Introduzione alla localizzazione di oggetti

VISIONE ARTIFICIALE

dott. Alessandro Ferrari

Social Q&A

@vs_AR
#askVisionary
www.vision-ary.net

Obiettivi delle prossime lezioni

1. Fornire lessico di base per comprendere le differenze tra localizzazione, tracking e riconoscimento di oggetti «rigidi».
2. Approfondire le strategie di localizzazione e tracking di oggetti per sistemi reali, possibilmente in ambito «mobile».
3. Implementare in MATLAB un esempio di localizzatore di volti.
4. Implementare in OpenCV un sistema di addestramento che possa fornire un modello per la localizzazione di automobili «in-the-wild», primo approccio ai sistemi di guida senza conducente.

Detection and Tracking

Obiettivo: localizzare la presenza del volto come indicatore di presenza umana, «inseguirlo» nel tempo mantenendo una coerenza spazio-temporale tra i frame elaborati. Generalmente viene apposta una «bouding box» sul viso localizzato.

Situazione di riferimento: videocamera posizionata in un punto di interesse, ad esempio nello schermo di un laptop nel caso di una webcam.

Risultato: Il sistema di tracciamento (face tracking) è il risultato dell'integrazione di diversi moduli che devono lavorare in tempo reale nonostante la complessità dell'analisi in atto.

Ambiti di applicazione: Video-Sorveglianza (face recognition) | Gaming (Kinect) | Marketing & Advertising (occhiali da sole interattivi, analisi clientela, analisi pubblicità, digital signage) | Medicale (gaze tracking) | Elettronica di consumo (cellulari, macchine fotografiche) | Arte interattiva

Tassonomia del task di L-T-R.

Localizzazione	Tracking	Riconoscimento
Scansionamento dell'intera immagine alla ricerca di una sotto-regione che dia "matching" positivo rispetto ad un modello.	Scansionamento di una sotto-regione dell'imagine dove "probabilmente" l'oggetto di interesse si trova.	Confronto tra oggetti individuati e modelli degli oggetti da riconoscere (identità di persone, categorie di automobile, ecc.).
Nessuna informazione a priori sulla posizione dell'oggetto. E' necessario avere un modello dell'oggetto per localizzarlo.	Informazione a priori sulla posizione dell'oggetto e (in genere) modello di spostamento. Non è necessario avere un modello dell'oggetto.	E' necessario individuare alcune caratteristiche di "allineamento" dell'oggetto per agevolare il riconoscimento. Modello necessario.
Risultato: bounding box apposta sull'oggetto, oggetto "anonimo".	Risultato: bounding box aggiornata sull'oggetto, identità mantenuta.	Risultato: classe di appartenenza o verifica dell'identità.
Peso computazionale: alto.	Peso computazionale: basso/medio.	Peso computazionale: N/A (offline).

Esempio di un sistema reale

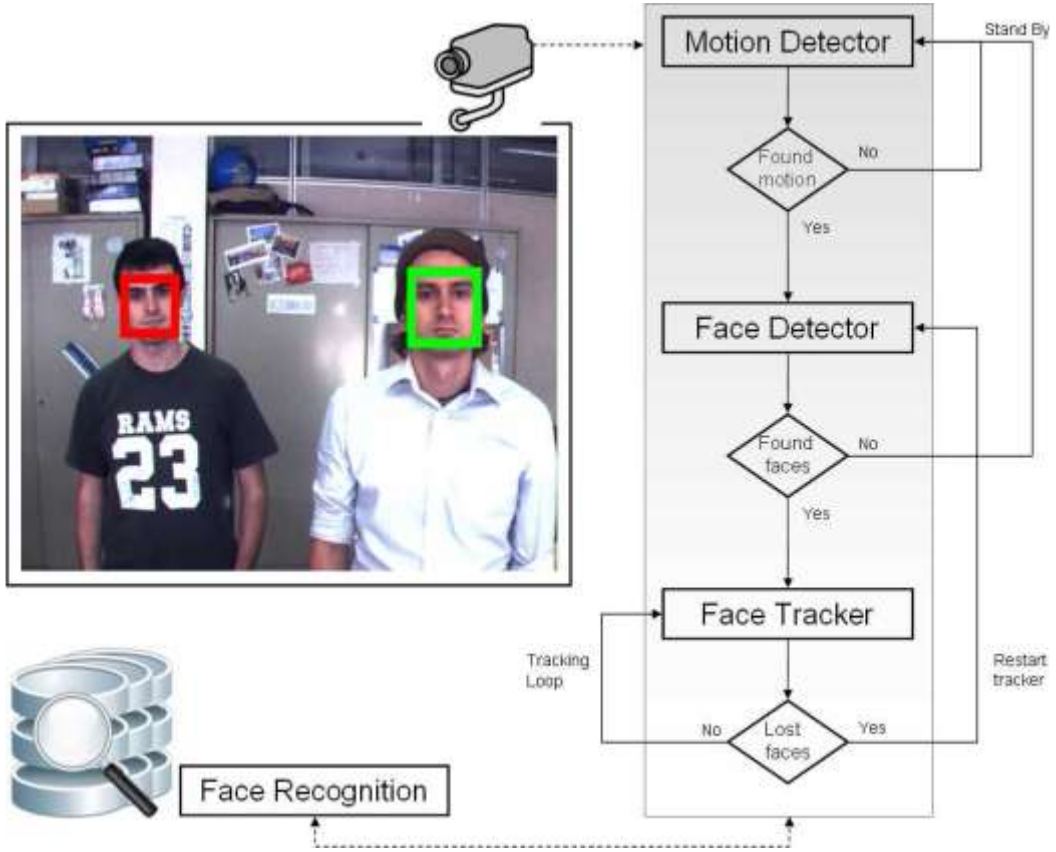


Diagramma di un sistema di video sorveglianza, nei sistemi più moderni (<5 anni) il motion detection è stato eliminato poiché non più necessario (estremizzazione dell'efficienza computazionale dei sistemi di face detection).

Il sistema è composto da quattro moduli principali. I primi tre lavorano in locale sui dati che la camera di sorveglianza fornisce. Il modulo di face recognition invece interroga un database in remoto per recuperare l'identità delle persone tracciate. Questa attività avviene offline, non è richiesta particolare efficienza computazionale

Face detection e tracking: perché?

1. La localizzazione del volto è un chiaro esempio di «back-end» di una interfaccia interattiva moderna. E’ quindi propedeutico a introdurre le problematiche e le soluzioni relative alla localizzazione di oggetti.

2. Il volto è un indicatore certo della prossimità umana:
   - stimatore di attenzione visiva;
   - è un modulo di controllo dell'interfaccia molto potente.

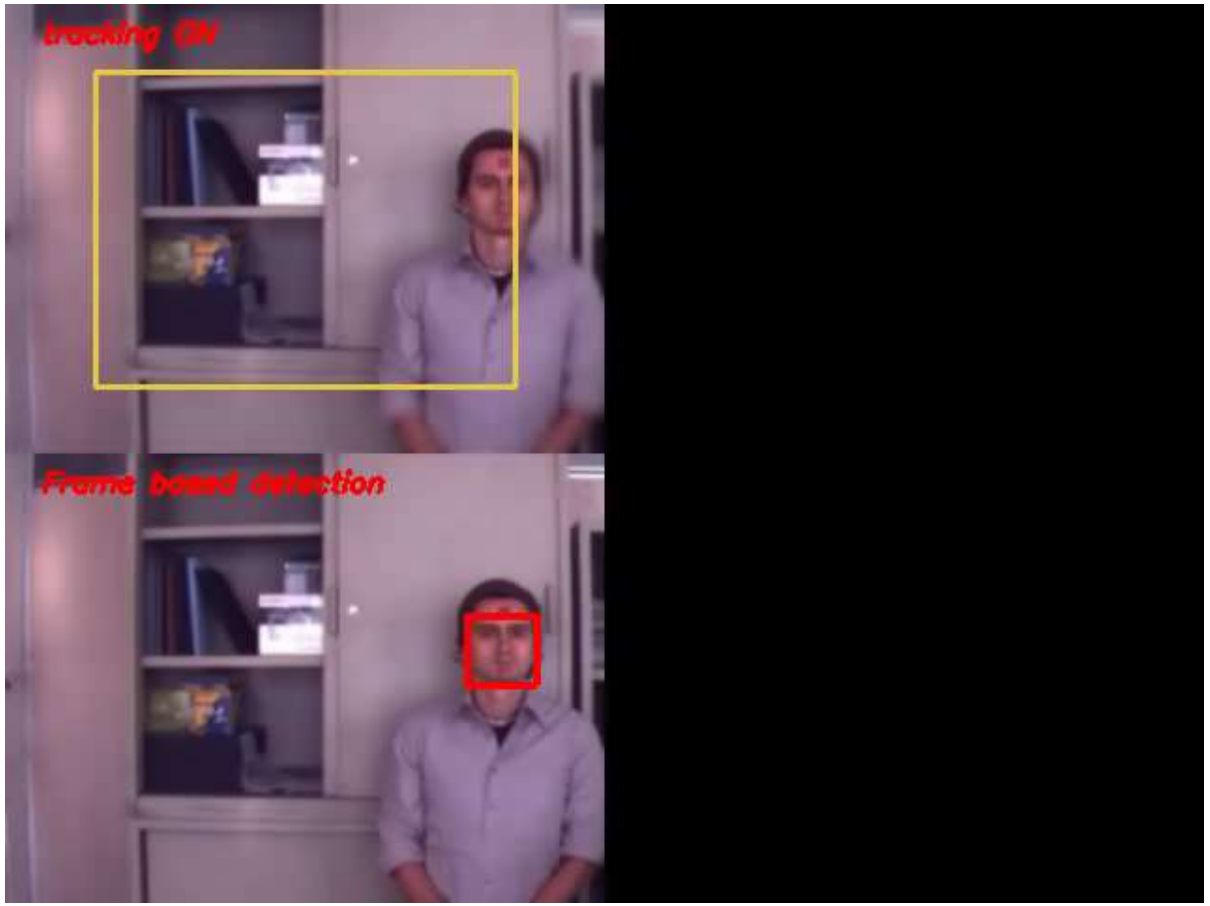
3. L'addestramento di un localizzatore di volti è rappresentativo del problema dell'addestramento (in conclusione a questo ciclo di lezioni).

Object Tracking: possibili approcci

1. Tracking Frame by Frame: il tracking è emulato. Si applica continuamente (senza coerenza spazio-temporale) un localizzatore di oggetti e si aggregano i risultati nel tempo.
2. Tracking by likelihood: tramite una misura di verosimiglianza si rafforzano le ipotesi di tracciamento più consistenti per mantenere una traccia coerente nello spazio-tempo.
3. Tracking by detection: la misura di verosimiglianza è data dallo stesso localizzatore di volti utilizzato in maniera «innovativa».

Queste definizioni non sono formalmente corrette ma rendono l'idea sui possibili approcci che si possono utilizzare per tracciare un oggetto. Nella fattispecie, il tracking by detection è riconducibile al caso del tracking by likelihood ma «storicamente» ha introdotto un approccio innovativo in letteratura e per questo si è preferito presentarlo separatamente.

Face Tracking: possibili approcci

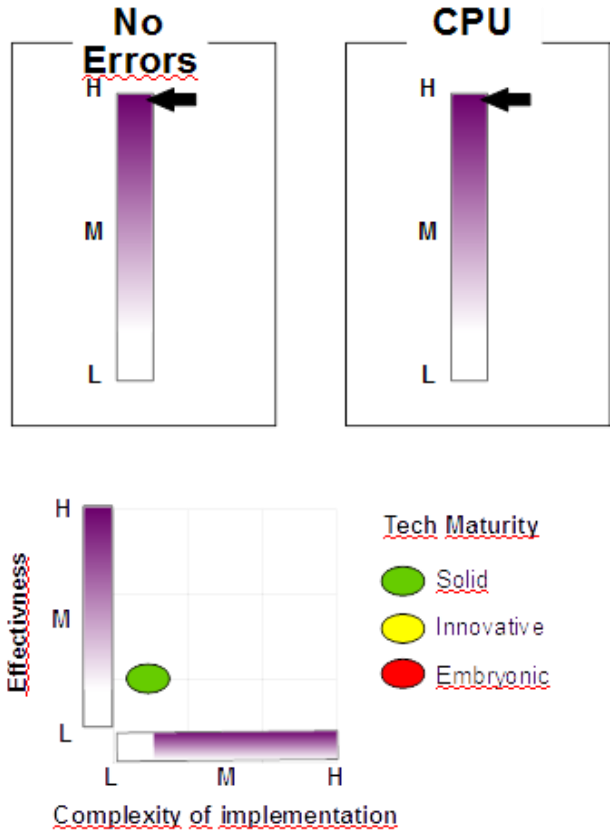


Tracking frame by frame

- Questo approccio si basa su una ricerca frame a frame dei volti presenti nelle immagini che compongono la sequenza video non tenendo conto della correlazione spazio-temporale che è naturalmente insita in una sequenza di immagini.

- Si noti inoltre che tale approccio non permette di realizzare un tracciamento in senso stretto poiché, isolando l'analisi all'interno di ogni singolo frame senza inferire la posizione dell'oggetto di interesse al frame successivo, viene meno la capacità di analizzare la coerenza spazio-temporale del movimento dell'oggetto.

- Le tecniche basate su questo approccio spesso presentano un output simile a quello di un sistema di tracking, poiché i rilevamenti effettuati frame a frame «emulano» il comportamento di un sistema di tracciamento, ma di fatto non lo realizzano realmente. E' necessario aggiungere un ulteriore livello di intelligenza.

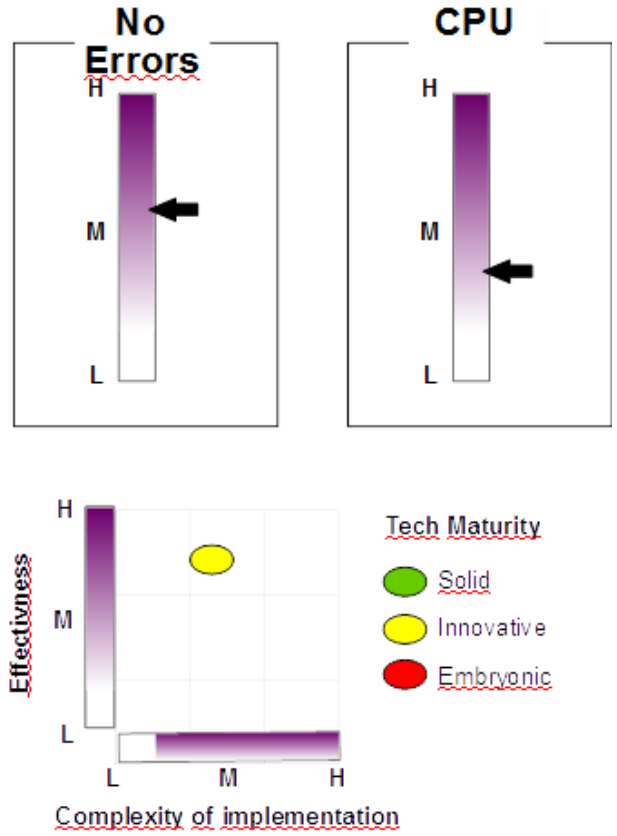


Tracking by likelihood

- Questo approccio si basa su una ricerca dell'oggetto nel frame solo al «tempo zero». Dal frame successivo si utilizza un meccanismo di «propagazione» della posizione dell'oggetto. Nel caso di framework probabilistici la conferma della previsione avviene tramite la verosimiglianza dell'osservazione.

- A differenza del precedente approccio (che utilizza solo informazione all'interno di un singolo frame), questa modalità sfrutta la correlazione della sequenza di immagini introducendo il concetto "logico'' di traccia, come evoluzione della posizione di un oggetto nello spazio e nel tempo. La correlazione spazio-temporale è sfruttata sia per mantenere una logica di tracciamento che per minimizzare il costo computazionale .

- Si noti che:
  - se la misura di verosimiglianza è «smart» si ha flessibilità nel tracking.
  - Generalmente si ha indipendenza dal tipo di oggetto tracciato.
  - Spesso è difficile trovare una misura «leggera».



Tracking by likelihood: the TLD

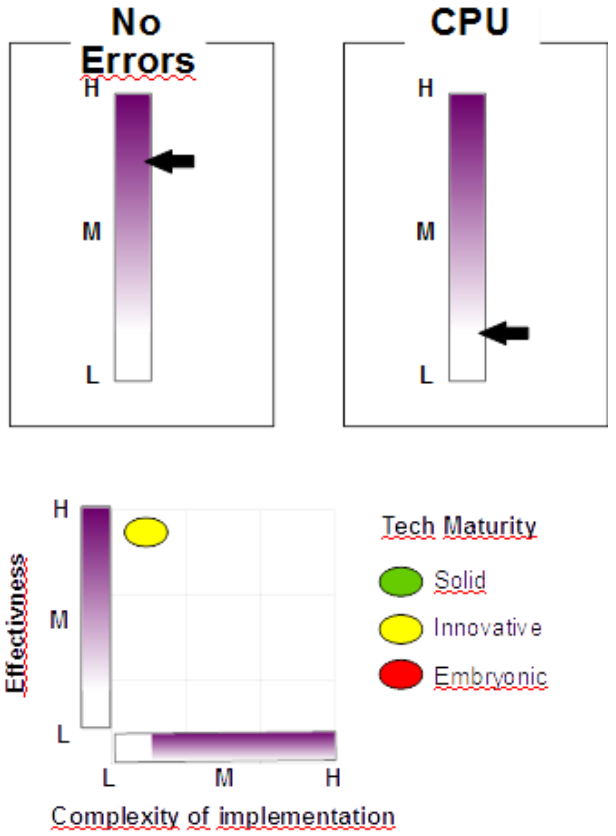
https://www.youtube.com/watch?v=W2qR60hrD2w

Tracking by detection

- E’ un caso particolare del tracking by likelihood.

- Questo approccio si basa su una ricerca dell’oggetto nel frame solo al «tempo zero». Dal frame successivo si utilizza un meccanismo di «propagazione» della posizione dell’oggetto. La verifica della verosimiglianza avviene tramite lo stesso meccanismo di inizializzazione del tracking, ovvero il face detector.

- Si noti che:
  - Il tracking è meno flessibile poiché solo oggetti «riconoscibili» dal detector possono essere anche tracciati (NO posa e rotazione in piano).
  - La verosimiglianza non richiede altre implementazioni addizionali, va solo «estratta» dal sistema di detection.
  - Se il localizzatore di oggetti non fornisce una stima probabilistica (ma solo un responso T/F) è necessario trovare una misura adeguata.
  - Totale dipendenza dal tipo di oggetto tracciato.
  - Efficienza esasperata, moltiplicatore 50-100x.



Object Tracking: da dove si parte?

Indipendentemente dall'approccio utilizzato è necessario localizzare il volto almeno al «tempo zero» per l'inizializzazione del sistema. L'inizializzazione può avvenire:

- Manualmente (app per cellulari, siti internet, ecc...). Viene chiesto all'utente, ad esempio tramite il mouse, di inserire una bounding box contenente la faccia o la posizione di occhi e bocca.

- Automaticamente, attraverso un sistema di localizzazione volti vero e proprio che in maniera trasparente all'utente localizza il volto e inizializza il sistema di tracking.

Il primo caso è ovviamente di scarso interesse. Per quanto riguarda il secondo, lo stato dell'arte è rappresentato dall'algoritmo di Viola/Jones.

Localizzatore di volti: Viola & Jones

L'algoritmo di Viola e Jones (2001) è lo standard de facto per la localizzazione del volto. Rispetto ai suoi (superati e oramai desueti) predecessori introduce tre elementi di novità:

1. Utilizzo delle **Haar features** in combinazione con una nuova rappresentazione dell'immagine detta **Integral Image**. Le features hanno basso costo computazionale e la nuova struttura dati permette di effettuare l'analisi in tempo costante indipendentemente dalla dimensione delle regioni analizzate.

2. Viene introdotto un metodo di selezione di feature di Haar attraverso l'algoritmo **AdaBoost** di Freud e Shapire (1995). Questa strategia permette di eliminare in addestramento la maggior parte delle feature di scarsa capacità discriminante e selezionare solo quelle più efficaci per il problema.

3. Viene introdotta una nuova strategia di analisi dell'immagine basata su **struttura a cascata** dove ogni livello della cascata è un classificatore creato con AdaBoost. La complessità dei livelli cresce man mano che si procede verso la fine della cascata. Le regioni «facili» vengono scartate velocemente ai primi stadi, quelle più «difficili» sono sottoposte a più livelli di verifica. Qualora una regione superi tutti gli stadi viene etichettata come regione contenente una faccia.

Localizzatore di volti in azione

https://vimeo.com/12774628

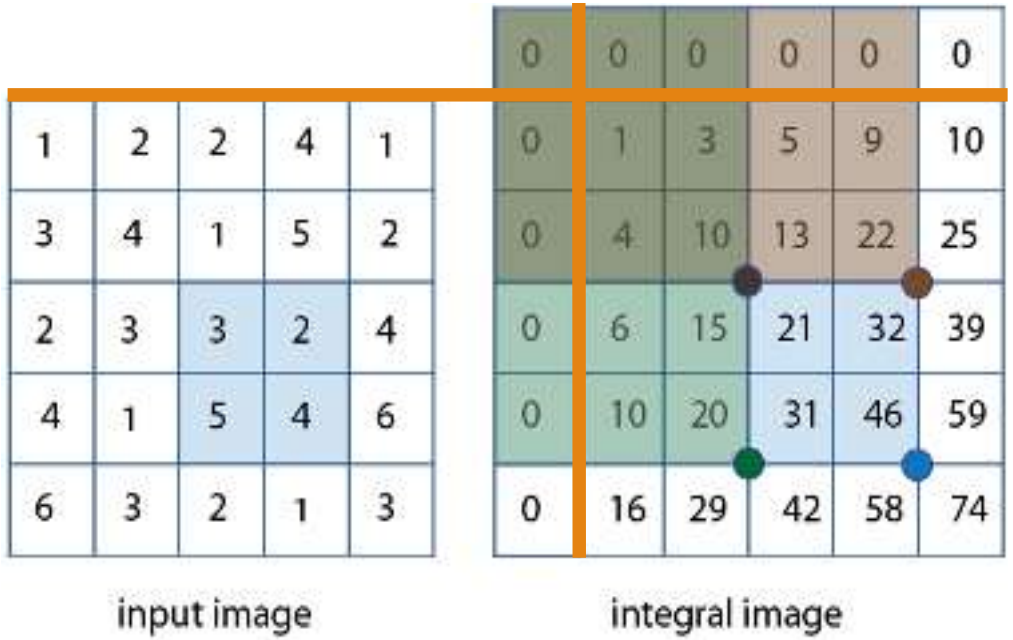
La «Integral Image»

- L'immagine integrale, che ha le stesse dimensioni dell'immagine originale (non in OpenCV), nella posizione di indice (x,y) contiene la somma dei livelli di grigio di tutti i pixel precedenti.

$$II(x_0, y_0) = \sum_{x=0}^{x_0} \sum_{y=0}^{y_0} I(x, y);$$

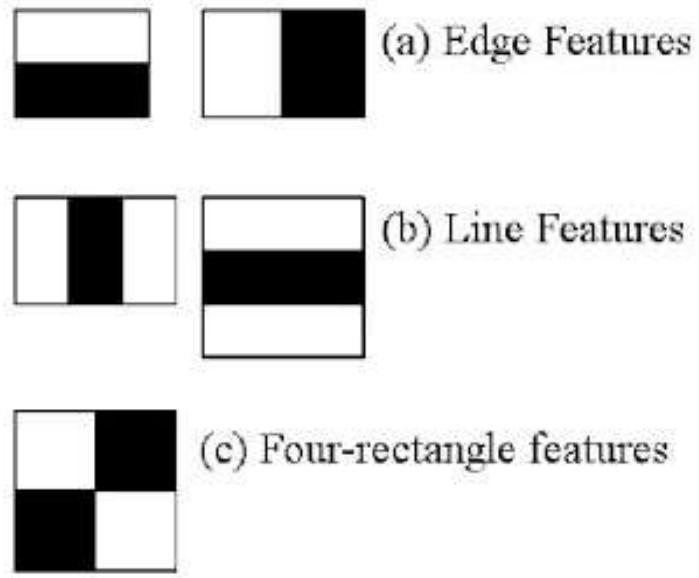
- **Efficienza del calcolo con la integral image**. Per calcolare la somma dei livelli di grigio dei pixel contenuti nel rettangolo sono sufficienti solo quattro riferimenti all'integral image.

- **Indipendenza delle feature dalla dimensione dell'immagine.** Che la feature di Haar da calcolare sia di 2 pixel (la più piccola possibile) o della dimensione dell'immagine, il calcolo ha la stessa complessità computazionale.



Le feature di Haar

- Il metodo ideato da Viola e Jones classifica le regioni di una immagine basandosi su valori calcolati attraverso le Haar features.
- Storicamente le feature sono di tre tipi. Per gestire il problema della rotazione in piano di oggetti sono state introdotte le «tilted» (oramai desuete).
- Basso costo computazionale, grazie soprattutto alla Integral Image.
- Le feature, oltre al tipo, si differenziano per dimensione e posizione all'interno della detection window (la finestrella di ricerca in evidenza nel video precedente).
- Il valore di una feature, una volta applicata ad una regione dell'immagine, è dato dalla differenza normalizzata tra i livelli di grigio dei pixel appartenenti ai rettangoli bianchi rispetto a quelli neri
- http://docs.opencv.org/2.4.9/modules/objdetect/doc/cascade_classification.html - Le feature di Haar nella implementazione OpenCV



Dalle Feature di Haar al classificatore

- In una detection windows di 24x24 pixel, si possono generare decine di migliaia di feature di Haar a fronte di solo 576 pixel; sebbene le feature di Haar siano molto efficienti, non è pensabile applicare tutto l'intero di set in maniera esaustiva sull'immagine poiché tale pratica richiederebbe tempi di calcolo proibitivi.

- Viola e Jones dimostrano che al fine di individuare le regioni contententi gli oggetti di interesse è sufficiente applicare un piccolo sotto-insieme di feature fortemente discriminanti; la chiave dell'algoritmo di Viola e Jones sta quindi nell'individuare queste feature e organizzarle in maniera gerarchica sfruttandone (in maniera incrementale e solo su necessità) la relativa capacità discriminativa. Il risultato di questo procedimento di scelta e organizzazione delle feature produce un «oggetto» chiamato classificatore. E' intuitivo che, dato un problema di classificazione, vi siano feature molto discriminanti e feature sostanzialmente inutili. Quelle «inutili» vengono ignorate priori, tra quelle «utili» invece si procede ad una sorta di organizzazione gerarchica che ne determina la priorità di utilizzo.

- La scelta delle feature si realizza attraverso una procedura di apprendimento basata su una variante del boosting. Tale variante, detta AdaBoost, addestra il classificatore presentandogli, più volte e secondo uno schema iterativo, una sequenza di esempi etichettati (faccia o non faccia). Ad ogni iterazione viene modificata l'importanza di ciascun esempio in modo da enfatizzare gli esempi "difficili" (esempi classificati erroneamente) così da migliorare le performance del classificatore nella successiva iterazione. Al termine delle iterazioni, il classificatore è una combinazione lineare pesata delle feature più discriminanti.

L’addestramento: strong & weak classifier

L'algoritmo alla base della costruzione e addestramento del classificatore usato da Viola e Jones è una variante di AdaBoost, algoritmo proposto nel 1995 da Y. Freund e R. Shapire. Nella sua versione originale, AdaBoost viene utilizzato per il boosting delle prestazioni di un qualsiasi algoritmo di apprendimento.

L'obiettivo è quello di costruire un classificatore forte (strong classifier) combinando tra loro classificatori deboli (weak classifier). L'algoritmo di boosting consiste nell'addestrare ciascun classificatore debole presentandogli, più volte e secondo uno schema iterativo, una sequenza di esempi etichettati (faccia o non faccia). Ad ogni iterazione viene modificata l'importanza di ciascun esempio in modo da enfatizzare gli esempi difficili (cioè gli esempi classificati erroneamente) così da migliorare la performance del classificatore debole nella successiva iterazione. Al termine delle iterazioni, il classificatore forte sarà rappresentato da una combinazione lineare pesata (w) di una selezione di classificatori deboli (h).

In fase di testing, il classificatore binario di tipo AdaBoost associa a un esempio x un'etichetta in accordo con il segno della somma pesata.

$$F(\mathbf{x}) = \sum_i w_i \cdot h_i(\mathbf{x})$$

L’addestramento: feature selection

Tracciando una analogia tra classicatore debole e feature risulta subito evidente come AdaBoost possa costituire, a tutti gli effetti, una procedura di selezione delle migliori feature da combinare per la costruzione del classicatore di volti finale.

Viola e Jones definiscono il classicatore debole come una singola feature di Haar alla quale viene associata una soglia di decisione. Se il responso dell'applicazione di una feature alla sotto-regione in esame supera la soglia definita, tale la regione è interpretata dal classicatore debole come possibile faccia.

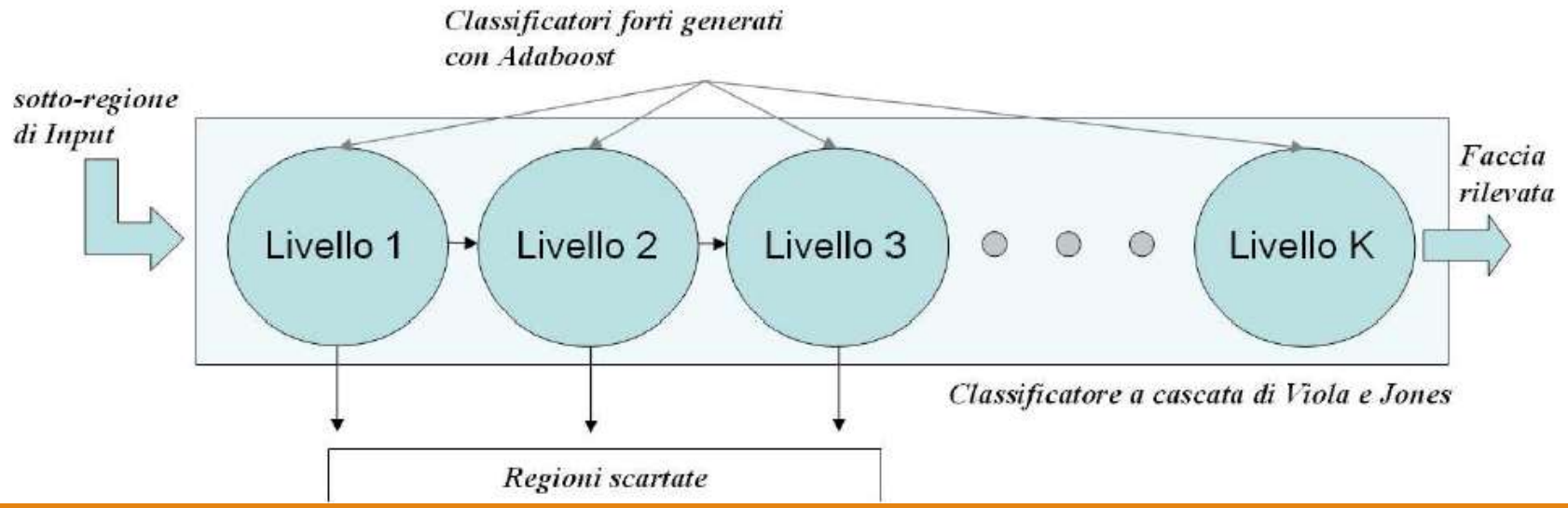
Si noti che una singola feature (più soglia) non permette mai di determinare con una precisione accettabile se una regione contiene una faccia o no, è necessario quindi combinare più feature in maniera sequenziale l'una dopo l'altra.

L'algoritmo di apprendimento, in primo luogo, determina per ogni feature disponibile la soglia ottimale che permette di classificare erroneamente il minor numero di esempi; in questa fase dell'apprendimento si produce una lista di feature ordinate per contributo alla risoluzione del problema. Successivamente la procedura di addestramento riduce il set delle feature utilizzabili selezionando solo quelle che meglio separano gli esempi positivi da quelli negativi. Da questo sotto insieme di feature viene poi generata la combinazione lineare pesata che struttura il classicatore forte

Dall’addestramento al campo

Il classificatore finale di Viola e Jones è una combinazione di classificatori forti organizzati come una cascata di successivi stadi di classificazione. La struttura a cascata nasce dall'esigenza di ottimizzare i tempi di calcolo.

L'idea di base è quella di valutare il minor numero possibile di feature sulle regioni dell'immagine di minor interesse in modo da ridurre il tempo impiegato. Per raggiungere tale obiettivo la struttura a cascata è organizzata diversi livelli (stadi), ognuno dei quali è rappresentato da un classificatore forte, addestrato con AdaBoost.



La scelta delle regioni positive

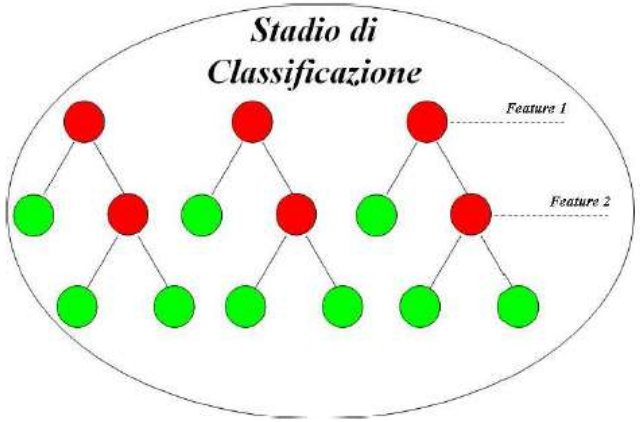
L'input di un livello è composto dalle regioni dell'immagine giudicate positivamente dal livello precedente. Questa segmentazione del problema ha come effetto positivo quello di rendere la complessità della catena di valuazione crescente mano a mano che la cascata diventa più profonda. Segue che i classificatori semplici, risultato della combinazione lineare di un basso numero di classificatori deboli, sono utilizzati nei primi livelli per scartare rapidamente la maggior parte delle regioni dell'immagine non contenenti facce, mentre classificatori complessi hanno il compito di valutare solo le porzioni di immagine molto simili a facce attraverso il calcolo di più feature.

Tutti i contributi di ogni singolo classicatore debole all'interno di uno stadio della cascata vengono sommati tra loro per formare il contributo totale che la combinazione lineare di classicatori lineari restituisce; tale valore viene in fine confrontato rispetto alla soglia di livello per valutare se la regione in esame può essere scartata (non-faccia) oppure deve essere promossa al successivo stadio di valutazione.

Le feature di Haar: estensione di Lienhart

- R. Lienhart nel 2002 estende il set base di feature utilizzato da Viola e Jones introducendo altre tipologie di feature che si dimostrano utili ad aumentare l'efficacia dell'algoritmo in termini di migliore capacità discriminante; tra queste si notino le tilted (inclinate di 45° rispetto alla versione base di partenza) e le center-surround (ottime per localizzare gli occhi).

- Inoltre viene dimostrato in modo empirico che la configurazione più efficiente è quella in cui ogni stadio di classicazione è una combinazione lineare di classificatori deboli di tipo CART (Classication and Regression Tree) con due split (ramicazioni) di decisione. La versione classica utilizza invece classificatori STUMP, una feature con due soglie di uscita.



Performance dell'addestramento

- L'addestramento del classificatore è svolto in modo tale che ogni stadio di classificazione accetti (erroneamente) un fissato rapporto «f» di regioni contenenti non-facce e accetti (correttamente) un fissato rapporto «d» di regioni contenenti facce:

$$p\left(F_i(\mathbf{x}) \geq 0 \mid F_{i-1}(\mathbf{x}) \geq 0, y = 1\right) = d$$

$$p\left(F_i(\mathbf{x}) \geq 0 \mid F_{i-1}(\mathbf{x}) \geq 0, y = -1\right) = f$$

Performance dell'addestramento

Sia F+ l'evento di classificazione con successo all'i-esimo stadio. Il tasso di rilevamento atteso del classicatore finale (detection rate) si ricava applicando ricorsivamente la regola del prodotto di probabilità (sotto certe condizioni che qui tralasciamo). A parità di capacità di classificazione, la versione organizzata a cascata presenta un'efficienza 10 volte superiore rispetto alla versione monolitica dello stesso (ovvero organizzato come un solo grande livello di classificazione addestrato con AdaBoost).

$$
\begin{aligned}
d_g = p\left(F_k^+, \ldots, F_1^+ \mid y=1\right) &= p\left(F_k^+ \mid F_{k-1}^+ \ldots, F_1^+, y=1\right) \cdot p\left(F_{k-1}^+, \ldots, F_1^+ \mid y=1\right) \\
&= \underbrace{p\left(F_k^+ \mid F_{k-1}^+, y=1\right)}_{d} \cdot p\left(F_{k-1}^+, \ldots, F_1^+ \mid y=1\right) \\
&\vdots \\
&= \prod_{i=1}^{k} p\left(F_i(\mathbf{x}) \geq 0 \mid F_{i-1}(\mathbf{x}) \geq 0, y=1\right) = d^k \qquad (2.5)
\end{aligned}
$$

$$
f_g = p\left(F_k^+, \ldots, F_1^+ \mid y=-1\right) = f^k
$$

Il problema dell'addestramento

Quando si affronta un problema di classificazione che richiede una fase di apprendimento sono sostanzialmente tre gli elementi in gioco:

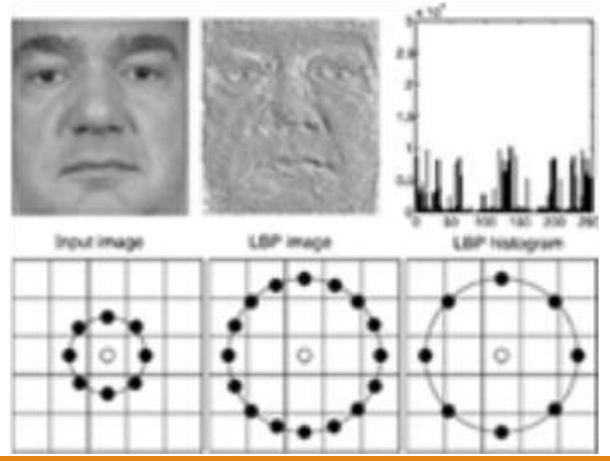
1. Dataset (Train Set vs. Test Set)
2. Feature (Haar, LBP, HOG, SIFT, ecc…)
3. Algoritmo di apprendimento (AdaBoost, SVM, ecc…)

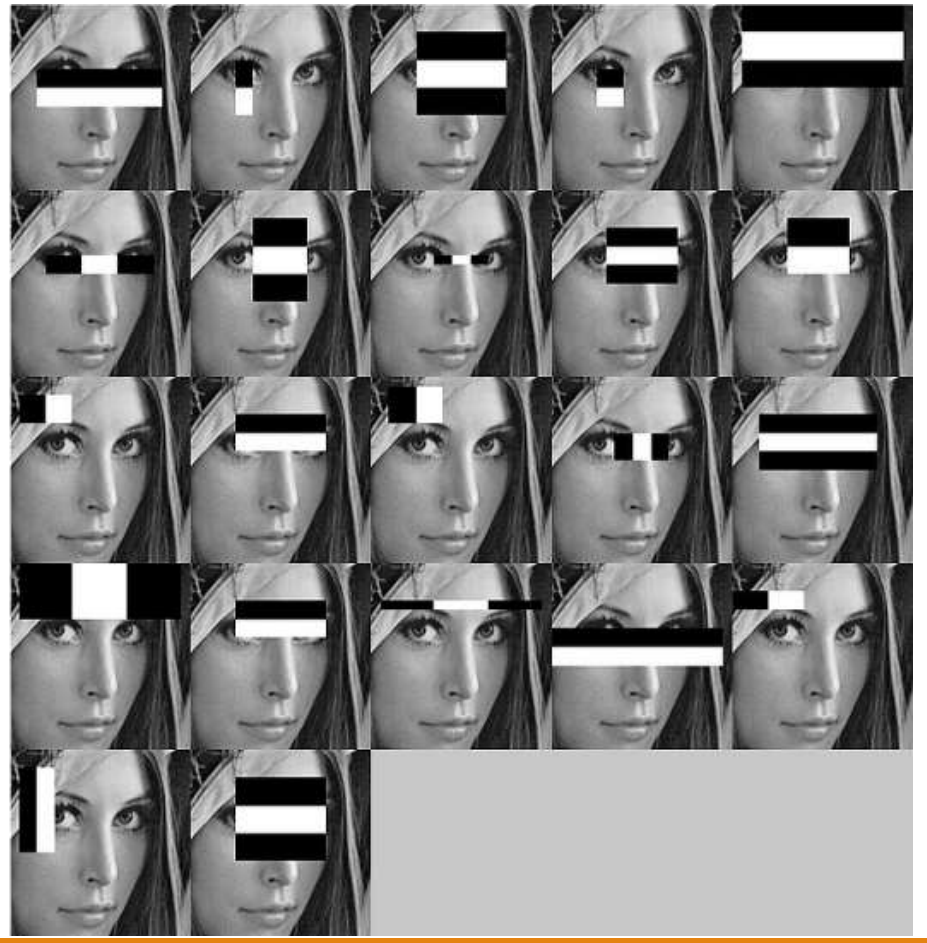
Questi tre elementi ci sono sempre, ma quali dataset, feature e algoritmi di apprendimento usare dipende sempre dal problema! Si noti che molto spesso problemi diversi si risolvono con tecniche simili e problemi molto simili si risolti brillantemente anche con tecniche molto diverse. Non sottovaluatare mai la produzione del Dataset, quasi sempre è una delle maggiori criticità. Ad oggi esistono decine di dataset diversi per le facce, pedoni, automobili ecc.. Non dovrebbe sorprendere che questi sono i problemi che hanno raggiunto performance superiori.

L’importanza del dataset

L’importanza del dataset

L’importanza delle Feature





Art vs Feature: CVDazzle by Adam Harvey

@adamhrv

www.cvdazzle.com

Approfondimenti

Viola & Jones https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

Lienhart http://www.lienhart.de/Prof._Dr._Rainer_Lienhart/Source_Code_files/ICIP2002.pdf

CONDENSATION http://link.springer.com/article/10.1023%2FA%3A1008078328650

Social Q&A

@vs_AR
#askVisionary
www.vision-ary.net