


# Immagini ed apprendimento statistico, ovvero il computer puo' imparare a vedere?




**Francesca Odone**

*Istituto Nazionale di Fisica della Materia,  
unita' di Genova*

il lavoro presentato e' svolto in collaborazione con Annalisa Barla,  
Emanuele Franceschi, Fabrizio Smeraldi, Alessandro Verri

# Breve presentazione



- Da alcuni anni sono un membro dello  che opera presso il DISI dell'Univ. di Genova

- ci occupiamo di:
  1. studiare metodi matematicamente fondati per risolvere problemi di classificazione
  2. sviluppare tecniche per estrarre informazione visiva da immagini digitali e sequenze di immagini

# Contesto

---

- *l' image understanding*, uno dei problemi centrali dell'intelligenza artificiale, è ben lontano dall'essere risolto
- La capacità innata nell'uomo di interpretare *immediatamente* il contenuto del segnale visivo non è stata trasmessa ai calcolatori i quali, talvolta, falliscono anche nello svolgere compiti molto semplici

# Contesto

---

- Anni di ricerca nel mondo della visione artificiale, dell'immagine processing e discipline affini non sono stati sufficienti ad aggredire il problema in modo significativo
- L'apprendimento statistico introdotto nell'ultimo decennio sembra fornire interessanti spunti di ricerca e mette a disposizione un nuovo modo di trattare problemi difficilmente modellabili

# Sommario

---

- motivazioni
- formalizzazione del problema
- elementi di visione artificiale
- apprendimento statistico
- costruire classificatori per immagini
- applicazioni
- stato dei lavori

# Obiettivi

---

- data una **qualunque** immagine digitale descriverne il contenuto

# Obiettivi

---

- Obiettivo “semplificato”:
  - data una **qualunque** immagine digitale rispondere ad un numero finito di domande che potrebbero aiutare a descriverne il contenuto:
    - contiene **oggetti** di una certa classe nota?
    - contiene **roba** di un certo tipo?
    - e' caratterizzata da **proprietà'** di un certo tipo?

# Obiettivi

---

- cosa e' cambiato?
  - data una **qualunque** immagine digitale  
descrivere il contenuto
  - data una **qualunque** immagine digitale  
rispondere ad un numero finito di domande  
che potrebbero aiutare a descriverne il  
contenuto...
- il problema e' stato ridimensionato:  
descriviamo alcuni possibili aspetti di  
un'immagine



# Obiettivi

---

- Esempi di “domande”:
  - **oggetti**: volti umani, pedoni, auto, animali, ...
  - **roba** (oggetti diffusi): cielo, mare, neve, prati,...
  - **proprietà' globali**: immagine di interni o esterni, diurna o notturna, città' o campagna, panorama o primopiano, ...

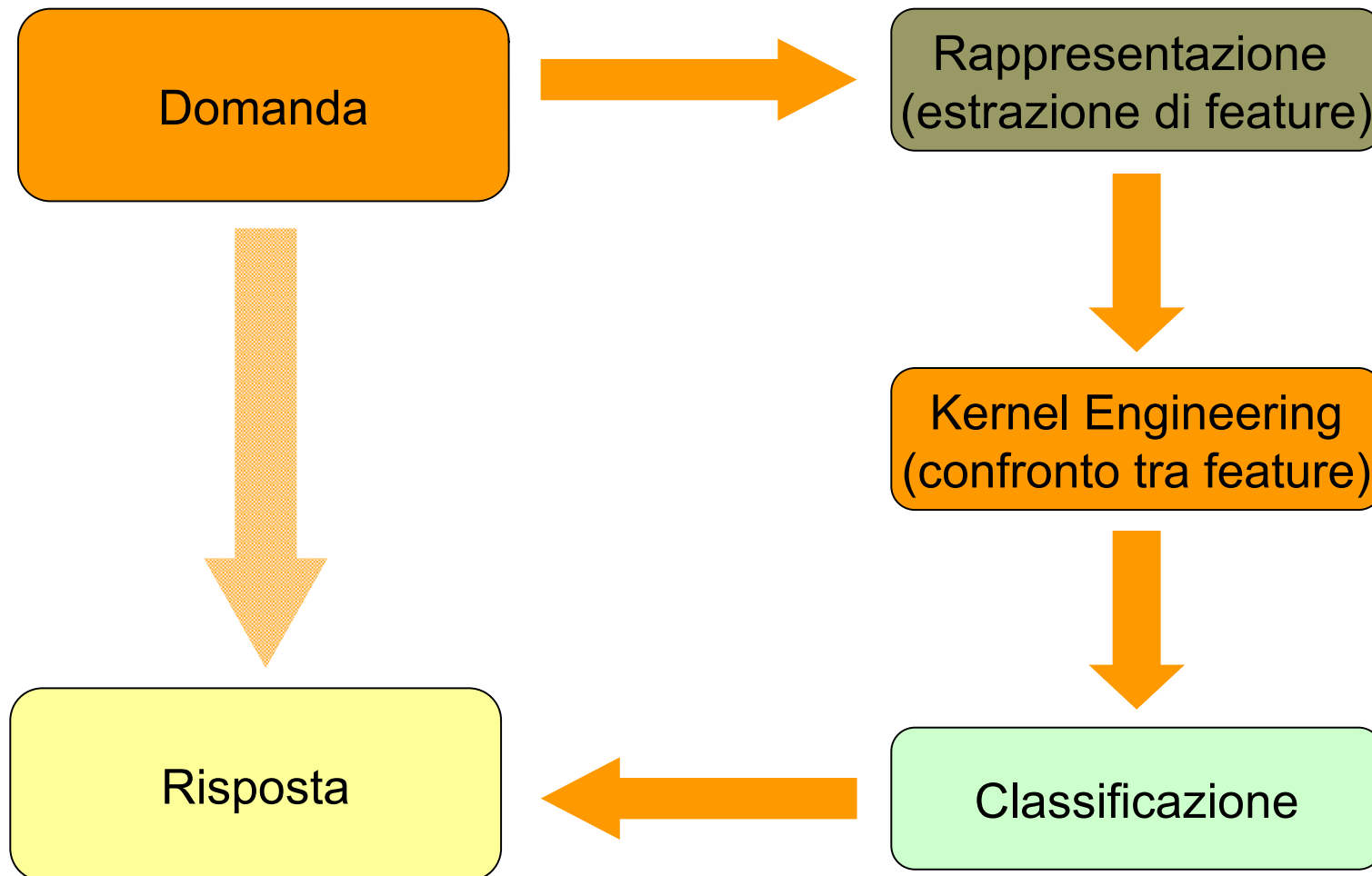
# Formalizzazione del problema

---

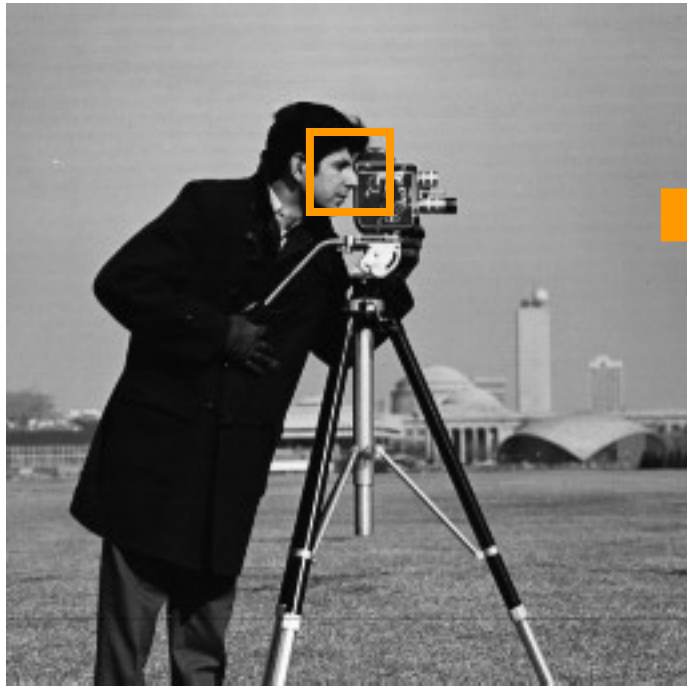
- Per rispondere ad una domanda del tipo visto sopra occorre:
  - rappresentare l'immagine in modo opportuno
  - utilizzare un algoritmo di classificazione che ci aiuti a classificare immagini *mai viste prima*
- Per rappresentare le immagini usiamo algoritmi di visione "low level"
- Per la classificazione algoritmi di apprendimento da esempi

# Formalizzazione del problema

---



# Le immagini digitali



12	12	10	10	9	
10	9	9	9	9	
10	11	10	9	9	
10	10	10	10	10	1
10	10	10	11	16	2
10	10	11	20	43	10
9	10	37	117	166	18
10	43	165	186	185	18
35	159	183	178	174	15
79	176	186	174	150	10
89	177	186	179	175	13
98	171	181	185	189	18
114	155	177	188	192	19
142	144	167	173	178	17

# Estrazione di feature *low level*

---

- **Scopo:**
  - mettere in evidenza dettagli o proprietà di un'immagine
- **Metodi utili e general purpose:**
  - statistiche del primo e del secondo ordine
  - filtri e banchi di filtri
  - rilevatori di feature: contorni, punti angolosi, linee, ...

# Un esempio di feature utile: gli *edge*

---

- Gli edge (contorni) sono pixel posizionati in zone di intensa variazione di luminosita'
- Per rilevare i punti di edge solitamente si studia il gradiente dell'intensita' luminosa:

$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$



# gli istogrammi

---

- Dato un segnale definito su un dominio  $D$  suddiviso intervalli limitati, un istogramma rappresenta la frequenza con cui il segnale assume valori in ognuno degli intervalli.
- Un istogramma di un immagine a 256 livelli di grigio:
  - il dominio e' l'insieme di valori  $[0,255]$
  - il dominio e' solitamente diviso in intervalli detti *bin*
  - l'istogramma e' calcolato contando quanti pixel dell'immagine cadono in ogni bin

# gli *istogrammi*

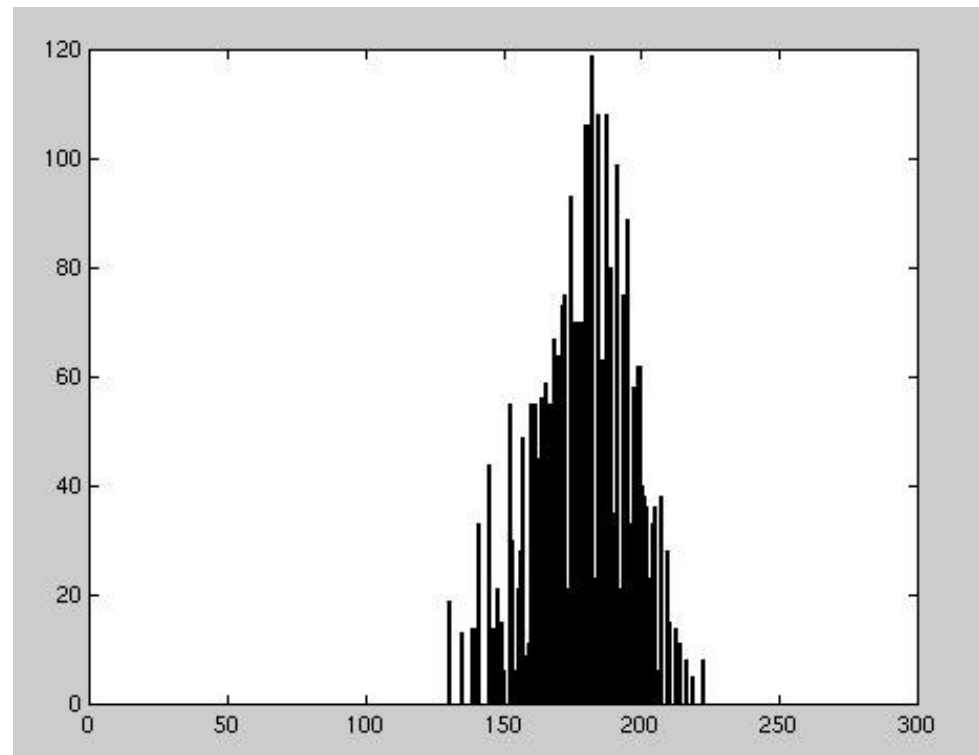
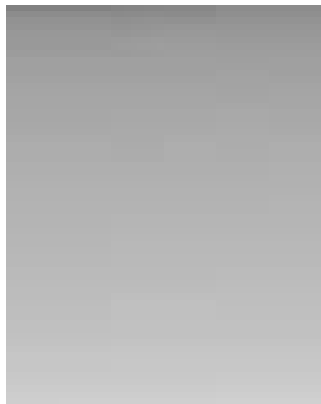
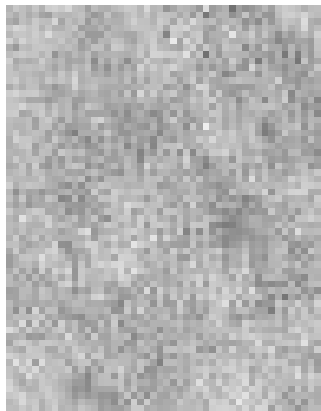
---

- gli istogrammi di immagini sono utili:
  - quando le proprietà di correlazione tra pixel contigui in un'immagine possono essere ignorate
  - quando siamo interessati alla distribuzione e non alla geometria di una feature
  
- si possono calcolare istogrammi sull'immagine originale (sui grigi/luminosità, sui colori, ...) oppure su feature (sui moduli o le direzioni dei gradienti o degli edge, ...)



# gli istogrammi

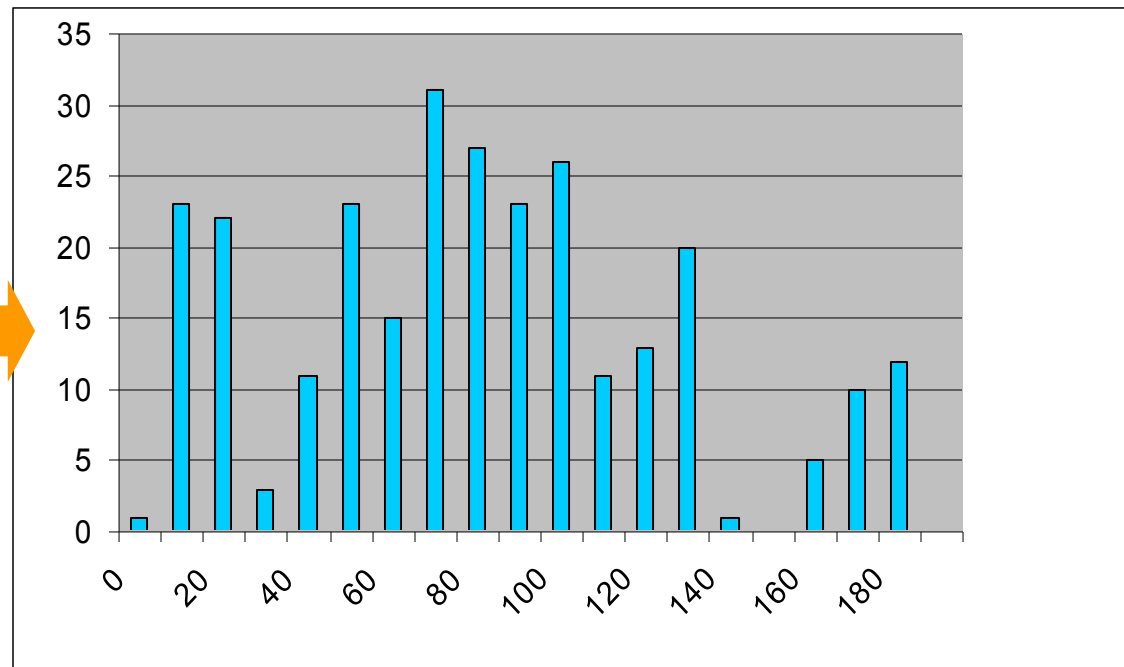
---



# proprietà' di orientazione utilizzando gli edge

## □ Il metodo

- Rilevare i punti di edge in un'immagine
- calcolare un istogramma della mappa delle direzioni degli edge  $E_o(i,j)=\arctan(J_y/J_x)$



# Classificazione a due classi

---

- Oggetti:  $x \in X, \quad X \subseteq \mathbb{R}^n$
- Classi:  $y \in Y, \quad Y = \{-1, 1\}$
- Dati:  $D_l \equiv \{(x_i, y_i) \in X \times Y\}, \quad i=1, \dots, l$

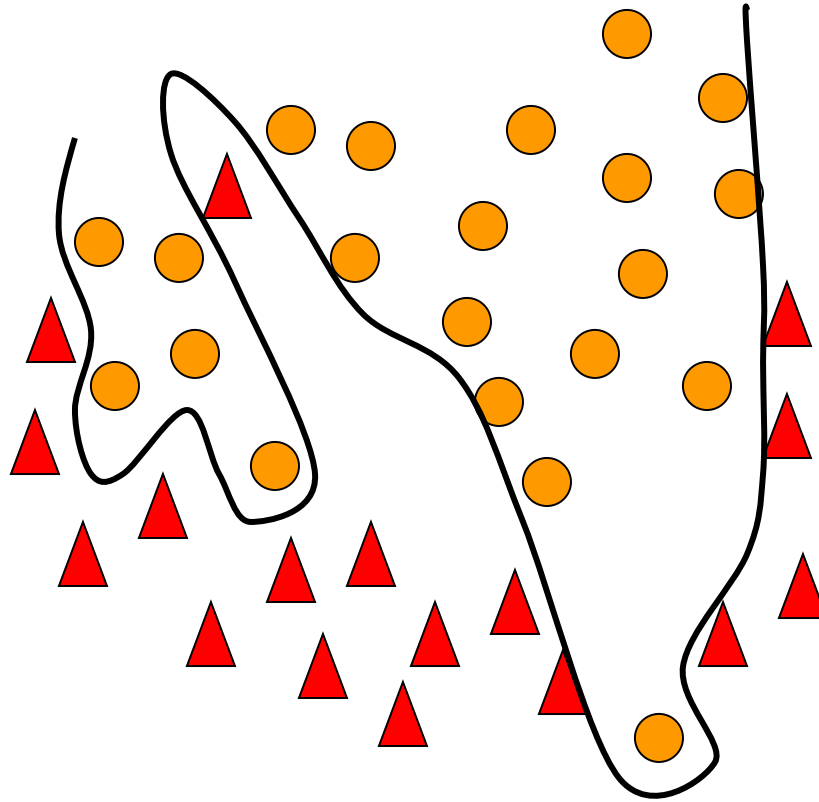
Estratti da  $X \times Y$  secondo  $P(X, Y)$  ignota.

**Obiettivo:**  $f_0 : X \rightarrow Y$

che consenta la separazione delle due classi

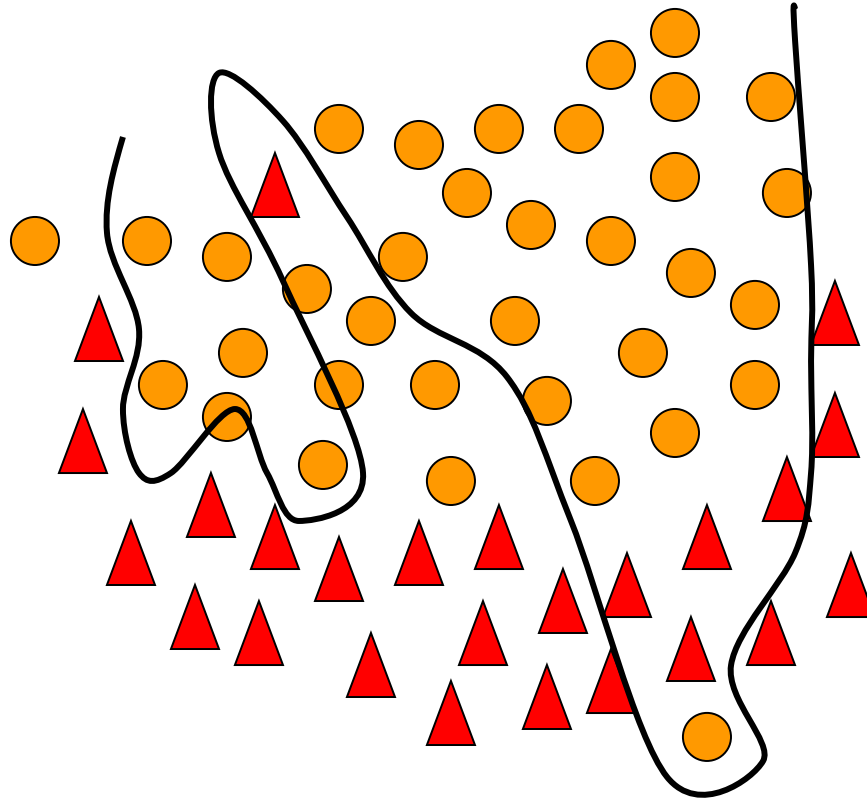
# Classificazione a due classi

---



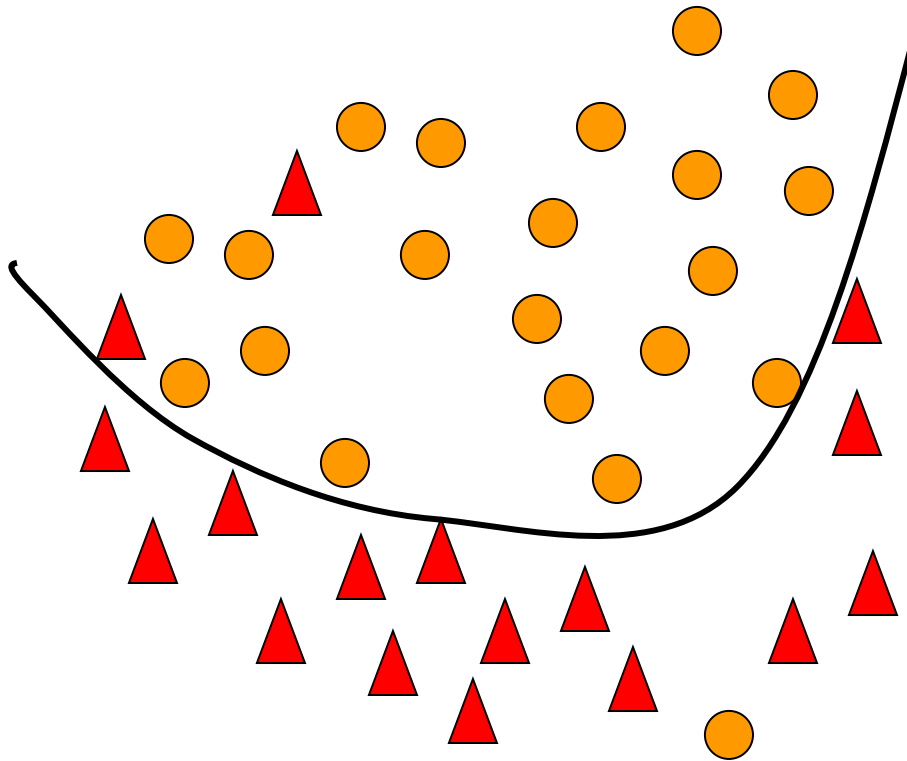
# Classificazione a due classi

---



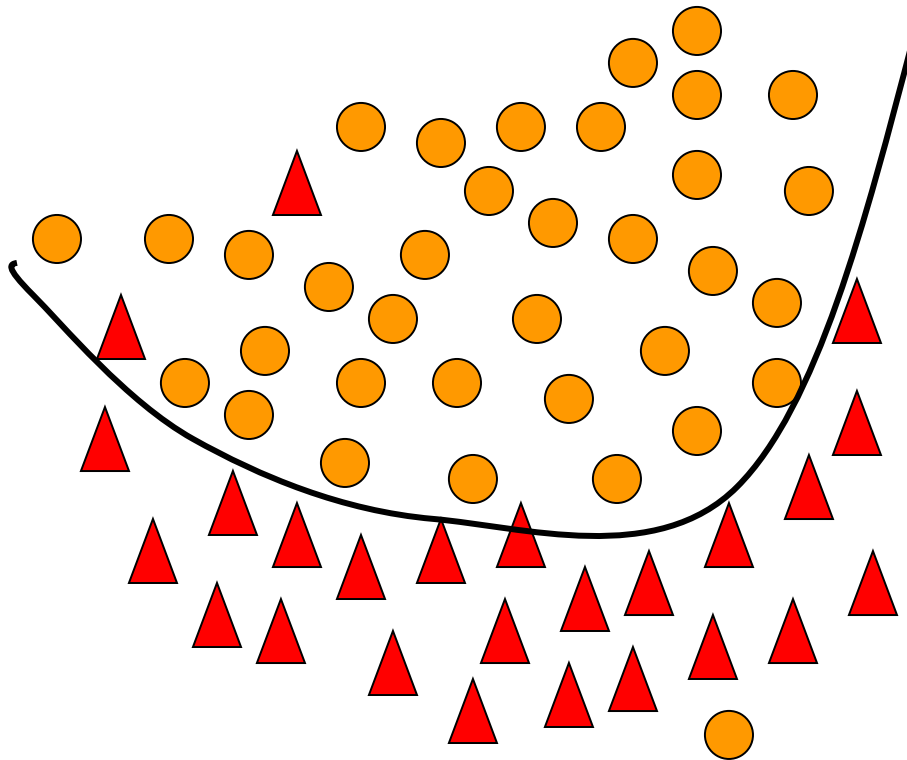
# Classificazione a due classi

---



# Classificazione a due classi

---



# Support Vector Machines (SVM)

## (Vapnik, 1995)

---

Dato un insieme di punti  $\{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$   
i.i.d. estratti da una distribuzione di  
probabilita' fissata ma ignota, il problema  
consiste nel trovare una funzione  $f$  che  
minimizzi il funzionale

$$\min_f \frac{1}{\ell} \sum_{i=1}^{\ell} V(f(x_i), y_i) + \lambda \|f\|_K^2$$



# Support Vector Machines (Vapnik, 1995)

---

La soluzione (il miglior compromesso tra l'errore empirico e il termine di regolarizzazione) e' del tipo

$$\hat{f}(x) = \sum_{i=1}^{\ell} \alpha_i K(x_i, x)$$

# Caratterizzazione dei kernel

---

- Una funzione  $K$ , per essere un kernel, deve essere:
  - Simmetrica
  - Definita positiva

# Caratterizzazione dei kernel

---

- Una funzione  $K: X \times X \rightarrow R$  è **definita positiva** se per ogni intero  $n$ , per ogni  $x_1, \dots, x_n \in X$ , e per ogni  $\alpha_1, \dots, \alpha_n \in R$ ,

$$\sum \sum \alpha_i \alpha_j K(x_i, x_j) \geq 0.$$

- Questo assicura che  $K$  si comporti come un prodotto scalare in un qualche spazio delle feature opportunamente definito:

$$\varphi(x_i) \cdot \varphi(x_j) = K(x_i, x_j)$$

# Scelta dei kernel e conoscenza a priori

---

- La conoscenza a priori del problema di classificazione considerato puo' (e dovrebbe) essere utilizzata per progettare funzioni kernel *ad hoc*:
  - *questo permette di apprendere da un numero minore di esempi*
  - *e soprattutto di ridurre il preprocessing dei dati*

# Kernel per immagini

---

- Intuitivamente una funzione kernel e' una funzione che valuta la similarita' tra i dati
  - misure di correlazione per immagini
  - misure di similarita' per istogrammi

# Applicazioni

---

## □ Domanda:

“questa e' un'immagine di interno o di esterno?”

## □ Formalizzazione:

- abbiamo raccolto un training set di circa 2500 immagini di interni e 2500 di esterni
- le immagini sono state rappresentate come istogrammi nello spazio di colore HSV
- abbiamo allenato una SVM a due classi sui dati così rappresentati e un kernel adatto ai dati considerati (basato sulla misura di similarità *intersezione di istogrammi*), ottenendo i pesi  $\alpha_i$  per la soluzione
- una nuova immagine, rappresentata anch'essa come un istogramma di colore, viene classificata utilizzando la funzione

$$\hat{f} = \sum_{i=1}^{\ell} \alpha_i K(x_i, x)$$

# esempi di interni/esterni

---



# Risultati

---

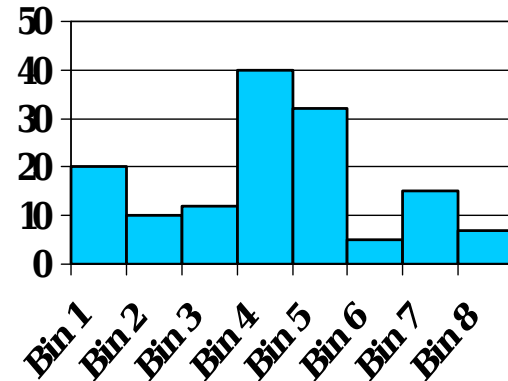
- Su un validation set di 400 elementi otteniamo percentuali di riconoscimento intorno al 90%.
  
- Assunzioni implicite:
  - l'immagine e' una foto
  - l'immagine non e' ritoccata
  - l'immagine non e' un primo piano di un oggetto
  - meglio se l'immagine e' di qualita'



# Non dimentichiamo che...

---

questo e' quello che il PC  
usa per rispondere alla  
domanda  
relativamente a  
questa immagine



# Applicazioni

---

## □ Domanda:

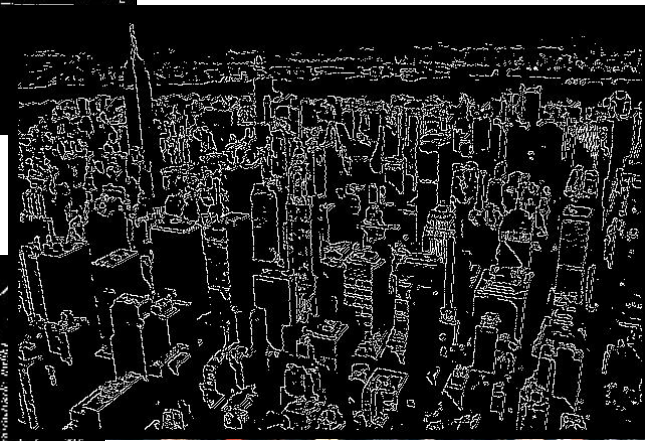
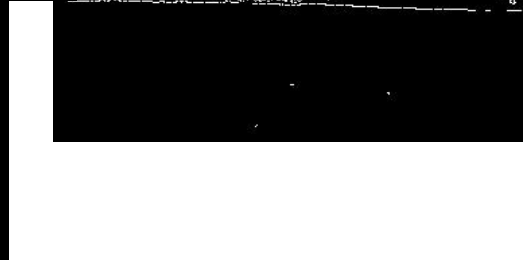
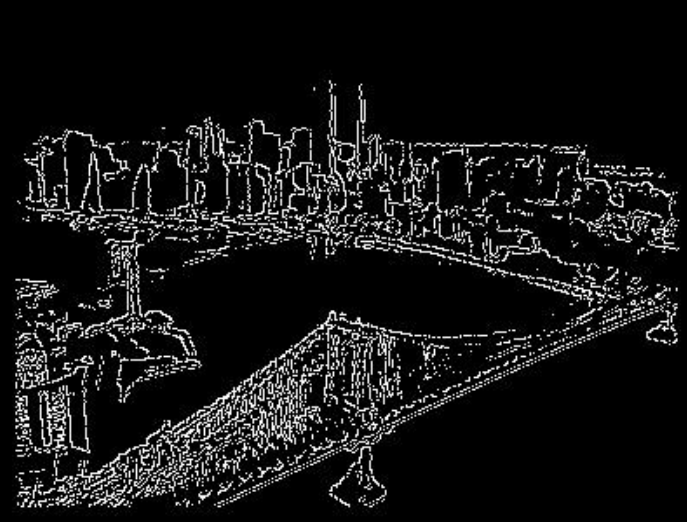
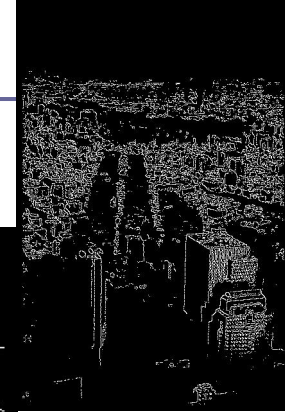
“questa e' l'immagine di un cityscape?”

## □ Formalizzazione:

- abbiamo raccolto un training set di circa 800 immagini di citta' e 800 di altro tipo
- le immagini sono state rappresentate come concatenazione di istogrammi delle direzioni degli edge e istogrammi nello spazio di colore HSV
- abbiamo allenato una SVM a due classi sui dati cosi' rappresentati ottenendo i pesi  $\alpha_i$  per la soluzione
- una nuova immagine, rappresentata anch'essa come un istogramma di colore, viene classificata utilizzando la funzione

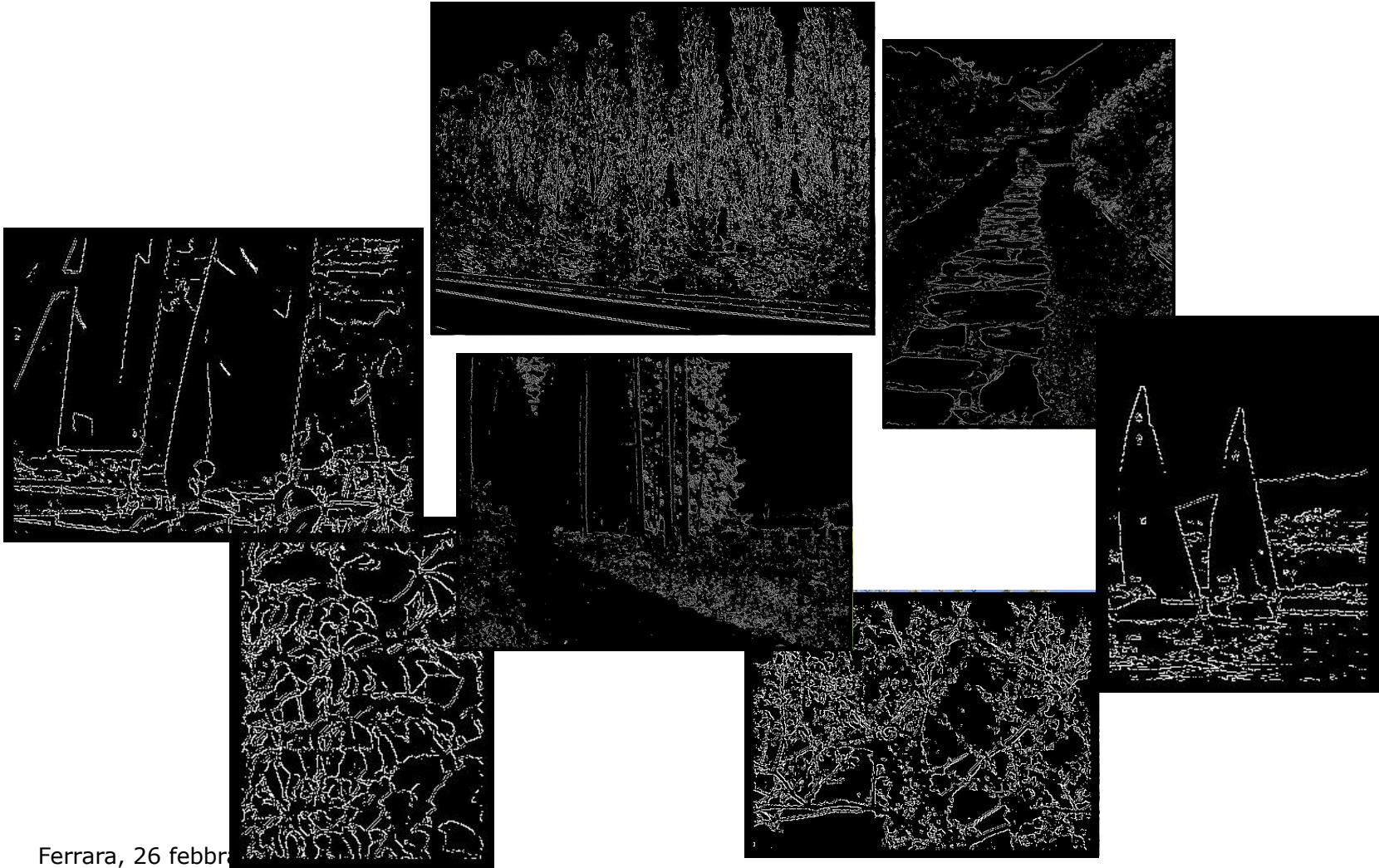
$$\hat{f} = \sum_{i=1}^{\ell} \alpha_i K(x_i, x)$$

# esempi di cityscape



# esempi negativi (non cityscape)

---



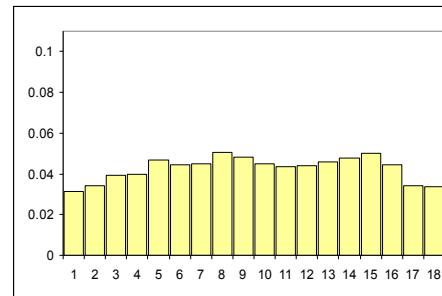
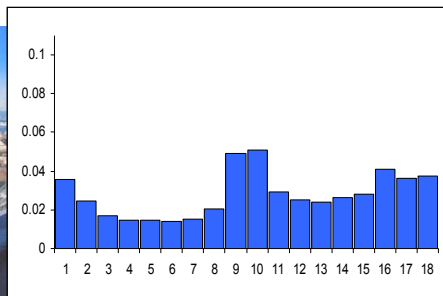
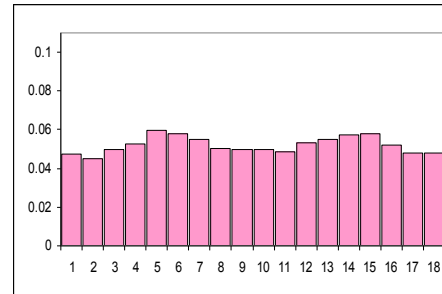
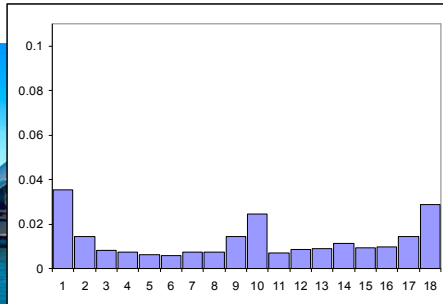
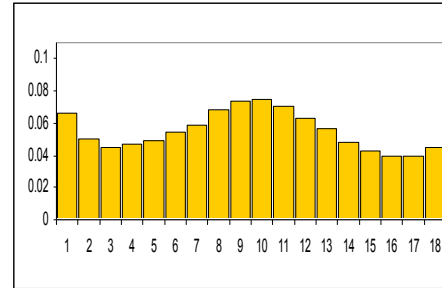
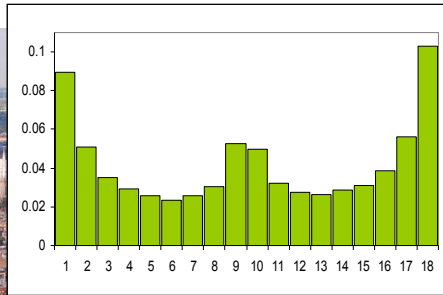
# Risultati

---

- I. Precision **90%** (percentuale di cityscape veri tra quelli rilevati dal sistema)
- II. Recall **85%** (percentuale di cityscape rilevati sul totale di cityscape esistenti nel validation set)

non male considerando la rappresentazione..

# ...perche' funziona...?



# I limiti di questo approccio

---

- Fino ad oggi abbiamo portato a termine una decina di classificatori (solo alcuni in modo soddisfacente)
- A questo ritmo diventeremo vecchi prima di riuscire a descrivere immagini anche in modo molto elementare
- I due punti deboli di questa tecnica sono:
  - scegliere la rappresentazione dei dati per un determinato problema
  - raccogliere i training set

# Scelta della rappresentazione

---

- Se abbiamo a disposizione un'ampia batteria di rappresentazioni possibili, il problema di scegliere quale rappresentazione usare per un certo problema si traduce in fare **selezione di feature**



# classificazione basata su test d'ipotesi

---

- assumiamo di avere un buon numero di esempi *positivi*
- come procediamo:
  - calcoliamo un grande numero di feature (decine di migliaia)
  - selezioniamo un sottoinsieme di feature *buone* (migliaia)
  - tra queste ultime identifichiamo un sottoinsieme di feature *indipendenti* (centinaia)

# classificazione basata su test d'ipotesi

---

- A questo punto abbiamo una descrizione calzata sui dati di input e quindi specializzata al problema in esame a partire da una rappresentazione ridondante e multipurpose
- I dati rappresentati in questo modo possono diventare i vettori di input di un classificatore tipo SVM
- Abbiamo anche provato ad usarli all'interno di un classificatore *monoclasse* basato su test statistici multipli

# classificazione basata su test d'ipotesi: un esempio

---

## FACE DETECTOR



### Faces

(19x19pixels)

Training: 2429

Test: 472

### Nonfaces

(19x19pixels)

Training: ~~4548~~

Test: 23573

# training con test d'ipotesi

---

- I. Calcoliamo circa 16000 feature *low level* (medie dei grigi, tomografie in varie direzioni, ranklet) sulle immagini positive del training set
  
- II. Estraiamo un sottoinsieme di feature *buone* e da queste un sottoinsieme di feature *indipendenti*
  
- III. costruiamo un test statistico per ognuna delle feature sopravvissute utilizzando il training set

# rilevamento di facce

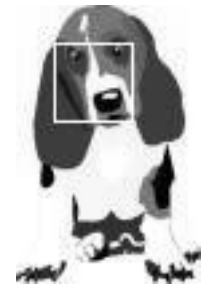
---

Per tutte le posizioni dell'immagine, un insieme di scale e una soglia ***t***

- I. calcoliamo le feature *sopravissute*
  
- II. effettuiamo tutti i test statistici ad un certo livello di confidenza
  
- III. si rileva un volto in una determinata posizione e scala se almeno ***t*** test sono passati

# Alcune facce...

---



... altre facce ...



Ferrara, 26 febbraio 2004

# ... ancora facce ...

---





# Stato dei lavori

---

- Feature selection basata su test d'ipotesi per catturare rappresentazioni *ad hoc* :
  - rilevamento di oggetti dominanti, es. cielo, mare, prati, folla, ...
- DEMO ONLINE di alcuni nostri classificatori!! <http://slipguru.disi.unige.it>

---

**FINE !**

# Un esempio di feature utile: *gli edge*

---

- L'algoritmo (Canny edge detector)
  - filtrare l'immagine per attenuare il rumore di acquisizione
  - calcolare il gradiente e stimarne la direzione
  - eliminare i punti di non massimo
  - applicare una soglia per trattenere solo i punti a contrasto alto

# Quando gli istogrammi non bastano

---

un esempio: le matrici di co-occorrenza

rappresentano la distribuzione dei livelli di grigio di coppie di pixel a posizioni relative fissate

