

Visione - Assignment 2

Giovanni Castelli, Ivan Sollazzo, Gabriele Nicolò Costa

May 2024

1 Introduzione

Il progetto si divide principalmente in cinque parti:

1. Elaborazione dei dati ed estrazione del descrittore HOG per ciascun campione, a cura di Gabriele;
2. Addestramento di una Support Vector Machine (SVM) utilizzando i dati precedentemente elaborati, a cura di Giovanni;
3. Implementazione di una multi-scale Sliding Window per il rilevamento dei pedoni nelle immagini, a cura di Ivan;
4. Implementazione di Non Maximum Suppression (NMS) per selezionare la BBox migliore da una serie di BBox sovrapposte, a cura di Giovanni;
5. Realizzazione dei codici per gli iperparametri ottimali C e T, rispettivamente curati da Ivan e Gabriele.

Inoltre, è importante sottolineare che, nonostante la suddivisione dei ruoli, tutti i membri del team hanno contribuito al progetto nel suo complesso, dimostrando un impegno collettivo per il successo del lavoro.

2 Working Pipeline

Per l'esecuzione del problema procediamo nel seguente modo:

2.1 Data extraction

Procediamo ad estrarre le subimages sulla base delle annotazioni delle boundary boxes del dataset nel caso delle immagini "positive", estrazione casuale delle subimages nel caso di immagini "negative". Per la corretta esecuzione dell'addestramento e dell'estrazione, bisogna organizzare le cartelle nel seguente modo:

- Folder Immagini positive, copiate e scaricate dal link del dataset;
- Folder Immagini negative, ovvero le immagini fornite dal repository della Professoressa (su sharepoint);
- Nella main folder (dove eseguiamo il codice) devono essere presenti i file contenenti lo splitting delle immagini da considerare per le immagini positive;
- Nella cartella annotation invece vi sono tutte le annotazioni.

Tuttavia, in questa fase iniziale di estrazioni abbiamo tenuto in considerazione due versioni:

- Estrazione con preprocessing delle immagini, ovvero prendo ogni immagine e: trasformo in scala di grigi, equalizzo l'immagine, applico un filtro gaussiano e resize.
- Estrazione con preprocessing delle immagini ma solo applicando filtro gaussiano di blurring e resize dell'immagine.

Il motivo di questa scelta ci porta quindi alla realizzazione di due modelli separati, descritti sotto nei paragrafi successivi. Il perché abbiamo scelto invece di applicare una elaborazione delle immagini risiede nel fatto che abbiamo deciso di adottare la seguente euristica: se un pedone è presente lo deve essere sempre in condizione di luce invarianti.

2.2 Model Training

Procediamo quindi con l'addestramento del modello di classificatore, stavolta valutiamo due versioni differenti:

- Versione con dati standardizzati
- Versione con dati non standardizzati

Ovviamente, valutiamo per ognuno dei 4 modelli (vedi sottosezioni sotto). Per ogni modello valutiamo quindi il miglior iperparametro che varia in un range tra $[0.01, 0.1, 1]$. Alla fine selezioniamo il miglior modello sulla base dell'f1 score e per ogni modello il miglior iperparametro.

2.3 Utility functions

Il nostro modello per funzionare correttamente ha bisogno di alcune funzioni tra cui:

- Multiscale sliding window
- Non-Maxima suppression

2.4 Select best parameter T

La funzione di Non-Maxima-Suppression viene addestrata per determinare il miglior iperparametro T di thresholding sui valori di IoU delle finestre. Nel dettaglio, facciamo variare il valore di T nel range: [0.3, 0.4, 0.5]. Poi scelgo il miglior T sulla base degli f1-score del set di validazione.

2.5 Testing classifier

Testing del nostro classificatore sull'insieme di test e risultati finali, calcolando quindi i valori di f1-score.

2.6 Testing visivo su diverse immagini

Mostriamo delle foto arbitrarie per mostrare il successo e l'insuccesso del nostro modello rispetto a quello di open CV e alle annotazioni.

3 Selezione degli iperparametri

Gli iperparametri da scegliere in questa tipologia di detector sono:

- L'iperparametro C inerente al **classificatore lineare SVM**;
- L'iperparametro T inerente alla tecnica di **non maxima suppression**.

Di seguito riportiamo i modelli con e senza image processing. Per ognuno di questi valutiamo le tabelle con e senza standardizzazione dei dati. Notiamo come in entrambi i casi è necessario utilizzare standardizzazione dati in quanto il valore di precision e f1-score variano di poco al variare dell'iperparametro c di addestramento, questo ci dice che comunque siamo in una situazione di **underfitting**

Per quanto riguarda i valori di c scegliamo di far variare l'iperparametro in un range tra $[0.01, 0.1, 1]$. Il perché deriva dal fatto che noi vogliamo fare in modo che il nostro classificatore sia il più generalizzante possibile:

- Con valori alti di c rischiamo di non generalizzare troppo **underfitting**
- Con valori troppo bassi di c rischiamo di generalizzare troppo, quindi **overfitting**

Decidiamo di non prendere valori elevati di c per questioni di risorse di tempo e hardware per addestrare.

Inoltre, per quanto riguarda il valore dell'iperparametro T decidiamo di far variare questo nel range $[0.3, 0.4, 0.5]$ perché idealmente noi vogliamo che vengano ridotte le windows che si sovrappongono con un valore di IoU che non sia troppo alto o troppo basso, dunque in maniera empirica vediamo cosa varia.

3.1 Modello senza image preprocessing

3.1.1 Con standardizzazione

Di seguito sono riportate le tabelle per (rispettivamente):

- Risultati dell'iperparametro T
- Risultati dell'iperparametro c
- Risultati di CCS sul test-set
- Risultati di OpenCV sul test-set

C	TP	FP	FN	Precision	Recall	F1 score
0.01	18954	828	714	0.9581	0.9636	0.9609
0.1	18958	953	710	0.9521	0.9639	0.9579
1.0	18260	936	1408	0.9512	0.9284	0.9396

T	TP	FP	FN	Precision	Recall	F1 score
0.3	1477	3561	18191	0.2931	0.0750	0.1195
0.4	1840	4709	17828	0.2809	0.0935	0.1403
0.5	2339	6425	17329	0.2668	0.1189	0.1645

precision ccs	recall ccs	f1 score ccs	TP ccs	FP ccs	FN ccs
0.2530	0.1184	0.1613	2073	6120	15435

precision cv	recall cv	f1 score cv	TP cv	FP cv	FN cv
0.1003	0.0185	0.0313	325	2913	17183

3.1.2 Senza standardizzazione

C	TP	FP	FN	Precision	Recall	F1 score
0.01	18780	934	888	0.9526	0.9548	0.9537
0.1	18740	918	928	0.9533	0.9528	0.9530
1.0	18732	925	936	0.9529	0.9524	0.9526

3.2 Modello con image preprocessing

3.2.1 Standardizzazione

Di seguito sono riportate le tabelle per (rispettivamente):

- Risultati dell'iperparametro T
- Risultati dell'iperparametro c
- Risultati di CCS sul test-set
- Risultati di OpenCV sul test-set

C	TP	FP	FN	Precision	Recall	F1 score
0.01	18724	919	944	0.9532	0.9520	0.9526
0.1	18675	930	993	0.9525	0.9495	0.9510
1.0	17981	1007	1687	0.9469	0.9142	0.9303

T	TP	FP	FN	Precision	Recall	F1 score
0.3	1288	3764	18380	0.2549	0.0654	0.1042
0.4	1596	4910	18072	0.2453	0.0811	0.1219
0.5	1980	6645	17688	0.2295	0.1006	0.1399

precision ccs	recall ccs	f1 score ccs	TP ccs	FP ccs	FN ccs
0.2155	0.1032	0.1395	1807	6578	15701

precision cv	recall cv	f1 score cv	TP cv	FP cv	FN cv
0.1003	0.0185	0.0313	325	2913	17183

3.2.2 Senza standardizzazione

C	TP	FP	FN	Precision	Recall	F1 score
0.01	18780	934	888	0.9526	0.9548	0.9537
0.1	18740	918	928	0.9533	0.9528	0.9530
1.0	18732	925	936	0.9529	0.9524	0.9526

4 Classificatori

Per quanto riguarda i test visivi, dai dati delle tabelle della precedente sezione concludiamo che il miglior modello che classifica pedoni che abbiamo trovato è quello **senza image processing e con standardizzazione dei dati**, difatti ha un valore di precisione elevato e più alto rispetto al classificatore di OpenCV, confrontando entrambi i dati su campioni del test set.

4.1 Verifica visiva con immagini

In questa sezione commenteremo vari esempi dei classificatori CCS (ovvero il nostro) e quello di OpenCV, paragonandoli alle annotazioni già presenti.

Le BBox sono suddivise in colori, nel dettaglio:

- **Blu:** Classificatore CCS
- **Rosso:** Classificatore OpenCV
- **Verde:** Annotazioni già presenti

Prendendo in maniera randomica alcune immagini dal nostro dataset, osserviamo che il nostro classificatore funziona discretamente bene e riesce quantomeno a rilevare le persone dove sono rispetto alle annotazioni.

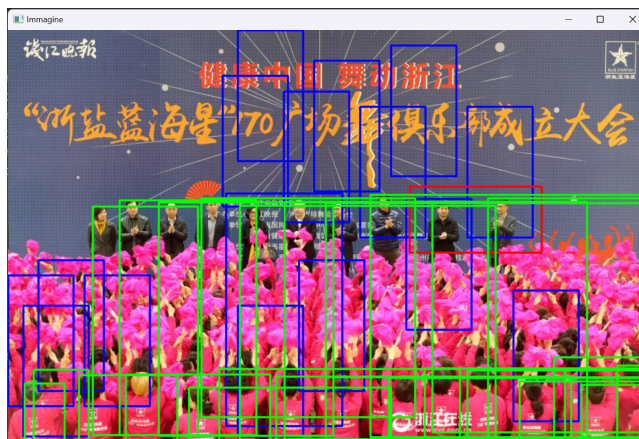


Figure 1: Prima immagine di prova.

Con riferimento alla figura 1, si può notare che il classificatore riesce più o meno a rilevare i pedoni presenti nelle annotazioni, con tuttavia la presenza di alcuni falsi positivi.

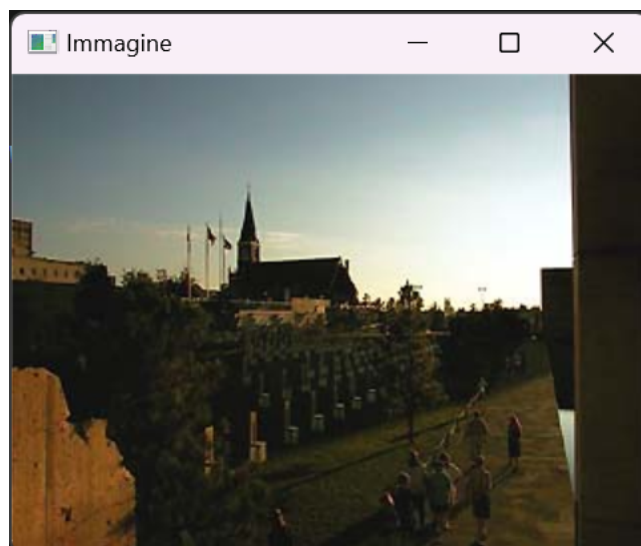


Figure 2: Seconda immagine di prova.

Nel caso dell'immagine di 2 non è stata rilevata alcuna presenza di pedoni e/o falsi positivi.



Figure 3: Terza immagine di prova.

Lo stesso risultato si ottiene altresì nell'immagine di figura 2, così come nell'immagine di figura 3.

Osserviamo invece un caso in cui il classificatore non rileva completamente i pedoni, ma bensì altri elementi dell'immagine. Un caso del genere lo si ha nell'immagine di figura 4.

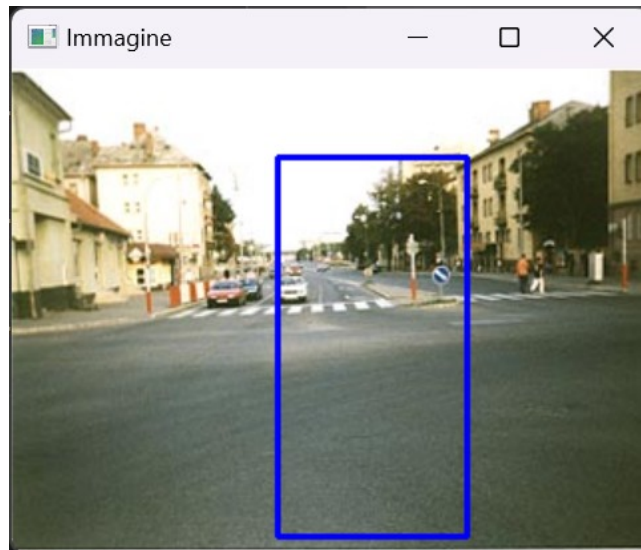


Figure 4: Quarta immagine di prova.

Infine, come ultimo test è stato testato il classificatore con una foto non appartenente al test set.

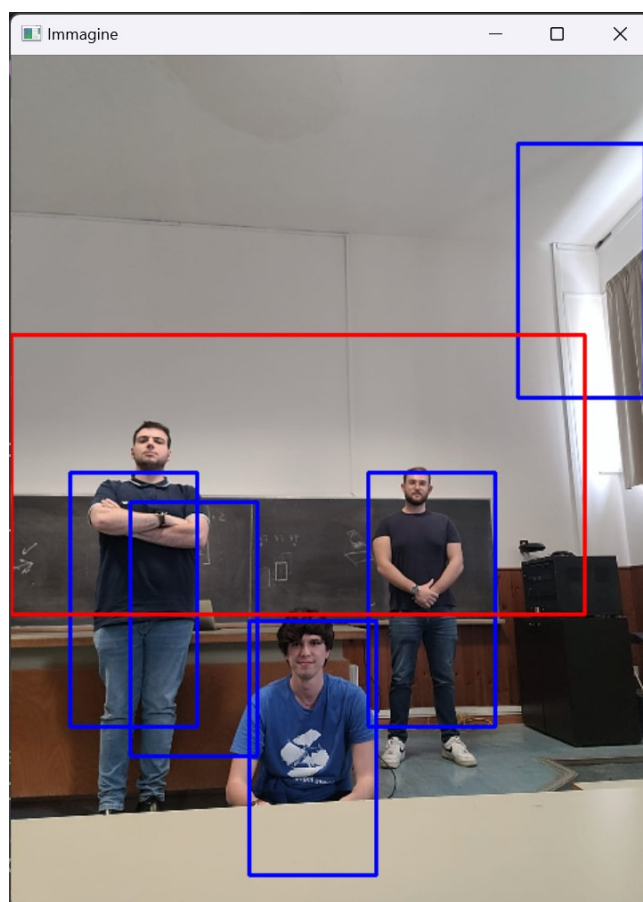


Figure 5: Quinta immagine di prova.

Si può osservare in questo caso che i tre soggetti sono stati rilevati correttamente, ad eccezione di un falso positivo in alto a destra.