



SAPIENZA
UNIVERSITÀ DI ROMA

Student face detection, flow prediction and class-room optimal allocation

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corso di Laurea in Ingegneria Informatica

Candidato

Francesco Zappia

Matricola 1757366

Relatore

Roberto Capobianco

Correlatore

Lavinia Amorosi

Anno Accademico 2019

Student face detection, flow prediction and classroom optimal allocation

Tesi di Laurea. Sapienza – Università di Roma

© 2019 Francesco Zappia. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: zffromGerace@gmail.com

Sommario

Questa tesi è incentrata sullo sviluppo di un'applicazione da utilizzare in ambito universitario atta ad automatizzare il processo di raccolta dati riguardanti il flusso di studenti in particolari circostanze (temporali ed accademiche) utilizzati a loro volta per stimare il numero di allievi presenti in nuove contesti; i risultati vengono in seguito impiegati per assegnare le diverse classi a set di aule rispettivamente con capienze adatte a soddisfare la loro richiesta ottimizzando la loro allocazione in base a specifiche ed ulteriori richieste.

Contents

1	Introduzione	1
1.1	Il problema	1
1.2	Struttura del documento	1
2	State of the art	5
2.1	Face Detection	5
2.1.1	Problematiche comuni	5
2.1.2	Approcci feature-based	5
2.1.3	Algoritmo Viola-Jones per la Face Detection	7

Chapter 1

Introduzione

1.1 Il problema

In ambito universitario è problematica molto comune quella di cercare di assegnare aule a gruppi di studenti in modo tale da rispettare le richieste in numero di posti di ciascuno di essi senza nemmeno concretamente conoscere il numero di allievi che frequenteranno realmente il corso in questione, dovendo l'istituto assegnarle in anticipo rispetto all'inizio delle lezioni.

Si pone quindi l'ulteriore problema di facilitare la raccolta di dati che permetta di effettuare delle stime più accurate, quindi di registrare in modo automatico la partecipazione degli universitari a ciascuna lezione e quindi inferire attraverso di essi queste stime.

Quest'applicazione si pone l'obiettivo di risolvere esattamente questa problematica (il funzionamento è illustrato in 1.1): a partire da foto (che si intende essere di aule) riconosce il numero di volti e quindi di studenti in essa presenti. Questa informazione, associata ad informazioni riguardanti la lezione in questione, viene fornita insieme a quest'ultime ad uno stimatore che, in base ai dati ricevuti cerca di costruire un modello matematico capace di stimare il flusso di allievi in nuove e richieste situazioni: delle stime che verranno impiegate per distribuire ciascun gruppo nelle diverse aule con data capacità in modo ottimale, cercando cioè di massimizzare il valore di una certa funzione matematica che rappresenta il problema di allocazione.

1.2 Struttura del documento

Il documento è strutturato in

- **State of the Art**, in cui vengono accennati i concetti teorici utilizzati per la risoluzione del problema; vengono analizzati articoli e fonti che hanno affrontato le stesse problematiche.
- **Metodo**, nel quale sono approfondite le metodologie usate per la realizzazione dell'applicazione e vengono trattate ulteriori questioni sorte nell'implementazione.
- **Risultati**, che illustra gli esiti e l'efficienza della realizzazione scelta.

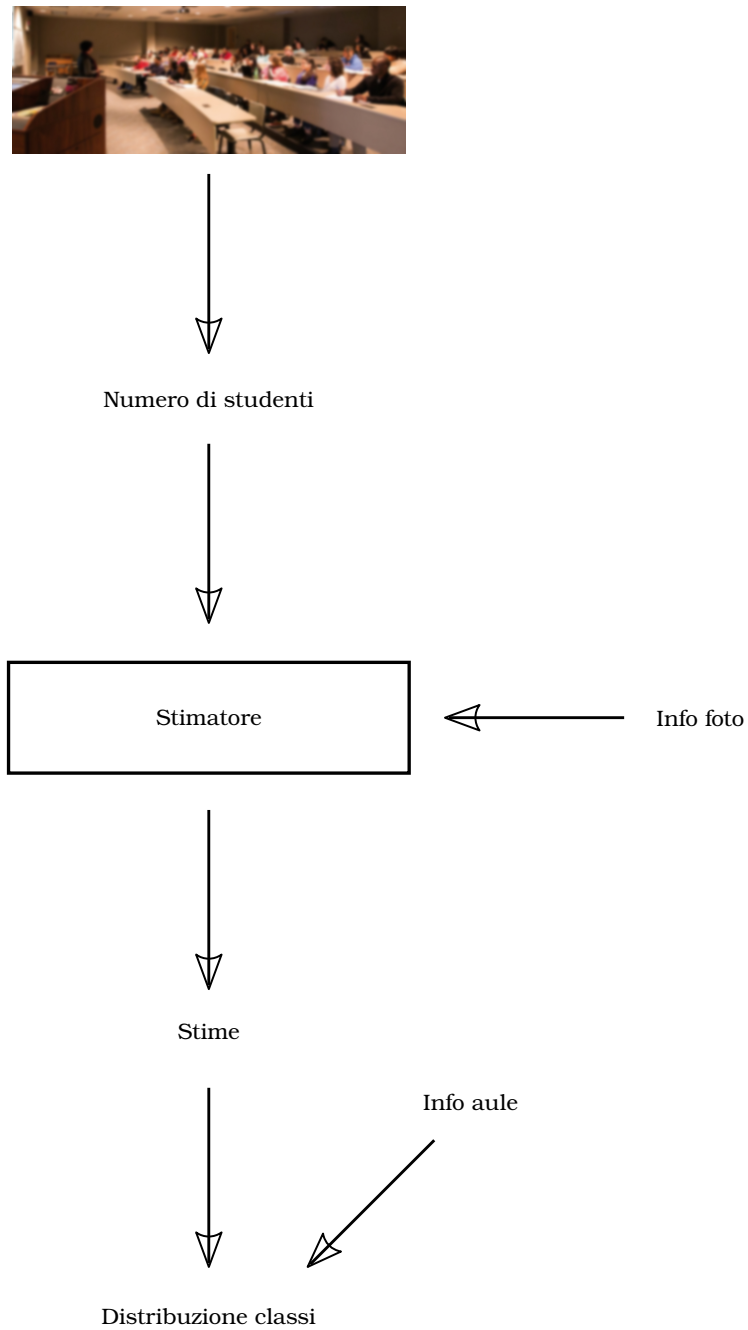


Figure 1.1. Il flusso di funzionamento dell'applicazione

- **Conclusioni**, contenente le osservazioni finali dedotte dai risultati ottenuti.

Chapter 2

State of the art

In questo capitolo saranno inizialmente esposti i concetti teorici alla base dell'applicazione sviluppata: dagli algoritmi più comuni per la **Face Detection** ad un'introduzione al **Machine Learning** per finire poi con elementi di **Programmazione Lineare Multiobiettivo**.

Verranno quindi analizzati precedenti progetti che hanno affrontato le stesse problematiche.

2.1 Face Detection

La Face Detection è una tecnologia che permette di localizzare ed estrarre da un'immagine la regione contenente un volto [DDB15]; ha oggi diverse applicazioni ed utilizzi, dal video coding al content-based image retrieval.

2.1.1 Problematiche comuni

Le sfide che una tecnica di face detection si trova ad affrontare sono solitamente ([DDB15]):

- **Occlusione.** Spesso parte dei volti sono nascosti da oggetti e/o altri volti
- **Espressioni.** L'aspetto dei volti è fortemente influenzato dall'espressione della persona.
- **Posa.** La posizione da cui è scattata l'immagine influenza la prospettiva del volto (2.2).
- **Luminosità.** Livelli di luminosità eccessivi o troppo bassi impediscono di riconoscere contorni e linee del viso.

2.1.2 Approcci feature-based

In un periodo di ricerca che coinvolge circa gli ultimi trenta anni sono state usati diversi approcci che permettessero di sviluppare tecniche di face detection (2.1), dividendosi principalmente in due categorie: **Feature-based** e **Image-based**.

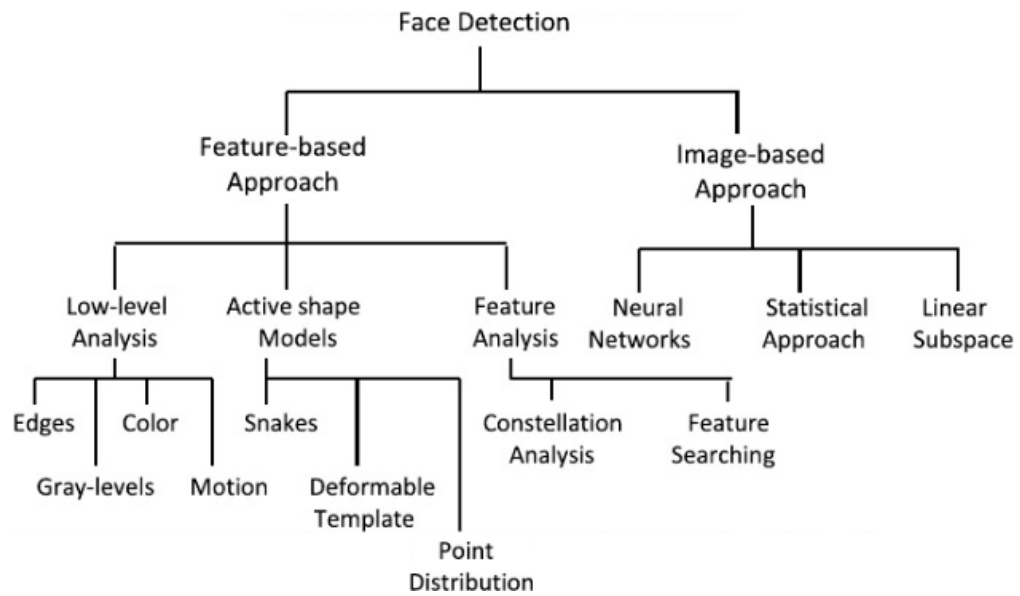


Figure 2.1. Diversi approcci alla face detection



Figure 2.2. Un classico esempio di problema di face detection

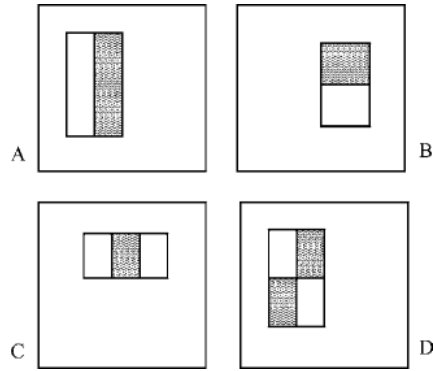


Figure 2.3. Al valore dei pixel nella regione bianca viene sottratto il valore nella regione scura

Nei primi vengono solitamente estratte le feature di un volto da un'immagine (come ad esempio occhi, labbra, sopracciglia) per poi verificare, attraverso la relazione che persiste tra di loro, la presenza di un volto.

All'interno di questa categoria possono essere individuate ulteriori distinzioni ([DDB15], 2.2):

- **Low Level Analysis.** Feature vengono estratte basandosi sulle proprietà dei pixel come ad esempio colore o valore nella scala di grigi.
- **Feature Analysis.** Vengono sfruttate le proprietà geometriche del volto per cercare di localizzare ed individuare le diverse parti del viso.
- **Active shape models.** Questi modelli, che vanno da *snakes* ai PDM (point-distributed models) sono usati per l'estrazione di feature complesse e per tracciare le iridi degli occhi e le labbra.

2.1.3 Algoritmo Viola-Jones per la Face Detection

Invece tra gli svariati approcci **Image-based** uno che ha avuto particolare successo, soprattutto per la velocità di calcolo che permette il suo utilizzo in applicazioni che utilizzano grandi moli e flussi di immagini, è l'algoritmo Viola-Jones.

Esso fa uso di Haar functions per il l'individuazione delle features, i.e. si basa sul calcolo della somma e differenza di pixel in particolari rettangoli [VJ04], come mostrato in 2.3.

Il calcolo viene enormemente velocizzato attraverso le *integral images*, immagini in cui il valore dei pixel è pari alla somma dei pixel in alto a sinistra, i.e.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

con $ii(x, y)$ il valore del pixel nella *integral image* e $i(x', y')$ il valore nell'immagine originale.

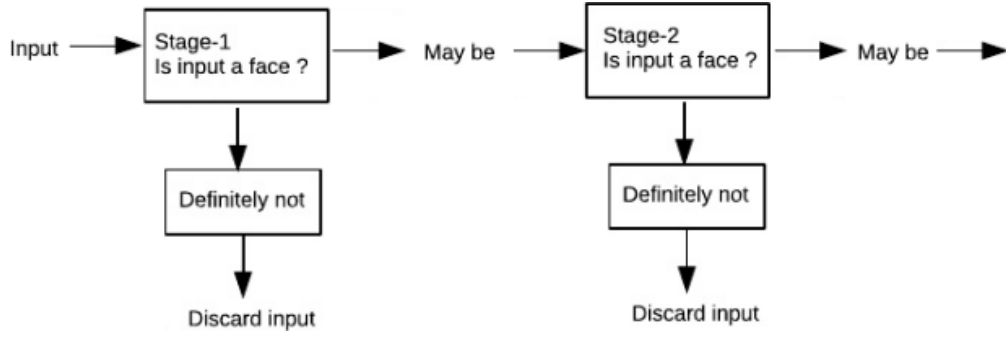


Figure 2.4. Struttura dei classificatori a cascata

Tali *Haar functions* vengono usate quindi da **classificatori**, delle funzioni che mappano un'osservazione (in questo caso un insieme di pixel) su un set finito di valori, in questo caso [Wan14] $f : \mathbb{R}^d \rightarrow \{-1, 1\}$.

Attraverso un Ada Boosting questi singoli ed imprecisi classificatori vengono combinati tra di loro per ottenere un classificatore "forte" che risulta essere molto più accurato [Sch13] dei singoli.

Dato un esempio da classificare x , l'esito della classificazione "forte" è pari a

$$F(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (2.2)$$

con $h_t(x)$ classificatore "debole" a cui corrisponde un peso "di voto" pari a α_t : il risultato è quindi ottenuto come la maggioranza pesata delle classificazioni deboli.

Il rendimento può essere ulteriormente migliorato con l'utilizzo di classificatori a cascata (2.4). Un classificatore a cascata è costituito da diversi *stages*, ognuno contenente uno classificatore "forte", ciascuno dei quali determina se la finestra in input non contiene sicuramente un volto oppure potrebbe. Quando viene predetta l'assenza certa di un viso l'immagine viene scartata, altrimenti è passata allo *stage* successivo [DDB15].

Bibliography

- [DDB15] Asit Kumar Datta, Madhura Datta, and Pradipta Kumar Banerjee. *Face Detection and Recognition*. Taylor & Francis, Oct. 28, 2015. 352 pp. URL: https://www.ebook.de/de/product/25204083/asit_kumar_datta_madhura_datta_pradipta_kumar_banerjee_face_detection_and_recognition.html.
- [Sch13] Robert E. Schapire. “Explaining AdaBoost”. In: *Empirical Inference*. Springer Berlin Heidelberg, 2013, pp. 37–52. DOI: 10.1007/978-3-642-41136-6_5.
- [VJ04] Paul Viola and Michael J. Jones. “Robust Real-Time Face Detection”. In: *International Journal of Computer Vision* 57.2 (May 2004), pp. 137–154. DOI: 10.1023/b:visi.0000013087.49260.fb.
- [Wan14] Yi-Qing Wang. “An Analysis of the Viola-Jones Face Detection Algorithm”. In: *Image Processing On Line* 4 (June 2014), pp. 128–148. DOI: 10.5201/ipol.2014.104.