

# Relazione Retrieval using Bag of Visual Words

Marco De Stefano

March 2025

## 1 Introduzione

Abbiamo visto, quindi, che si può utilizzare il modello BoW (*Bag of Words*) per classificare e identificare immagini a partire da un dizionario di “parole visive” (*visual words*).

In particolare, le parole del nostro vocabolario sono identificate come **feature locali** estratte dalle immagini.

Per costruire il vocabolario visivo, si raccolgono tutte le feature locali da un insieme di immagini e si applica un algoritmo di clustering (come il *k-means*) per **quantizzare** lo spazio delle feature. Il risultato è un insieme di centroidi, ciascuno dei quali rappresenta una *visual word*. Questo insieme viene chiamato *visual vocabulary*, dove ogni parola è quindi il centroide di un cluster di descrittori simili.

Una volta ottenuto il vocabolario, ogni immagine viene rappresentata tramite un **istogramma di occorrenze**, chiamato *Bag of Visual Words*: per ogni feature dell’immagine, si identifica il centroide più vicino (utilizzando ad esempio la distanza euclidea o la similarità del coseno) e si incrementa il contatore corrispondente.

Il risultato è un vettore di lunghezza pari al numero di parole visive, che rappresenta quante volte ciascuna *visual word* è stata “attivata” in quell’immagine. Questo vettore può quindi essere usato come input per algoritmi di classificazione, clustering o *image retrieval*.

In fase di confronto tra immagini (ad esempio, per trovare immagini simili), si confrontano i rispettivi vettori BoW utilizzando una metrica di distanza tra vettori (come la distanza euclidea o la similarità del coseno). Un’immagine sarà considerata simile a un’altra se i rispettivi istogrammi di parole visive risultano vicini nello spazio.

In questo lavoro esploriamo il problema dell’image retrieval, ovvero il compito di, data un’immagine query, riordinare un insieme di immagini in modo che quelle più simili alla query appaiano in cima alla lista.

Ci concentriamo sul modello BoW, applicando la quantizzazione delle feature per rendere il retrieval più efficiente anche in presenza di immagini contenenti

migliaia di descrittori locali (come i SIFT, vettori di 128 dimensioni). Come dataset utilizziamo il Holidays dataset (INRIA), che contiene 1491 immagini annotate e 500 immagini query. I descrittori SIFT sono già stati estratti e verranno usati per costruire il vocabolario visivo e per rappresentare ciascuna immagine come un istogramma di parole visive.

## 2 Analisi

Nel laboratorio oggetto di analisi, vengono studiate diverse caratteristiche legate alle prestazioni degli algoritmi implementati all'interno del notebook fornito.

Il notebook utilizza un archivio contenente diversi *visual vocabularies*, con dimensioni variabili da 100 fino a 200,000 parole visive, ottenute da un sottinsieme del dataset **Flickr60k**. L'utente può selezionare la dimensione del vocabolario da utilizzare.

Le *visual words*, ricordiamo, corrispondono ai **centroidi** ottenuti tramite clustering (ad esempio, k-means) dei descrittori locali.

Per ogni immagine è disponibile una matrice di dimensione  $N \times 128$ , dove  $N$  rappresenta il numero di descrittori locali **SIFT**, ciascuno dei quali ha 128 dimensioni.

Prima di iniziare l'Esercizio 1, il notebook carica in memoria tutte le *visual words*, che vengono memorizzate in un array chiamato **centroids** di dimensione **n\_centroids**  $\times$  128.

### Esercizio 1

Nel primo esercizio viene calcolata la rappresentazione *Bag of Visual Words* (BoW) per una singola immagine, utilizzando due diverse strategie di assegnamento:

- **Hard Assignment:**

In questo approccio, per ogni descrittore della prima immagine si calcola la distanza (secondo una metrica specifica, ad esempio **euclidea** o **coseno**) rispetto a tutti i centroidi. Viene selezionato il centroide con **massima somiglianza** (nel caso della distanza coseno) oppure **minima distanza** (nel caso della distanza euclidea).

Successivamente, si incrementa il valore del bin dell'istogramma corrispondente all'indice del centroide selezionato.

*Nota:* con la **similarità coseno**, due vettori simili producono un valore vicino a 1; con la **distanza euclidea**, invece, maggiore è la somiglianza, più basso è il valore.

- **Soft Assignment:**

In alternativa, si utilizza un assegnamento "morbido", basato su una funzione di tipo **softmax**, definita come:

$$\text{softmax}(\mathbf{z})_i = \frac{e^{-||x-\mu_i||^2}}{\sum_j e^{-||x-\mu_j||^2}}$$

Dove  $x$  è un descrittore e  $\mu_i$  è il centroide  $i$ -esimo.

Questa funzione produce un **vettore di probabilità**, che esprime quanto un descrittore appartiene a ciascun centroide. Il vettore risultante viene quindi **accumulato** direttamente nell'istogramma finale.

A seguire, analizziamo i risultati ottenuti utilizzando un vocabolario di **2000 visual words**.

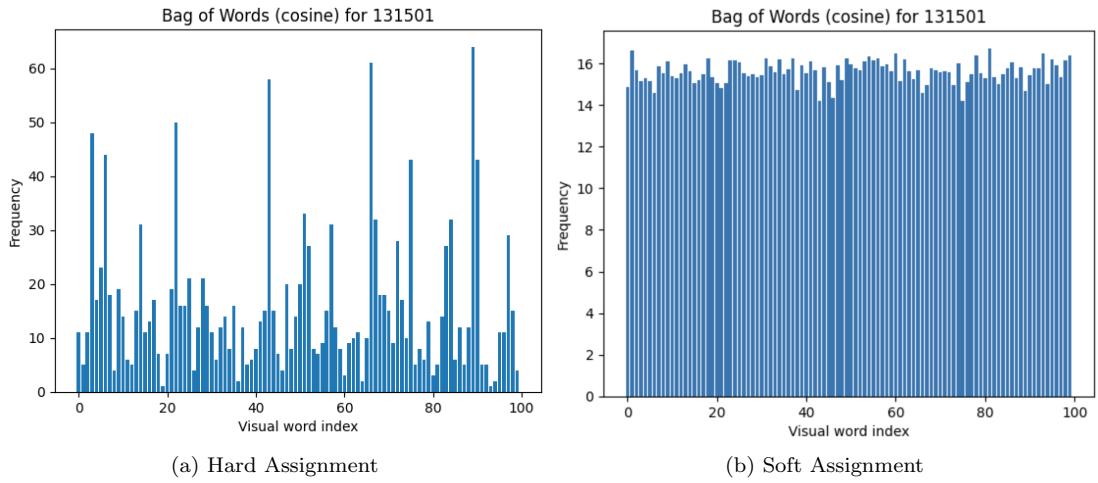


Figure 1: Confronto tra Hard Assignment e Soft Assignment

Vediamo che nel caso (a) il risultato ha picchi più accentuati, indicando un clustering più distintivo mentre il caso(b) ha un comportamento più uniforme.

Questo accade perché applicare la softmax su tutti i centroidi tende a distribuire in modo troppo uniforme le probabilità tra i vari cluster, causando una sorta di “sfocatura” nei risultati. Per ovviare a questo, è utile applicare la softmax solo su un sottoinsieme di centroidi vicini, utilizzando la funzione **Nearest Neighbor**. Aumentando il parametro in ingresso  $n_{neighbor}$ , si ottiene un comportamento simile a quello del soft assignment tradizionale, in cui i descrittori sono distribuiti tra più centroidi. Riducendo il parametro  $n_{neighbor}$ , invece, si effettua un “filtraggio” che considera solo i  $n_{neighbor}$  cluster più vicini, rafforzando la distinzione tra di essi. Si ottiene quindi la figura 2 che ha un comportamento simile all’ Hard Assignment di prima.

### 2.0.1 Tempo di esecuzione

Bisognerebbe notare il tempo di esecuzione per ogni dimensione di vocabolario. Per esigenza di tempo terremo conto solo delle tempistiche per i dizionari di

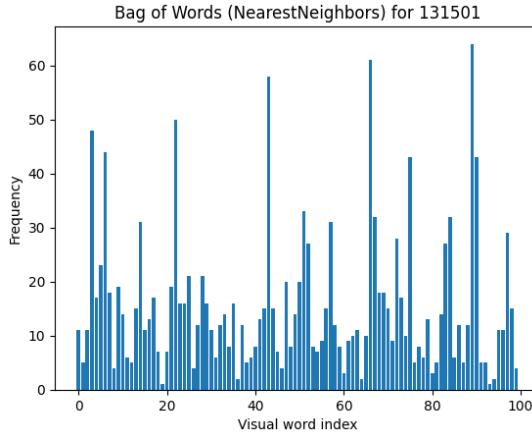


Figure 2: Creazione BoW con NearestNeighbor e Soft Assignment

dimensioni 100, 2000, 20000: Prima di tutto, si vuole tenere conto che questi

Dimensione Dizionario	Funzione	Assegnamento	Tempo	Risultato
20K	Fatta a mano	Soft	48s	Cattivo
20K	Fatta a mano	Hard	41.71s	Buono
20K	NearestNeighbor	Hard	28s	Buono
20K	NearestNeighbor	Hard	30s	Buono
2K	Fatta a Mano	Hard	2.31s	Buono
2K	Fatta a Mano	Soft	5s	Cattivo
2K	NearestNeighbor	Soft	2.9s	Buono
2K	NearestNeighbor	Hard	2.78s	Buono
100	Fatta a mano	Soft	0.34s	Cattivo
100	Fatta a mano	Hard	1.70s	Buono
100	NearestNeighbor	Hard	0.29s	Buono
100	NearestNeighbor	Soft	1.08s	Buono

Table 1: Confronto tra diversi metodi di calcolo di una Bag of Word

risultati sono stati fatti tenendo conto della metrica coseno e non quella euclidea. Si può notare che, a partire dal vocabolario più grande, il tempo per calcolare una BoW è minori nei processi in cui si utilizza la funzione Nearest Neighbor, questo perchè la funzione in questione è ottimizzata per questi tipi di calcoli. Vediamo poi l'effetto che dicevamo prima nel caso dei Soft senza considerare un gruppo ristretto di vicini:

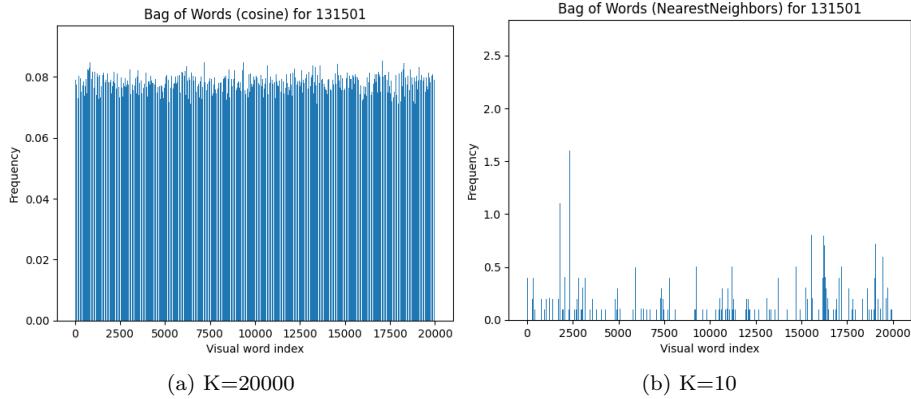


Figure 3: Confronto tra Soft Assignment (senza neighborhood) e con

## 2.1 Esercizio 2

Mentre nel primo esercizio abbiamo calcolato una Bag of Word di una immagine, adesso quello che si vuole, quindi, è calcolare le BoW di tutte le 1491 immagini utilizzando come prima le nostre dimensioni 100, 2000, 20K. Ci aspettiamo, ovviamente, dei tempi di completamento che saranno crescenti via via che la dimensione dei vocabolari cresce.

Dimensione vocabolario	Tempo per il calcolo delle BoWd
20000	22.58 min
2000	138.82 s
100	18.81 s

Table 2: Confronto del tempo di esecuzione di tutte le BoW al variare della dimensione dei dizionari

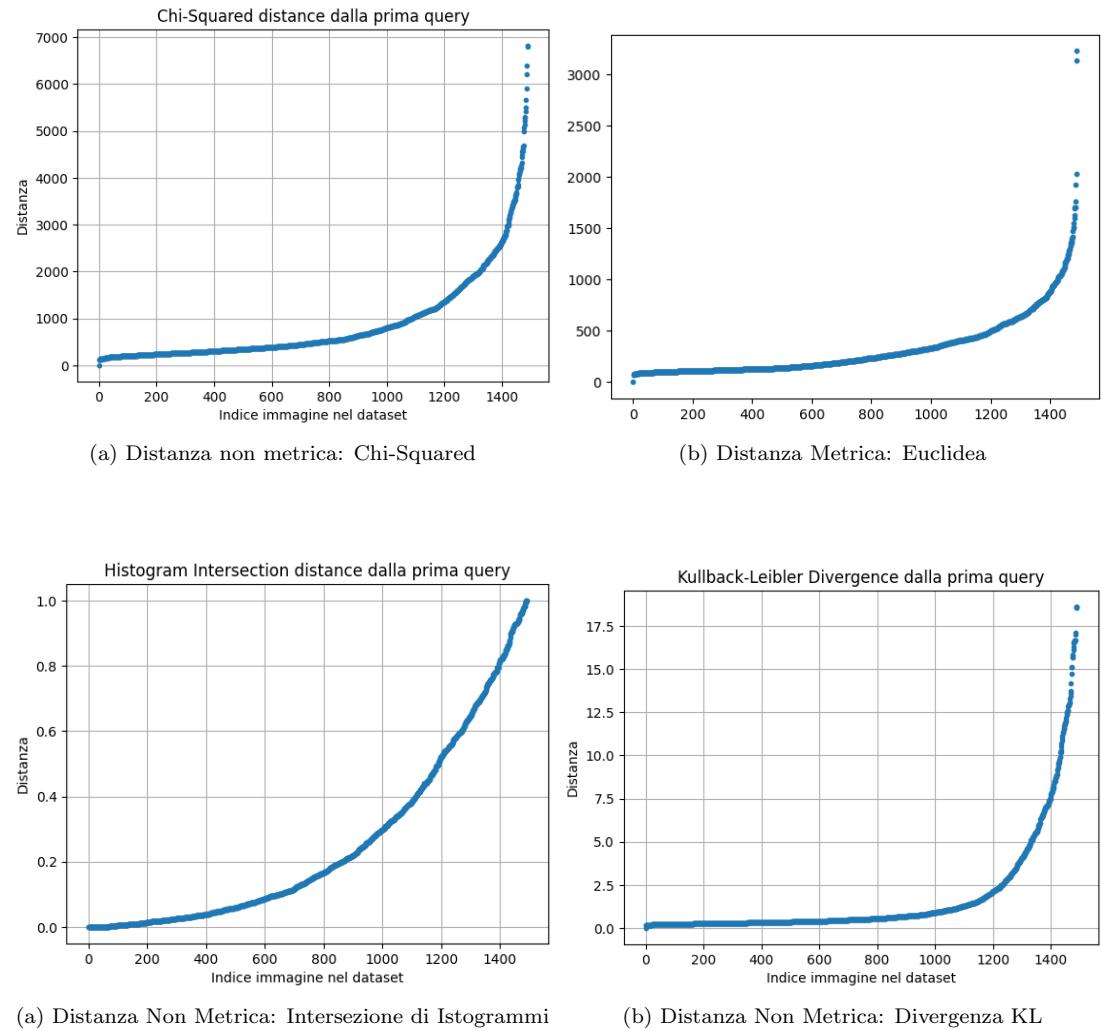
Questi tempi sono stati ottenuti utilizzando l'Hard Assignment, una tecnica in cui ciascun descrittore dell'immagine viene assegnato unicamente al centroide più vicino. Di conseguenza, se due centroidi risultano equidistanti da un descrittore, solo uno dei due verrà selezionato — tipicamente in modo arbitrario. Diversamente, nel Soft Assignment, il contributo del descrittore viene distribuito tra più centroidi, proporzionalmente alla loro distanza.

## 2.2 Esercizio 3

Avendo calcolato le Bag of Words (BoW) per l'intero dataset, possiamo selezionare alcune di esse e eseguire un retrieval utilizzando la distanza metrica euclidea tramite il metodo Nearest Neighbor (con  $n_{neighbor} = 1491$ ).

In aggiunta a questa, possiamo calcolare la distanza tra le BoW utilizzando distanze non metriche, come l'intersezione tra istogrammi, la distanza Chi-Squared, o la divergenza di Kullback-Leibler.

I grafici riportati mostrano i risultati ottenuti confrontando la BoW della prima immagine nel set di query con le 1491 BoW delle immagini del dataset, utilizzando le distanze sopra menzionate.



Dai grafici possiamo osservare che la Divergenza di Kullback-Leibler mantiene distanze relativamente basse e mostra una maggiore corrispondenza per la maggior parte del dataset, rispetto ad altre metriche che presentano una crescita più uniforme delle distanze. Notiamo che ogni grafico inizia dal punto di coordinate  $(0, 0)$  poiché le distanze calcolate vengono restituite come un array ordinato in ordine crescente. Questo significa che, essendo le query contenute nel dataset, ci sarà sempre una coppia con distanza pari a zero, che corrisponde alla query stessa.

Proviamo a ridurre il numero di neighbor per ogni calcolo della distanza, facendo così è come se si zoomasse sul grafico. Lo facciamo, modificando il numero di neighbor ed in questo caso lo faremo per tre casi (300,700,1000): Nella figura 6

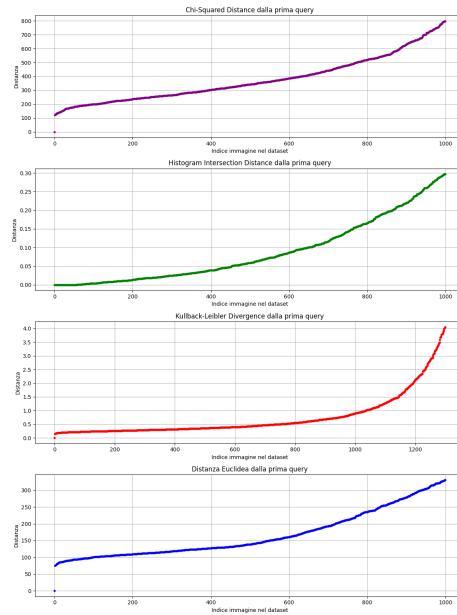


Figure 4: Grafico distanze con 1000 vicini

notiamo una peculiare caratteristica dell' Histogram Intersection, ovvero la sua evoluzione a gradini rispetto alle altre distanze.

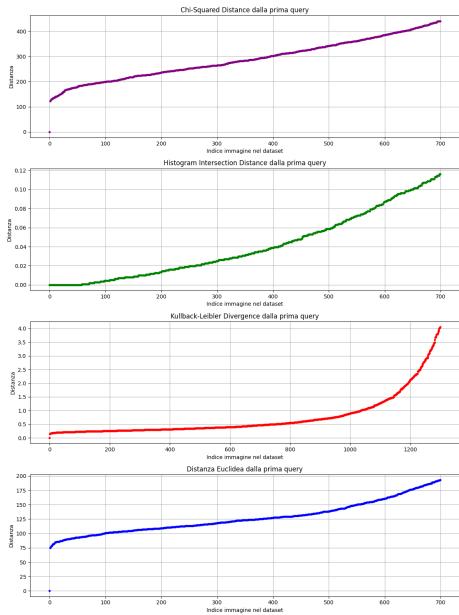


Figure 5: Grafico distanze con 1000 vicini

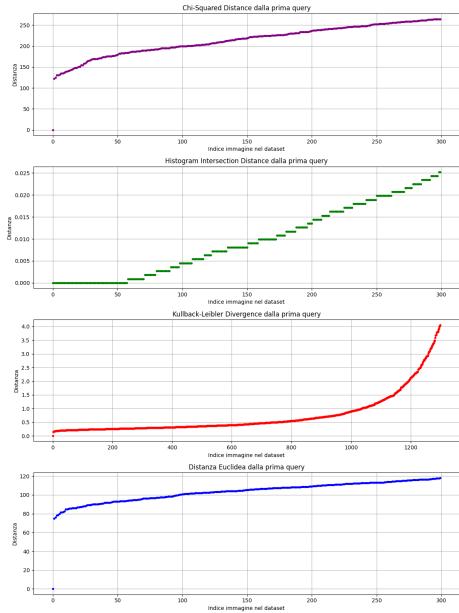


Figure 6: Grafico distanze con 1000 vicini

## 2.3 Esercizio 4

Se si aumenta il dizionario cosa succede? Il primo pensiero va al costo computazionale che si alza vertiginosamente al crescere della dimensione del vocabolario. Oltre a questo, vuol dire che se il vocabolario è diventato più grande, conterrà più visual word riconoscendo più patches delle foto messe come query. Le misure di prestazioni del nostro sistema di retrieval è il mAP (mean Average Precision) , cambiando il dizionario vediamo il mAP si comporterà in un certo modo.

In particolare, la metrica di precisione media (mAP) può variare significativamente in base alla dimensione del vocabolario utilizzato. La tabella ?? mostra i risultati di mAP ottenuti per diverse dimensioni di vocabolario, inclusi 100, 1000 e 2000 termini.

Dimensione Dizionario	Distanza	mAP
100	Euclidean	0.334
100	Chi Square	0.371
100	KL Divergence	0.290
100	Histogram Intersect	0.032
1000	Euclidean	0.383
1000	Chi Square	0.429
1000	KL Divergence	0.061
1000	Histogram Intersect	0.028
2000	Euclidean	0.338
2000	Chi Square	0.368
2000	KL Divergence	0.036
2000	Histogram Intersect	0.030

Table 3: Confronto del mAP con diverse distanze al variare della dimensione del dizionario

Come mostrato, la dimensione del vocabolario da 100 parole genera il miglior mAP, seguita da quella con 6000, mentre la dimensione intermedia di 2000 risulta la meno performante. Questo comportamento è il risultato di un possibile equilibrio tra generalizzazione e precisione.

### 2.3.1 Vocabolario piccolo (100 parole)

Un vocabolario con un numero ridotto di parole visive tende ad essere meno preciso ma più robusto rispetto ai dati. Gli istogrammi BoW risultano più densi e meno influenzati dal rumore questo può consentire una **migliore generalizzazione**, riducendo la variabilità tra immagini simili, e contribuendo così a una maggiore precisione nelle prime posizioni del ranking. Pertanto, un vocabolario ridotto può evitare l'**overfitting**, migliorando il mAP. Ricordiamo che l'overfitting è un fenomeno che si ha quando il modello si adatta troppo bene ai

dati in input imparando anche il rumore.

### 2.3.2 Vocabolario medio (2000 parole)

Con una dimensione di vocabolario intermedia, il modello potrebbe soffrire di un fenomeno di **ambiguità**. Le parole visive sono più specifiche, ma non sufficientemente per ridurre efficacemente il rumore e le somiglianze non pertinenti tra immagini. In questo caso, la distanza tra gli histogrammi BoW tende a riflettere più il rumore che le reali somiglianze tra le immagini, portando a un incremento dei falsi positivi. Questo fenomeno riduce la precisione media e, di conseguenza, il mAP.

### 2.3.3 Vocabolario grande (20000 parole)

Con un vocabolario molto ampio, il modello diventa più **discriminativo**, in grado di catturare dettagli più fini e separare meglio le immagini simili. Tuttavia, questa maggiore precisione può anche portare a un **overfitting** se il modello non è sufficientemente robusto o se i dati di addestramento sono insufficienti. Inoltre, l'aumento della dimensione del vocabolario comporta una maggiore complessità computazionale, con un potenziale aumento del rumore semantico. Il mAP medio risulta inferiore rispetto alla dimensione di 100 parole, poiché il vocabolario più grande potrebbe non riuscire a generalizzare altre caratteristiche semantiche comuni in modo altrettanto efficace.

## 2.4 Retrieval

Vediamo, quindi, i risultati del retrieval facendo vedere i risultati con Falsi Positivi e i risultati perfetti.

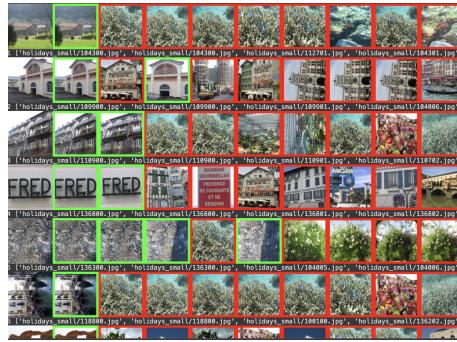


Figure 7: Seconda e Quarta riga hanno un falso positivo

In queste due foto notiamo che si ha due falsi positivi nella seconda e nella quarta riga dati dal fatto che una foto evidenziata di rossa è ordinata prima rispetto a quella successiva che è evidenziata di verde. Il rosso rappresenta che la foto non corrisponde alla query mentre il verde invece sì. La foto che non è

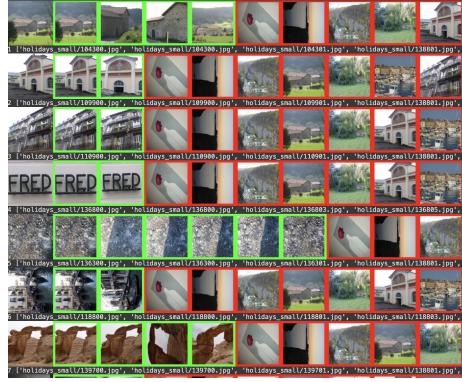


Figure 8: Sono matchate tutte le foto

evidenziata è l'immagine query, più si va verso le foto a destra e meno si ha una corrispondenza con la query.

## 2.5 Esercizio 5

Come ultimo esercizio, cerchiamo di ricalcolarci tutte le Bag of Words utilizzando l'algoritmo di clustering chiamato **KMeans**, negli esercizi precedenti le scaricavamo direttamente da un dizionario di dimensioni varie. Mentre adesso quello che si fa è calcolarsi i centroidi utilizzando il dataset di immagini. I centroidi li calcoliamo, quindi, con il seguente algoritmo e li mettiamo in un array chiamato `bow_array`. A quel punto calcoliamo le distanze utilizzando sia metriche che non metriche che abbiamo visto precedentemente. Con queste distanze e gli indici in output dalle funzioni distanza fatte, si calcola il mAP. Stampiamo i risultati ottenuti per ogni distanza fatta.

La tabella mostra i valori di *mean Average Precision (mAP)* ottenuti variano la dimensione del dizionario visivo (numero di centroidi) e il tipo di distanza utilizzata per confrontare le rappresentazioni BoW.

- **Influenza della dimensione del dizionario:** all'aumentare della dimensione del dizionario da 10 a 1000 si osserva un aumento delle mAP, mentre con 10000 elementi si nota un leggero calo o stabilizzazione, suggerendo possibile overfitting o introduzione di rumore.
- **Confronto tra distanze:** le distanze *Chi Squared* e *KL Divergence* risultano le più efficaci, specialmente con dizionari di media dimensione. La *Histogram Intersection* ha mostrato performance inferiori in quasi tutti i casi.
- **Distanza Euclidea:** pur essendo una distanza semplice, la metrica Euclidea ha mostrato buoni risultati, in particolare con dizionario di dimensione 1000, raggiungendo lo stesso valore massimo di mAP della *Chi Squared*.

Dimensione Dizionario	Distanza	mAP
10	Euclidean	0.161
10	Chi Square	0.179
10	KL Divergence	0.179
10	Histogram Intersect	0.011
100	Euclidean	0.321
100	Chi Square	0.359
100	KL Divergence	0.275
100	Histogram Intersect	0.032
1000	Euclidean	0.321
1000	Chi Square	0.359
1000	KL Divergence	0.275
1000	Histogram Intersect	0.032
10000	Euclidean	0.264
10000	Chi Square	0.226
10000	KL Divergence	0.053
10000	Histogram Intersect	0.031

Table 4: Confronto del mAP con diverse distanze al variare della dimensione del dizionario

In conclusione, le migliori prestazioni si ottengono generalmente con dizionari di dimensione intermedia (circa 1000) e con metriche specializzate per istogrammi come la *Chi Squared*.

### 3 Conclusioni

In questa sezione finale abbiamo osservato che, per calcolare le Bag of Words, è molto utile utilizzare la funzione NearestNeighbors, che ottimizza notevolmente le prestazioni rispetto a un’implementazione basata direttamente sulla teoria.

Abbiamo poi discusso un aspetto importante del Soft Assignment: se si utilizzano troppi cluster (ad esempio, l’intero dataset come centri), si può introdurre rumore, ottenendo istogrammi troppo uniformi. È quindi preferibile utilizzare un numero limitato di neighbors per ottenere descrizioni più significative.

In seguito, abbiamo analizzato diverse distanze, sia metriche che non metriche, visualizzandole attraverso grafici: sull’asse x sono rappresentati gli indici delle immagini e sull’asse y la distanza tra l’immagine query e tutte le altre.

Infine, abbiamo calcolato le mean Average Precision (mAP) usando varie metriche di distanza, confrontando i risultati ottenuti con un dizionario pre-addestrato scaricato da internet e uno generato da noi a partire dallo stesso dataset.