemi operativi (Windows, Mac OS X, Linux, PSP, VCRT).
Inoltre essa è sviluppata anche per altri linguaggi di programmazione: C, Python e Java.

Qt Creator, ambiente di sviluppo utilizzato, comprende un debugger visuale e una GUI integrata.
Esso utilizza il compilatore C++ della GNU Compiler Collection su Linux e FreeBSD, mentre in Windows sfrutta MinGW o MSVC.

Approcci possibili per il conteggio delle persone

Esistono diverse soluzioni e diversi approcci per il conteggio delle persone presenti in un determinato luogo.
E' possibile scrivere un programma che utilizzi una semplice telecamera che lavora in luce visibile o utilizzare strumenti più precisi e mirati come, ad esempio, microcontrollori, laser e Arduino.
Inoltre è possibile scegliere di analizzare e rilevare solo persone in movimento o anche persone ferme.
La scelta va fatta ponendo particolare attenzione alle specifiche del progetto e agli svantaggi che accompagnano qualsiasi tecnica si scelga.

Analizzando ad esempio la scelta di usare degli strumenti precisi e dettagliati quali ad esempio i laser, bisogna tener presente il fatto che tali dispositivi devono essere posizionati e configurati molto attentamente e con altissima precisione, mentre utilizzando una qualsiasi telecamera in luce visibile essa può essere posizionata e configurata in poco tempo.

Stato dell'arte

L'elemento alla base del riconoscimento di oggetti in un immagine è il blob, cioè un gruppo di pixel che l'elaboratore identifica come un oggetto.

I blob possono essere quindi associati ad oggetti diversi come mani, volti ecc. e la loro estrazione dalle immagini è stata, ed è tuttora, oggetto di studio per molti sviluppatori e programmatori, soprattutto di quelli che lavorano e sviluppano programmi in OpenCV o Flash (ActionScript3).

Essi hanno sviluppato diversi metodi risolutivi, la cui precisione ha raggiunto livelli elevati, ma tuttora limitati al riconoscimento di singoli oggetti, pertanto gli oggetti non devono essere sovrapposti.

Alcune di queste soluzioni si basano su algoritmi Motion Detection.
La motion detection è il processo per rilevare un cambiamento di posizione di un oggetto rispetto all'ambiente circostante, o le modifiche nei dintorni relativi ad un oggetto.
Ciò può essere ottenuto con metodi meccanici ed elettronici.
Il movimento può essere rilevato tramite:

- Infrarossi (sensori attivi e passivi)
- Ottica (sistemi video e fotocamera)
- Energia delle frequenze radio (radar e motion detection tomografica)
- Audio (microfoni e sensori acustici)
- Vibrazioni (sismica)
- Magnetismo (sensori magnetici e magnetometri).

Un rivelatore di pedoni in movimento viene fornito con le versioni recenti di OpenCV (dalla 2.2 in poi), ed è fruibile ai seguenti percorsi "*modules/objdetect/src/hog.cpp*" e "*samples/cpp/peopledetect.cpp*".

Un approccio ulteriore, invece, prevede la features detection in un'immagine. Essa consiste nel riconoscere oggetti differenti basandosi su classificatori opportunamente addestrati.

Approccio seguito

Dopo un'attenta analisi di obiettivi e strumenti a disposizione, e considerando che nel nostro caso le persone nella stanza sono principalmente ferme, si è scelto di utilizzare un approccio di tipo features detection.
Un processo di riconoscimento di oggetti (persone, nel nostro caso) è generalmente

efficiente se basato sul rilevamento di caratteristiche che includano informazioni aggiuntive riguardo la classe di oggetti da rilevare.
In questa tesina si sono utilizzate le Haar-like features.

Un Haar-like feature considera regioni adiacenti di forma rettangolare poste in una particolare posizione nella finestra in cui si sta facendo il riconoscimento. In ognuna di queste regioni vengono sommate le intensità dei pixel e calcolate le differenze tra queste somme.
Queste differenze vengono utilizzate, poi, per categorizzare le sottosezioni di un'immagine. Per esempio, se guardiamo una faccia umana, possiamo osservare che, per qualsiasi faccia, la regione degli occhi è più scura di quella delle guance. Quindi una haar feature comune per il riconoscimento di un volto è un insieme di 2 rettangoli adiacenti che coprono la regione degli occhi e quella delle guance.
La posizione dei rettangoli è definita all'interno di un rettangolo più grande che conterrà l'intera faccia.

Dato che il contesto preso in esame è quello delle aule universitarie si è deciso di utilizzare come caratteristiche per individuare le persone presenti in aula le seguenti parti del corpo umano:

1. corpi interi
2. parte superiore del corpo
3. volti.

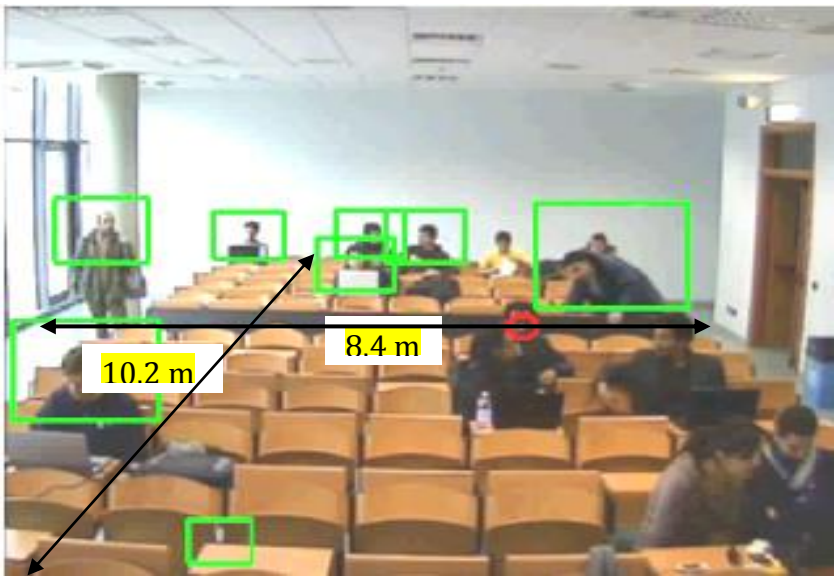
Tali caratteristiche sono estratte tipicamente utilizzando un classificatore a cascata (*Cascade Classifier*) che deve essere "addestrato" in modo da riconoscere con precisione oggetti differenti.
Ai fini di questa tesina sono stati utilizzati i classificatori già addestrati messi a disposizione da OpenCV :

- haarcascade_frontalface_alt.xml, per individuare volti di persone di cui si vede solo la testa (poiché le altre parti del corpo sono nascoste da altre persone o altri oggetti);
- haarcascade_upperbody.xml, per individuare la parte superiore del corpo di persone sedute e non;
- haarcascade_fullbody.xml, per individuare l'intero corpo di persone in piedi.

Modalità di ripresa (scenario)

Il conteggio delle persone è stato effettuato in "luce visibile", posizionando la telecamera in punto alto della stanza.
Per le prove la telecamera è stata posta ad un'altezza di 2,13 metri da terra, con un angolo di inclinazione di 12°.

Area visibile negli esperimenti:



Funzionamento

Abbiamo implementato il conteggio delle persone presenti in una stanza attraverso i classificatori Haar, già disponibili in opencv, per full body, upper body e face.

Essi vengono applicati in cascata in questa maniera:
I classificatori vengono sempre applicati all'intera immagine, ma si evita di contare la stessa persona più volte con più classificatori diversi.

Come funziona:
Prima vengono applicati i classificatori per il full body all'intera immagine e vengono conteggiate le persone riconosciute.
Nel passo successivo vengono applicati, sempre all'intera immagine, i classificatori per l'upper body, vengono conteggiate le persone riconosciute e poi si controlla che esse non siano già state contate. Qualora lo fossero, il contatore viene decrementato in modo da eliminare i "doppioni".

Il controllo del "doppione" viene effettuato nel seguente modo:
se il box riconosciuto è presente all'interno di un box individuato al passo precedente (il controllo non è rigido: abbiamo inserito una soglia pari ad 1/5 della dimensione del box), allora esso viene riconosciuto come "doppione" e viene decrementato il contatore generale delle persone riconosciute.

Successivamente si esegue la stessa operazione per le facce frontali, controllando che non siano contenute né all'interno di upper body, né all'interno di full body.

Sono stati utilizzati dei puntatori per tenere traccia della parentela tra upper body e full body, in modo da non decrementare più volte il contatore per la stessa persona:

se tramite il riconoscimento delle facce frontali viene riconosciuta una persona nella posizione (x,y) che è sia all'interno di un upper body (che chiamiamo A), sia all'interno di un full body (B) che contiene l'upper body A, allora il contatore generale deve essere decrementato solo 1 volta e non 2 come si farebbe se non si sapesse che l'upper body (A) è imparentato con il full body (B).

Inoltre si è deciso di non utilizzare i classificatori per il riconoscimento di facce laterali, poiché, dopo diverse prove di funzionamento si è appurato che venivano riconosciuti esclusivamente dei falsi positivi: Non c'è stato nessun caso in cui veniva riconosciuta correttamente una faccia posta lateralmente.

Implementazione

L'algoritmo vero e proprio di conteggio delle persone è stato inserito all'interno di un file di header (conta_persone.h), in modo da poter essere utilizzato anche al di fuori del nostro programma.

L'applicazione sviluppata prevede 2 modalità di interazione:
- una "grafica", in cui vengono mostrate l'immagine catturata e il numero di persone riconosciute;
- una che stampa sul terminale il numero delle persone trovate nell'area visibile.

In entrambi i casi il numero rilevato viene scritto anche su di un file (conteggio.txt) insieme alla data e all'ora del rilevamento.

Per una maggiore precisione si è preferito effettuare il conteggio basandosi su 3 immagini successive (distanziate di un tempo pari a quello necessario per la corretta acquisizione di una nuova immagine): vengono prelevate 3 immagini, effettuati i calcoli su tutte e 3 (e memorizzati) e poi il risultato finale è calcolato come media aritmetica delle 3 rilevazioni.

Per accedere alla modalità "grafica" è possibile avviare il programma aprendolo:

a) da terminale senza passare alcun parametro
b) cliccando 2 volte sull'eseguibile

Invece per accedere alla modalità "terminale" è possibile avviare il programma passando il parametro "-t" oppure "-T".

Nella modalità "grafica" è possibile specificare l'indirizzo IP della videocamera all'interno dell'apposito campo.
Nella modalità "terminale", invece è possibile specificarlo tramite il parametro: "-i indirizzo" (ad esempio << –i "http://192.168.1.1" >>)
ATTENZIONE: L'indirizzo deve essere specificato tra virgolette (esempio: -i "http://www.nostro_progetto.it/camera.jpg")

L'indirizzo di default è
http://admin:4321@192.168.1.128/cgi-bin/video.cgi?msubmenu=jpg

Username e password:
Qualora la videocamera preveda l'inserimento di uno username e di una password è possibile inserirle nell'indirizzo in questo modo:

http://**NOME_UTENTE**:**PASSWORD**@dominio

(
Come esempio basti guardare l'indirizzo di default:
l'indirizzo della camera è
http://192.168.1.128/cgi-bin/video.cgi?msubmenu=jpg

ma abbiamo inserito come nome utente "admin" e come password "4321"
)

Problemi ed esigenze implementative:

1. Da quanto riportato su diversi siti di settore (http://blogs.msdn.com/b/oldnewthing/archive/2009/01/01/9259142.aspx), alcune versioni di Windows non supportano in maniera appropriata le applicazioni "dual mode", cioè quelle che a seconda di come vengono avviate, fanno partire l'interfaccia grafica o l'interfaccia per console.
   Per tale motivo, in alcuni casi, anche se il programma viene avviato con il doppio click, oltre all'interfaccia, viene aperta una finestra di terminale.

2. Utilizzando la videocamera in dotazione, ogni volta che si richiede un'immagine alla camera, è necessario ristabilire la connessione passando nome utente e password. Pertanto nel codice si troverà il rilascio della risorsa video ogni volta che si finisce un'elaborazione, per poi riacquisirla nel passaggio successivo.

Algoritmo per il conteggio di oggetti in dotazione con la videocamera

La videocamera di rete Samsung SND-3080C che abbiamo utilizzato per fare tutte le rilevazioni, è dotata di un software interno in grado di contare il numero di oggetti inquadrati in ogni istante di tempo.
La rilevazione ha 2 modalità di funzionamento:

1. Conteggio di oggetti che attraversano una soglia stabilita;
2. Conteggio di oggetti presente in un'area ben definita.

La prima modalità di funzionamento si divide a sua volta in 2 sottofunzioni: una calcola il numero di persone che attraversano la soglia da destra verso sinistra, e l'altra invece quelle che attraversano la soglia da sinistra verso destra.

Essa è molto utile in tutti i casi in cui si posizioni la camera sopra ad una porta o un ingresso di una stanza, in quanto conta quanti oggetti/persone attraversano l'entrata e quindi permette di sapere in tempo reale quante persone sono entrate e uscite dalla stanza.
Tuttavia, essa funziona male qualora la telecamera non venga posta perpendicolarmente al terreno e qualora più persone/oggetti attraversino la soglia senza essere distanziate l'una dall'altra.

La seconda modalità di funzionamento, invece, richiede la definizione di un poligono (con al massimo 11 lati) di forma e dimensione prefissata. Tali parametri (forma e dimensione) saranno quelli di riferimento per il conteggio: in pratica è necessario specificare una forma più o meno standard che gli oggetti da identificare potranno avere e l'algoritmo conterà solo gli oggetti che avranno più o meno quella dimensione e quella forma.
Se ad esempio si definisce la forma corrispondente ad un uomo adulto difficilmente verranno contati anche i bambini.
Inoltre, anche per questa seconda modalità è necessario che la telecamera sia posta perpendicolarmente al terreno.

E' possibile accedere a tali funzioni tramite l'interfaccia web della videocamera, o salvare i risultati sulla scheda di memoria inserita nella videocamera, oppure inviare i dati tramite FTP.

Confronto con la nostra tecnica

Non è possibile stabilire a priori se il nostro algoritmo sia migliore di quello interno alla videocamera.
Nonostante il conteggio degli oggetti in dotazione con la videocamera funzioni molto bene nel caso in cui la videocamera venga posta perpendicolarmente al pavimento sopra all'ingresso in una stanza, i risultati peggiorano notevolmente quando la telecamera viene posta in un angolo della stanza allo scopo di riprendere un'area più grande possibile.

Inoltre, è possibile far notare che, il nostro programma, al contrario dell'algoritmo interno alla videocamera, permette di distinguere una persona da uno scatolone avente all'incirca la stessa occupazione spaziale.

Pertanto, per gli scopi di questo progetto, è molto più utile utilizzare il programma sviluppato piuttosto che quello interno alla videocamera.

Risultati ottenuti e limiti

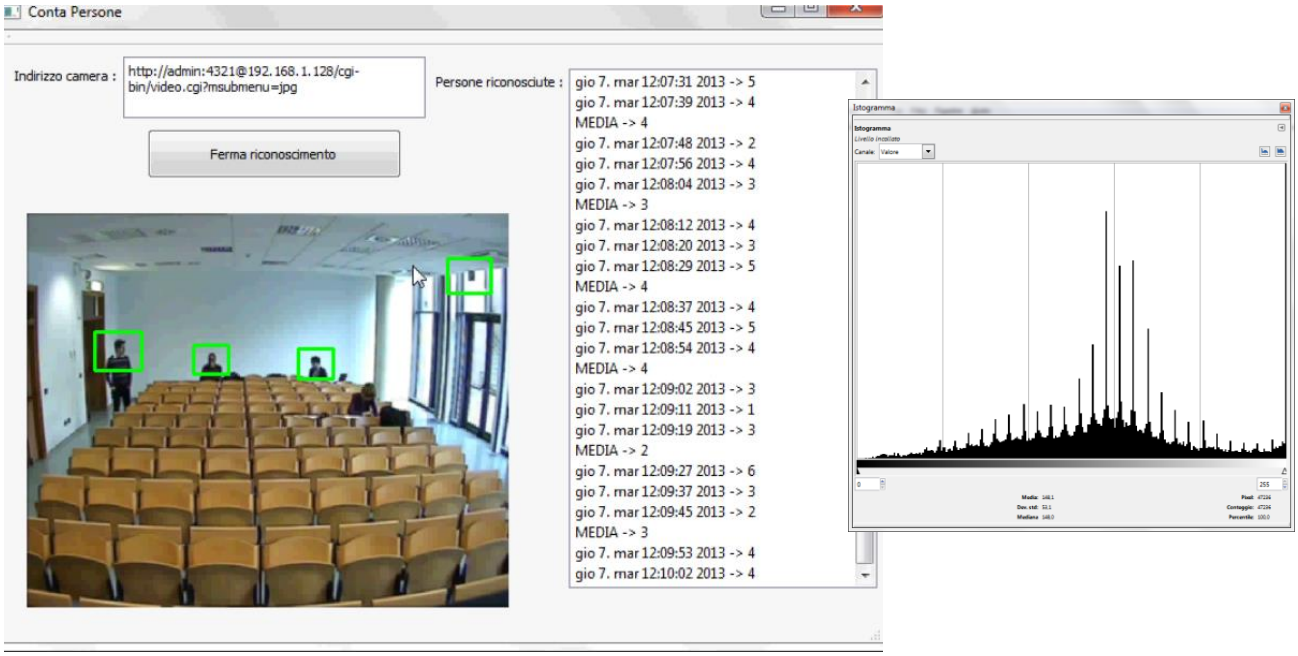
Abbiamo riscontrato diverse anomalie nel riconoscimento di persone.
Innanzitutto, quando ci sono molte persone una vicina all'altra l'algoritmo tende a riconoscerne solo una.
Inoltre, avendo impostato i parametri della funzione DetectMultiScale in modo da riconoscere anche persone lontane, succede che l'algoritmo riconosca come facce o upperbody elementi costituiti da piccole parti simili ad un corpo umano.

E' di fondamentale importanza la luminosità della stanza.
Nei video allegati è infatti possibile verificare che gli oggetti meglio illuminati vengono riconosciuti più facilmente di quelli non illuminati.

Di seguito vengono riportati 4 estratti dei video allegati in cui è possibile notare le differenze nel riconoscimento dipendenti dalla luminosità:

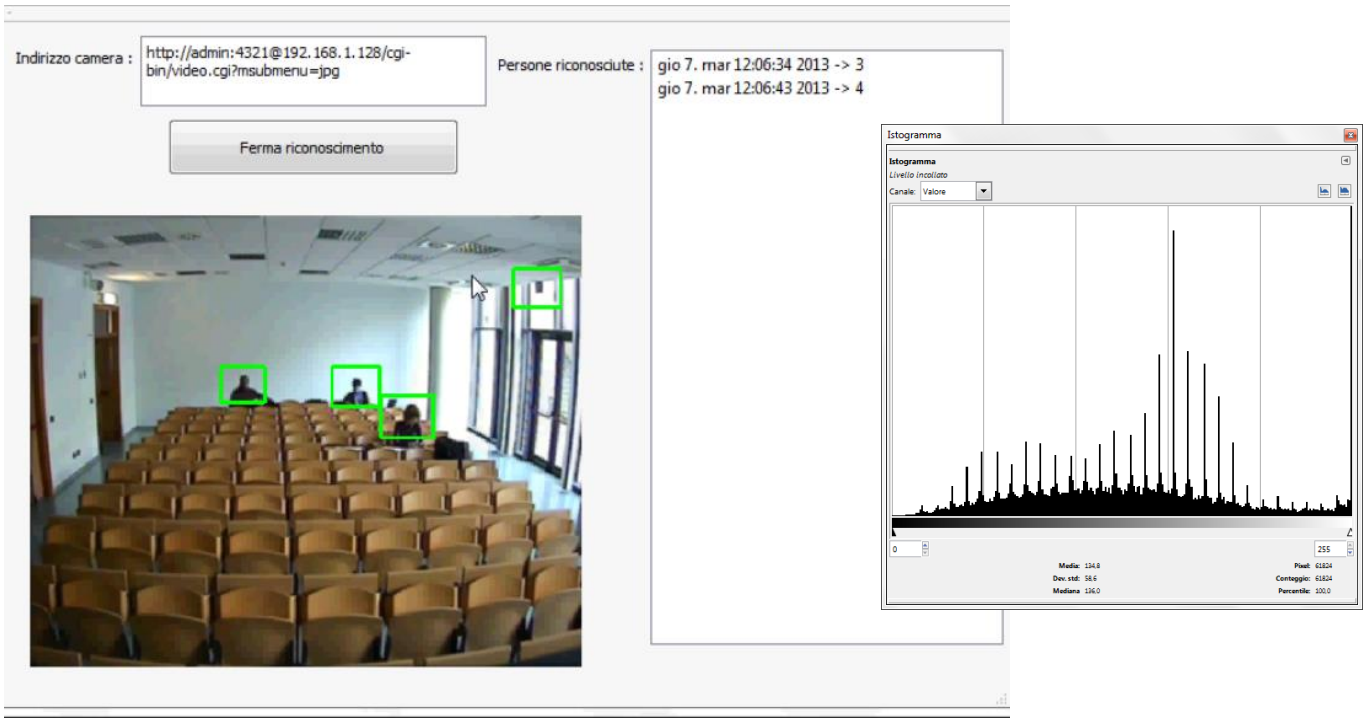
Differenze di luminosità

Altissima luminosità (finestre aperte e luci accese)



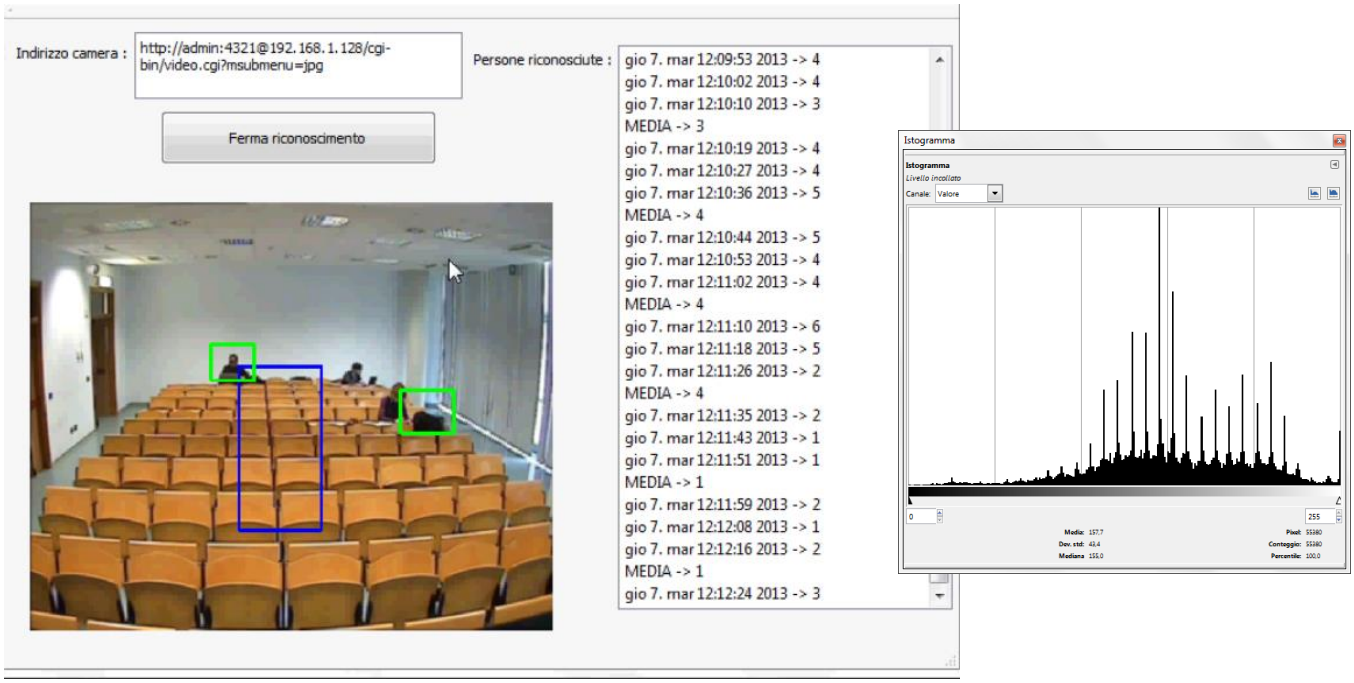
In questa immagine c'è molta luce proveniente dall'esterno e come è possibile notare, vengono riconosciute 3 persone correttamente su 4 e in più però viene riconosciuto un oggetto da non riconoscere.

Alta luminosità (finestre aperte e luci spente)



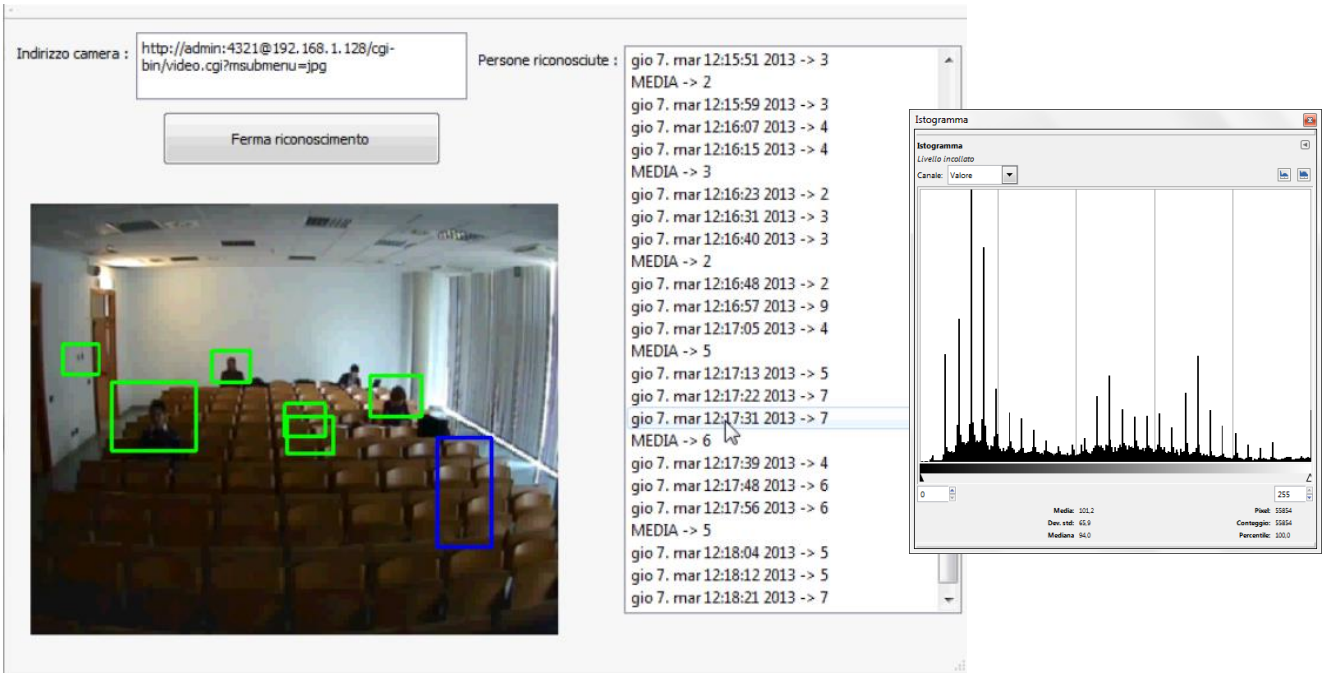
In questo frame la luminosità è buona ed infatti vengono riconosciute 3 persone su 3. L'unico difetto è che oltre ai tre riconosciuti correttamente c'è un falso positivo, ma è in una zona in cui la luce è troppa.

Media luminosità (finestre chiuse e luci accese)



In questa immagine la luce è ben distrubuita ma poca soprattutto nelle zone periferiche della stanza! Infatti viene riconosciuto correttamente l'individuo posto in fondo in una zona abbastanza illuminata ma non gli altri 2 posti nelle zone più periferiche.
Inoltre al centro è possibile notare un falso positivo.

Bassa luminosità (finestre chiuse e luci spente)



In quest'ultima immagine, in cui la luce è scarsa, nonostante vengano riconosciute 3 persone su 4, è possibile notare la presenza di molti falsi positivi.

In allegato vengono forniti 3 video riportanti i risultati ottenuti :
dimostrazione_GUI.avi -> dimostrazione del funzionamento con interfaccia grafica
dimostrazione_console.avi -> dimostrazione del funzionamento con console
video_luminosita.avi -> dimostrazione delle diverse configurazioni di luminosità

Accuratezza del riconoscimento

In una stanza in cui sono presenti 5 persone ben distribuite su tutto lo spazio disponibile e abbastanza ferme, l'algoritmo riconosce tra le 4 e le 5 persone.

Quando più persone sono vicine e tendono a sovrapporsi, l'algoritmo è portato a riconoscerle come un'unica persona se non proprio nessuna.

Quando il numero di persone aumenta l'algoritmo tende a riconoscerne poco più della metà: abbiamo fatto la prova con 11 persone e ne riconosceva tra 6 e 9.

Possibili sviluppi (miglioramenti)

Modifica dei parametri della funzione DetectMultiScale

Volendo si potrebbero modificare i parametri della funzione DetectMultiScale (scaleFactor e minSize) in modo da riconoscere bene gli oggetti vicini ma non quelli lontani.

Migliore qualità

Inoltre, si potrebbe provare ad utilizzare una telecamera con una qualità migliore di quella fornitaci, tuttavia, dal momento che i classificatori Haar si basano sulla differenza di luminosità, provando a sfocare un po' l'immagine acquisita i risultati non sono peggiorati moltissimo.

Conclusioni

A conclusione del presente lavoro di tesina, possiamo affermare che l'approccio utilizzato, cioè l'utilizzo di una telecamera in luce visibile per il riconoscimento di persone tramite classificatori in una stanza non è abbastanza efficace ed affidabile da poter essere utilizzato in casi in cui la precisione richiesta sia elevata: se si lavora per fasce (distinzione tra 10 e 20 persone), il nostro algoritmo può anche andare bene.
Forse l'utilizzo di una telecamera ad infrarossi potrebbe migliorare le cose, ma ovviamente l'algoritmo da utilizzare sarebbe diverso: non bisognerebbe più riconoscere delle forme, ma dei colori diversi.

Sitografia

www.andol.info/hci/1859.htm -> A review of people counting using OpenCV
http://en.wikipedia.org/wiki/Haar-like_features -> Haar-like features
http://www.provision-cctv.com/picts/pdf2011_761.pdf -> specifiche telecamera
http://www.samsungsecurity.com/product/product_view.asp?idx=6397#FL010000 -> Manuale telecamera in italiano
http://docs.opencv.org/
http://qt-project.org/doc/qt-4.8/