

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA

CAMPUS DI CESENA

SCUOLA DI INGEGNERIA E ARCHITETTURA

**CORSO DI LAUREA IN INGEGNERIA ELETTRONICA INFORMATICA
E TELECOMUNICAZIONI**

TITOLO DELL' ELABORATO

**Descrittori e metriche di similarità
per applicazioni CBIR.**

Elaborato in
Sistemi Informativi

Relatore:
Dott.ssa Alessandra Lumini

Presentata da:
Nicolò Cangini

Sessione III
Anno Accademico 2013/14

Keyword: Content Based Image Retrieval, Descrittori, Metriche di Similarità.

“Stay Hungry, Stay Foolish”

Steve Jobs

Sommario

Introduzione	9
Capitolo 1	11
Introduzione al problema CBIR.....	11
1.1 Information Retrieval.....	11
1.2 Dati Multimediali e Image Retrieval.....	11
1.3 Il Content Based Image Retrieval (CBIR)	12
1.4 Il Sistema CBIR	13
1.4.1 CBIR e Relevance Feedback.....	14
1.5 Panoramica sui sistemi CBIR esistenti.....	15
Capitolo 2	19
Descrittori e Misure di Similarità	19
2.1 Introduzione ai metodi	19
2.2 Descrittori del colore	22
2.2.1 Istogramma RGB e HSV	23
2.2.2 Color Coherence Vector.....	28
2.3 Descrittori di Forma.....	29
2.3.1 Istogramma dei gradienti orientati.....	29
2.4 Descrittori di Tessitura.....	31
2.4.1 Descrittori Globali di tessitura.....	33
2.4.1.1 Tamura Features	33
2.4.1.2 Average Hash	34
2.4.1.3 Perceptive Hash.....	34
2.4.1.4 Different Hash	35
2.4.1.5 Haralick Texture Features	35
2.4.2 Descrittori Locali di Tessitura.....	36
2.4.2.1 Filtri di Gabor.....	36
2.4.2.2 LBP	38
2.4.2.3 LPQ.....	39
2.5 Descrittori su Key Point (punti salienti)	40
2.5.1 SIFT	40
2.5.1.1 Misure di Similarità applicabili al SIFT.....	41

2.5.1.2	RANSAC come metodo per il matching dei Key Point	42
2.5.2	SURF	44
2.6	Misure di Similarità.....	46
Capitolo 3	48
Indicatori di Prestazione	48
3.1	Introduzione ai Performance Indicator	48
3.1	Receiver Operator Characteristic	49
3.2	Area Under Curve	52
3.3	Accuracy ottenuta con il K-Nearest-Neighbors	53
3.4	Grafico Genuine Impostor	54
Capitolo 4	55
Risultati Sperimentali	55
4.1	Introduzione ai test.....	55
4.2	Descrizione dei Dataset utilizzati.....	56
4.3	Dati sperimentali ottenuti.....	64
4.3.1	Caltech Building	67
4.3.2	ZuBud.....	69
4.3.3	Shieffield.....	71
4.3.4	Object ZuBud	73
4.3.5	WangDB	75
4.3.6	ItaLa.....	77
4.3.7	ColorBrodatz.....	79
4.3.8	Risultati del metodo 3×3	81
4.4	Valutazioni dei dati sperimentali.....	89
Capitolo 5	90
Conclusioni	90
Bibliografia	91
Ringraziamenti	95

Introduzione

L'evoluzione tecnologica alla quale abbiamo assistito negli ultimi anni e che tutt'oggi continua sotto i nostri occhi, ha apportato grandi cambiamenti nella vita quotidiana delle persone, inserendosi in qualsiasi attività giornaliera semplificandone e migliorandone lo svolgimento.

Nel frattempo la continua espansione della rete ha portato ad una crescita esponenziale della creazione e condivisione di dati multimediali, tra i quali spiccano le immagini, che grazie al sempre più comune uso di smartphone, fotocamere digitali e memorie virtuali eccezionali, richiedono non solo applicazioni maneggevoli in grado di archivarle e/o condividerle, ma anche di metodi in grado di elaborarle e permetterne un utilizzo pratico in ambiti lavorativi e di supporto alle nostre attività.

Questo elaborato nasce con l'obiettivo di analizzare l'ambito dei sistemi Content Based Image Retrieval (CBIR), ovvero i sistemi di gestione delle immagini basati sul contenuto, con lo scopo di approfondirne i concetti, le importanti applicazioni pratiche che ne susseguono e i metodi utilizzabili per un buon sviluppo.

Lo scopo principale di questa tesi è quello di raccogliere informazioni sulle qualità degli algoritmi e sulle prestazioni di descrittori e misure di similarità applicabili ad uno specifico dataset di immagini, in modo da riconoscere quali metodi si adattano più efficacemente ad un determinato problema.

Si è quindi cominciato con un'analisi di articoli e sistemi proposti in letteratura, per poi passare ad una fase di testing di alcuni dei più noti algoritmi su alcuni problemi di classificazione di immagini attraverso i quali si è valutata l'efficacia e il funzionamento dei metodi principali nelle diverse applicazioni. Per questa fase sperimentale è stato utilizzato un framework di test implementato su MatLab sul quale sono stati implementati alcuni algoritmi.

Il lavoro è stato suddiviso in 5 capitoli: nel primo capitolo viene introdotto l'enorme ambito CBIR, riassumendone gli studi svolti dalla ricerca sul recupero e l'analisi delle immagini digitali, sottolineando l'importanza dei principali cambiamenti storici e descrivendo il funzionamento teorico di un sistema e i principali moduli che lo compongono, sono inoltre descritti aspetti applicativi già ampiamente utilizzati ed altri ancora in corso di studio; successivamente nel secondo saranno passati in rassegna i principali metodi di estrazione di informazioni delle immagini e verranno analizzate le principali misure di similarità ; nel terzo capitolo saranno introdotti gli indicatori di performance di cui ci siamo avvalsi per la valutazione dei nostri test.

Infine nel capitolo 4 sono presentati i test sperimentali e i dati che ne abbiamo ricavato, inoltre nello stesso capitolo è anche riportata una descrizione dei dataset e i loro attributi e per ognuno di essi sarà disponibile la raccolta di grafici, consultabili per l'approfondimento dei risultati. Il capitolo 5 è dedicato alle conclusioni e ai possibili sviluppi futuri.

Capitolo 1

Introduzione al problema CBIR

1.1 Information Retrieval

La grande maggioranza di dati digitali giace all'interno di database, o server, i quali, nel caso del World Wide Web, sono connessi costantemente alla rete.

Quando parliamo di information retrieval (IR), ci riferiamo ad un problema client-server nel quale lato client un utente agisce attraverso l'utilizzo di una query, interrogando il server al fine di ottenere, visualizzare o inserire informazioni.

L'utilizzo e la necessità di informazioni digitalizzate e la previsione futura di un loro continuo incremento, hanno portato negli ultimi tempi ad un aumento degli studi in questo settore, soprattutto per quanto riguarda l'efficienza dei sistemi per l'archiviazione e l'accesso ai dati, senza tralasciare aspetti legati alla sicurezza e alla protezione di informazioni sensibili.

1.2 Dati Multimediali e Image Retrieval

L'inarrestabile sviluppo tecnologico degli ultimi anni, ha reso possibile la diffusione di sorgenti di dati multimediali con capacità di produrre ed elaborare tali informazioni con estrema velocità e semplicità; inoltre con le possibilità odierne di condividere contenuti multimediali continuamente e con estrema facilità si è reso necessario l'incremento degli sforzi compiuti nella realizzazione e gestione di database multimediali, introducendo così l'*image retrieval* che come suggerisce la parola stessa vede come dati in analisi le immagini.

Un sistema di image retrieval è strutturato in modo da ricercare e restituire immagini da un database multimediale. I metodi tradizionali di recupero di immagini prevedono di associare manualmente ai file delle parole chiavi o una descrizione testuale in modo da effettuare la ricerca attraverso il testo. Questa metodologia risulta dispendiosa soprattutto in termini di tempo e inefficace in termini di qualità delle risposte, per cui si è resa necessaria la progettazione di funzioni basate sull'annotazione automatica delle immagini.

1.3 Il Content Based Image Retrieval (CBIR)

Con il termine CBIR (content based image retrieval) si indicano i sistemi di gestione e ricerca di immagini basati sul confronto di una o più caratteristiche intrinseche dell'immagine (tessitura, colore, forma, etc.).

La ricerca nell'ambito dell'image retrieval ha incominciato a svilupparsi intorno al 1970; all'inizio gli studi erano concentrati esclusivamente sulla ricerca testuale, con framework utilizzabili solo con criteri basati su nomi, dove le immagini necessitavano di essere nominate come prima cosa e solo dopo potevano essere ritrovate all'interno del database [1].

La limitazione principale di questo tipo di ambiente era legata al lavoro manuale enorme che veniva richiesto per l'annotazione dei file e alla conseguente perdita di tempo. La tecnica del Content Based Image Retrieval nasce proprio dalla necessità di eliminare tali limitazioni.

Uno dei primi sistemi CBIR è QBIC, query based image content [2], un progetto in cui il *contenuto* dell'immagine è usato come unico criterio per effettuare confronti invece dell'utilizzo di parole chiave o qualsiasi altro testo associato ad essa.

Questa tecnica, come suggerisce l'esperienza, fornisce più match di congruenza comparato con le vecchie metodologie basate su ricerche testuali.

L'obiettivo di tali sistemi (CBIR) è quindi descrivere, mediante modelli matematici ed implementazioni informatiche delle immagini il loro contenuto, cercando per quanto possibile di rispettare ed interpretare i principi visivi dell'occhio umano [3].

Come descriveremo nei capitoli successivi di questo elaborato, il concetto di '*contenuto*' può essere riferito a colore, tessitura, forma, o qualsiasi altra informazione che può essere ricavata da un'immagine digitale [2] [4].

1.4 Il Sistema CBIR

Possiamo dunque riassumere un'architettura CBIR avvalendoci della raffigurazione grafica in Figura 1 e descrivendo di seguito gli step caratteristici ed evidenziando in corsivo i passaggi rappresentati nel processo:

- **Il sistema** dispone di una *Image Collection* nella quale sono contenute le immagini appartenenti ad un determinato dataset, nella fase successiva di *Feature Extraction* attraverso l'applicazione di algoritmi matematici vengono estratte le caratteristiche memorizzate ed indicizzate all'interno di vettori multidimensionali che rappresenteranno le features delle immagini e che comporranno il Feature Database.
- **L'utente**, attraverso l'utilizzo di un'interfaccia grafica, potrà in ogni momento interrogare il sistema attraverso una *Query Image*. Presa l'immagine campione, il sistema la processa utilizzando la fase di *Feature Extraction*; dopo aver registrato le *Query Image Features* il sistema applicherà una misura di distanza confrontando il vettore delle features dell'immagine data dall'utente con tutti i vettori precedentemente registrati all'interno del *Feature Database*, questa fase, la *Similarity matching*, stabilisce quali immagini devono essere restituite all'utente come risultato della query iniziale. Tutte le immagini rilevanti costituiscono le/la *Retrieved Images*.

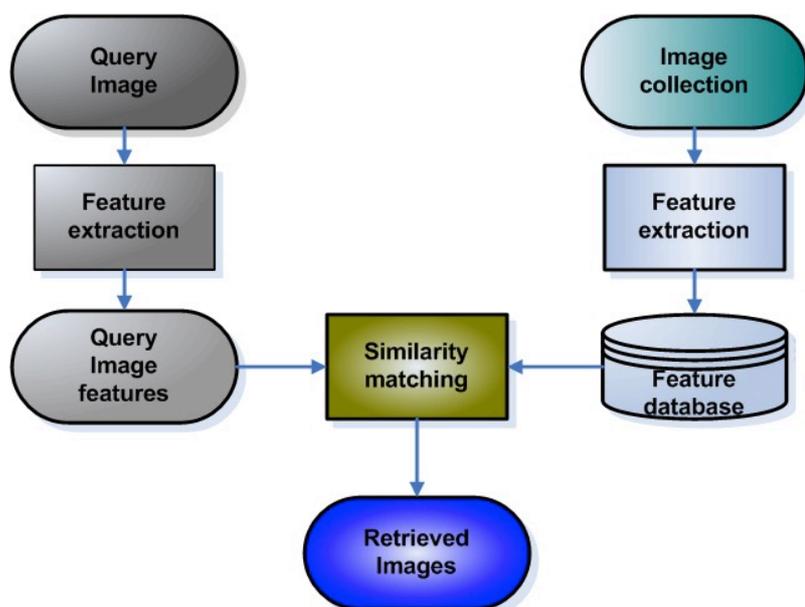


Figura 1: Modello generale di un Sistema CBIR.

1.4.1 CBIR e Relevance Feedback

La rappresentazione descritta nel paragrafo 1.4 rappresenta un sistema CBIR nella sua versione standard, è importante sottolineare che, nei sistemi più moderni è implementato il meccanismo del Relevance Feedback, oggetto di numerosi studi attraverso il quale il sistema ha la possibilità di interagire direttamente ed in tempo reale con l'utente, infatti servendosi del suo giudizio di valutazione è in grado di raffinare la query iniziale e migliorare l'insieme dei risultati finali.

L'articolo [5], descrive come l'uso del Relevance Feedback renda la ricerca di risultati un processo iterativo, dove ad ogni iterazione il sistema restituisce un numero di immagini in accordo con una misura di similarità prestabilita e richiede l'intervento dell'utente per catalogarle come rilevanti o meno. In questo modo si riesce ad adattare la misura di similarità così da ottenere in una seguente iterazione un nuovo insieme di risultati ripetendo poi il processo sino ad ottenere un unico risultato rilevante.

Eccetto la prima iterazione, il numero di risultati sarà quindi un parziale dell'insieme iniziale influenzato dal Relevance Feedback dell'iterazione precedente. Aumentando le ripetizioni si otterrà un numero di risultati sempre più prossimo a quello desiderato.

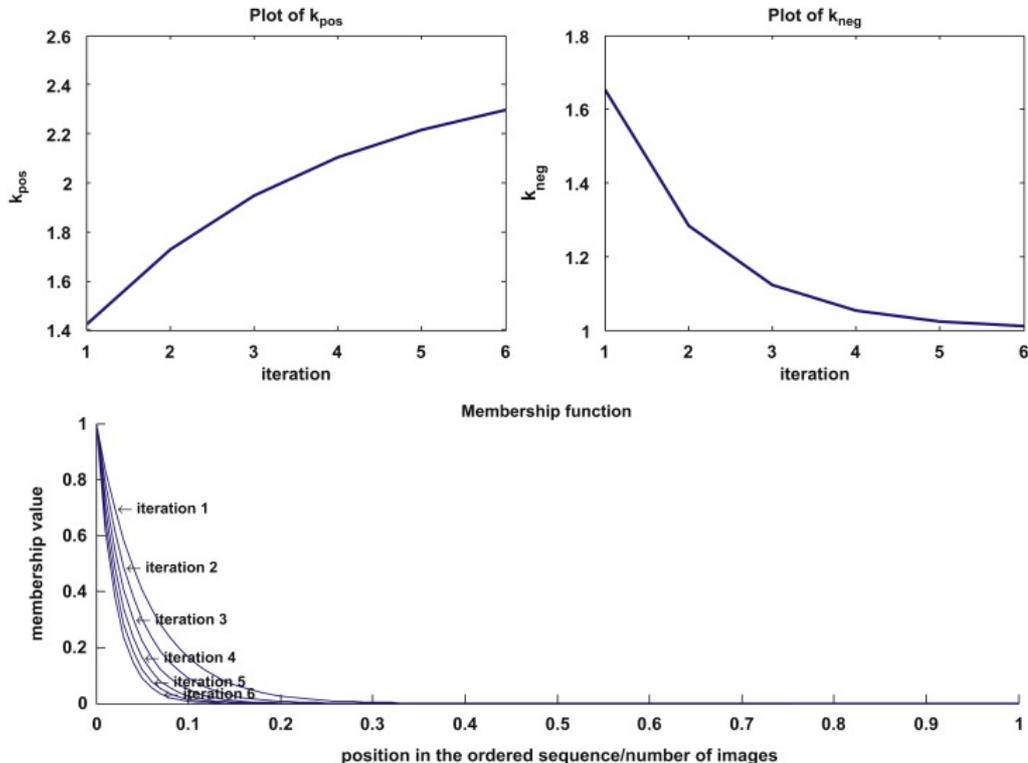


Figura 2: Comportamenti di un'istanza di CBIR in funzione delle iterazioni [5].

1.5 Panoramica sui sistemi CBIR esistenti

Tra gli ambiti di ricerca che utilizzano sistemi CBIR, uno dei più attivi è quello che riguarda i database di immagini digitali, che vengono ormai utilizzati in molti contesti differenti.

Tra le principali applicazioni descritte in letteratura abbiamo l'utilizzo di immagini e sistemi CBIR in ambiente medico, principalmente per lo studio di proteine, sequenze di DNA o supporti decisionali alle prognosi identificandone casi passati o simili (un esempio è riassunto in Figura 3), nel settore antropometrico e della sicurezza per il riconoscimento facciale, il matching di impronte digitali o la scansione della retina dell'occhio e sempre in ambito della sicurezza applicazioni per la telesorveglianza, per l'analisi automatica di immagini provenienti da telecamere.

Di fondamentale importanza sono le applicazioni GIS (Geographic Information System) basate sull'utilizzo e la gestione di immagini satellitari per scopi civili e militari.

Inoltre vanno ricordati ambiti applicativi come la realizzazione di sistemi di ricerca tramite immagine per il web, come quella fornita nel motore di ricerca di Google e infine anche realizzazione di applicazioni con scopi ludici o di apprendimento.

L'importanza dei metodi e delle tecniche CBIR consiste nel fatto che non escludono a priori l'utilizzo di altri metodi ma che possano cooperare in sinergia con altre forme tradizionali di information retrieval, favorendone le performance.

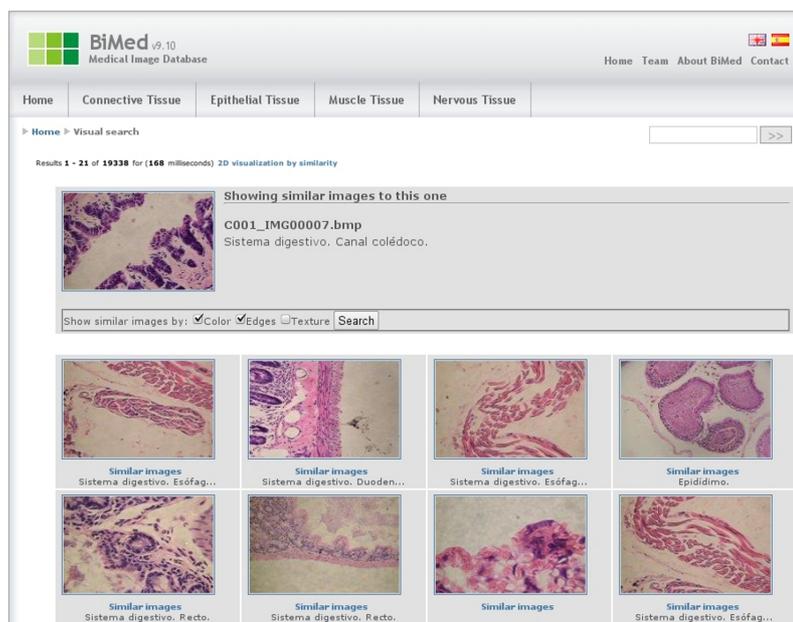


Figura 3: L'immagine riassume un'istanza di CBIR per il supporto decisionale in ambito di ricerca.

Negli ultimi anni sono stati progettati molti sistemi di Content Based Image Retrieval, alcuni molto semplici e basati solamente sul riconoscimento delle forme, altri più complessi che considerano molte se non tutte le tipologie di features e che possono prevedere la fase di Relevance Feedback.

Riportiamo alcuni tra i sistemi CBIR più noti in letteratura:

- **SQUID**, Shape Queries Image Database, è un sistema progettato dall'Università di Surrey, precisamente dal Center for Vision Speech and Signal Processing. Esso utilizza tecniche di indicizzazione e di ricerca basate esclusivamente sulle forme, infatti ogni immagine è elaborata per recuperarne il contorno che viene poi rappresentato da tre parametri di forma e dal valore massimo della curvatura nella sua immagine CSS (Curvature Scale Space). Il database impiegato in questo sistema è caratterizzato da 1100 immagini di creature marine.
- **MARS**, Multimedia Analysis and Retrieval System, è un sistema proposto dall'Università dell'Illinois, basato sul Relevance Feedback e implementato in C++.

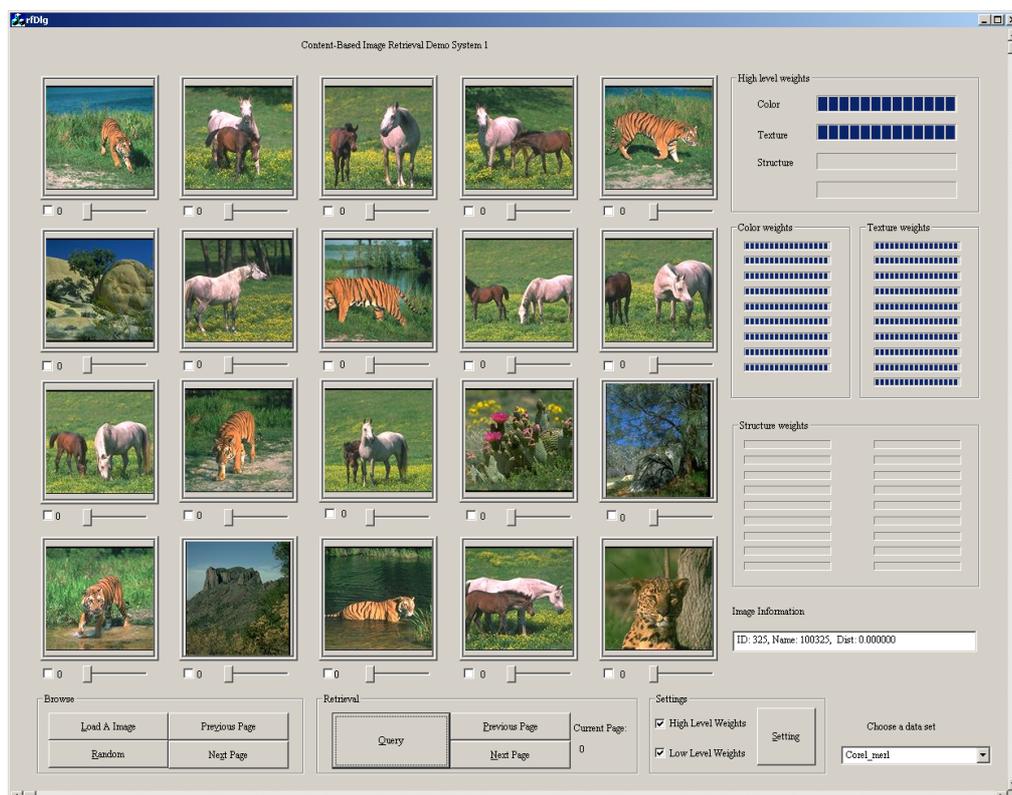


Figura 4: Schermata del sistema Mars.

- **CIRES**, Computer and vIision RESearch Center, è un sistema che si basa su query by example ed è stato proposto dall'Università del Texas.
- **QUICKLOOK**, è un sistema sviluppato in Italia dai ricercatori del Dipartimento di Informatica, Sistemistica e Comunicazione dell'Università di Milano e dall'Istituto per le Tecnologie della Costruzione del CNR di Milano. Viene applicato ad immagini fisse ma anche a video, è in grado di combinare interrogazioni basate sul contenuto visivo estratto automaticamente con interrogazioni basate sulla similarità testuale tra le descrizioni che sono associate agli elementi. Il sistema permette inoltre di raffinare progressivamente la risposta sulla base di un Relevance Feedback che l'utente fornisce sugli elementi visivi reperiti. Questa continua possibilità di interagire permette al sistema di sfruttare i parametri inseriti runtime dall'utente per aggiustare la mira della ricerca e producendo nuove informazioni e indicazioni sulla rilevanza dei documenti recuperati.
- **Google Image Search**, è il sistema introdotto nel motore di ricerca di Google per la ricerca tramite immagine.



Figura 5: Schermata di Google Image Search.

- **BRISC**, è un sistema provvisto di un framework per l'estrazione di features di tessitura e la comparazione di immagini mediche di tomografie. Il sistema è stato progettato per essere funzionale ed estendibile.

- **WindSurf**, Wavelet-based INDeXing of ImageS Using Region Fragmentation, è un sistema sviluppato dall'Università di Bologna. Il framework permette all'utente la selezione i metodi di estrazione delle features, di segmentazione dell'immagine e di comparazione.

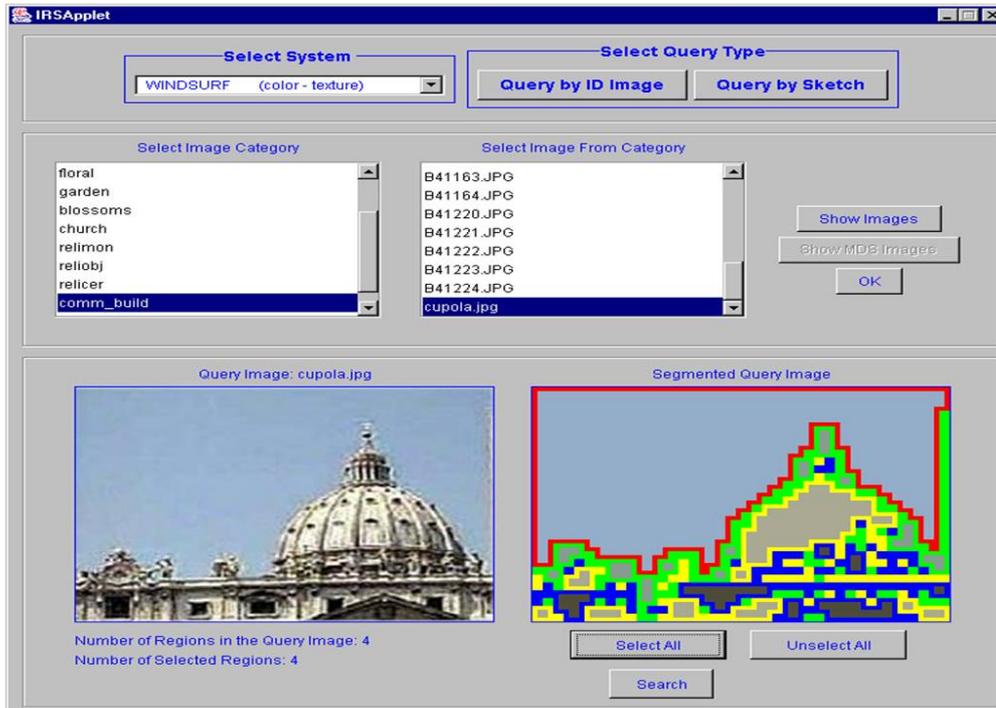


Figura 6: Schermata del sistema Windsurf.

Capitolo 2

Descrittori e Misure di Similarità

2.1 Introduzione ai metodi

In questo capitolo vengono introdotti i concetti di descrittore di una immagine digitale e quello di misura di similarità, elementi fondamentali nel campo CBIR senza i quali sarebbe impossibile realizzare una qualsiasi comparazione tra immagini.

I diversi descrittori possono essere classificati a seconda delle proprietà dell'immagine che intendono evidenziare: descrittori che si basano sul colore, tessitura, di forma ed altri ancora; possono essere anche utilizzati descrittori ibridi, unendo le capacità dei singoli.

La rappresentazione avviene attraverso un vettore di valori che contiene le informazioni (misure) estratte dall'immagine.

È intuibile che un semplice descrittore risulta fine a se stesso senza l'utilizzo di una funzione matematica apposita in grado di confrontare questi vettori e interpretarne il grado di similarità. Queste funzioni, chiamate misure di similarità, verranno raccolte all'interno del capitolo per ognuna verrà fornita l'espressione matematica che la caratterizza.

Alcuni lavori come [6] preferiscono definire un descrittore come l'unione del vettore delle caratteristiche e la funzione di similarità necessaria alla comparazione; come riassume l'esempio grafico mostrato in Figura 7:

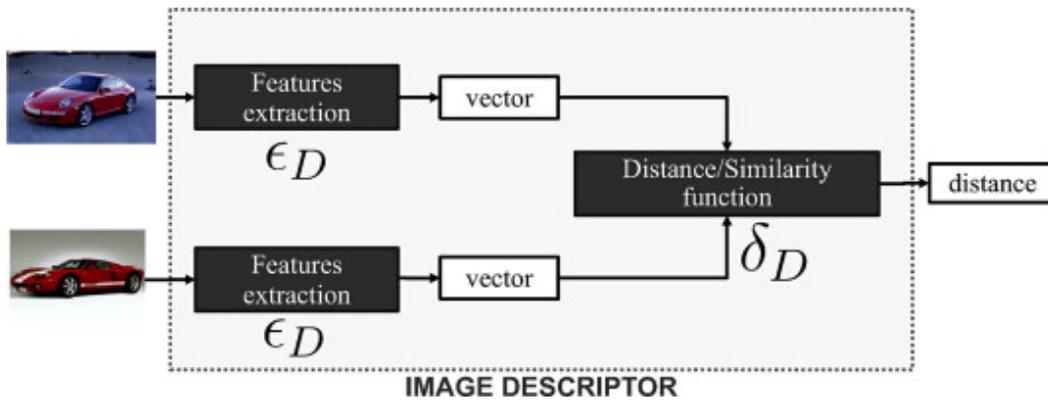


Figura 7: Definizione di descrittore in [6].

In questo lavoro di tesi, invece, il termine descrittore sarà associato al solo vettore di caratteristiche (feature) e saranno elencate varie misure di similarità utilizzabili per ottenere i confronti.

La letteratura offre numerose tecniche e misure per il confronto di immagini, ma la scelta del metodo migliore non è mai semplice, questo perchè le prestazioni che un sistema è in grado di offrire dipende dal problema al quale viene applicato. Ad esempio, se ci troviamo nel contesto del confronto di immagini satellitari, i migliori descrittori risultano quelli della tessitura, al contrario i descrittori del colore sono molto più performanti in problematiche rivolte al confronto di fotografie di paesaggi.

Come affrontato negli studi [7] [8], ed in tanti altri presenti in letteratura, quello che ci accingiamo ad eseguire, è analizzare il problema valutando e interpretando i risultati delle performance e dell'efficienza in modo da determinare quali descrittori, risultano maggiormente utili alle diverse casistiche di problemi CBIR.

Estraendo informazioni dal contenuto di un'immagine, possiamo individuare due metodologie con caratteristiche e scopi differenti. Un **approccio discreto**, descrivibile come un attributo binario, in grado di indicare la presenza o meno di una determinata caratteristica all'interno dell'immagine; questa tipologia risulta poco diffusa a differenza del più noto ed utilizzato **approccio continuo**, caratterizzato dalla descrizione dell'immagine in forma vettoriale che vede poi l'applicazione delle misure di distanza.

Ovviamente le immagini che risulteranno avere una distanza tra i vettori minore, avranno un grado di similarità maggiore.

Continuando ad introdurre l'approccio continuo, possiamo ulteriormente dividere i metodi appartenenti a questa categoria in base a:

- **Colore**, indubbiamente la più importante qualità di un contenuto visivo.
- **Tessitura**, altra importante caratteristica di un'immagine è la descrizione della tessitura dell'intero contenuto o solo di determinate regioni.
- **Forma**, si basano sull'acquisizione di informazione relative alla geometria dei contorni di oggetti simulando l'abilità umana di riconoscere oggetti interpretandone la forma.
- **Informazioni Locali**, ottengono informazioni locali individuando punti salienti all'interno del contenuto.
- **Motion** (non saranno trattati), utilizzo di quattro diversi descrittori che caratterizzano il movimento in una sequenza video.

L'associazione di un descrittore e di una buona misura di similarità saranno il garante delle prestazioni e dell'efficienza di un sistema Content Based Image Retrieval a seconda del problema che viene analizzato.

2.2 Descrittori del colore

Una delle più importanti proprietà della vista umana è il colore, questo fa sì che descrittori basati sulle feature dei colori, in particolare gli istogrammi, rappresentino uno dei principali approcci ampiamente utilizzati nell'ambito del recupero delle immagini [9] [10] [11] [12] [13].

In linea teorica, la letteratura fornisce tre principi per analizzare i descrittori che si basano su questo genere di proprietà: l'Approccio Globale, che si basa sull'estrazione del descrittore dall'intera immagine, molti di questi generano istogrammi come rappresentazione del vettore delle feature come ad esempio il *global color histogram* [13]. L'Approccio Locale, basato sulla divisione dell'immagine in regioni di dimensione regolare e ne estrae informazioni da ognuna di esse, ad esempio il *local color histogram* [13], infine l'Approccio Locale a Segmentazione che suddivide l'immagine in un numero non prestabilito di celle che può variare da un'immagine all'altra. Quest'ultimo metodo risulta sperimentalmente il più complesso ma il più efficace. In Figura 8 è mostrata la gerarchia della tassonomia dei descrittori del colore.

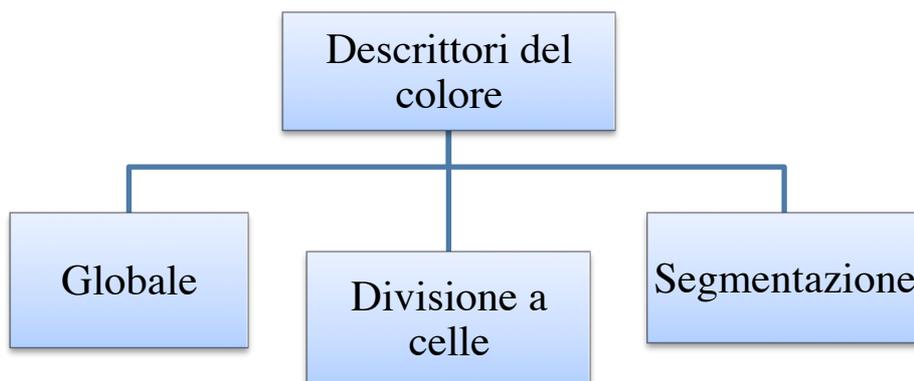


Figura 8: Tassonomia dei descrittori di colore.

2.2.1 Istogramma RGB e HSV

L'idea basilare per la realizzazione di un istogramma, è quella per cui ogni pixel dell'immagine viene esaminato e ne viene fatto corrispondere un valore che identifica le quantità di colori primari che contiene, successivamente si ottiene un istogramma facendo corrispondere ad ogni valore, un range prestabilito nello spazio dei colori. Utilizzando lo spazio dei colori RGB, si ottengono tre istogrammi, uno per ogni canale: rosso, verde, e blu.

È possibile variare il range di ogni canale per modificare la sensibilità in fase di analisi: ad esempio volendo rappresentare l'immagine con un vettore di 24 elementi si possono assegnare 8 bucket per ogni canale in modo da attribuire un'importanza equivalente ai tre colori.

Solitamente per l'identificazione viene utilizzato il sistema RGB. In [14] è stato osservato che utilizzando altri spazi di colore, non si ottengono differenze significative.

Un altro spazio molto utilizzato è l' HSV, che racchiude invece informazioni relative alle grandezze percepibili all'occhio umano e comuni a tutti i colori: la tinta (Hue), rappresentante la lunghezza d'onda del colore, la saturazione (Saturation), indicante la quantità di colore e la brillantezza (Brightness) che ne descrive la luminosità; anche in questo caso ogni caratteristica sarà rappresentata da un istogramma, ma a differenza di quello precedente, è possibile assegnare un numero maggiore di bucket all'istogramma della tinta in modo che in fase di analisi acquisisca maggiore rilevanza (a discapito della luminosità che risente delle variazioni di luce).

In rete è possibile anche rintracciare applicazioni in grado di restituire in tempo reale le diverse tipologie di istogrammi che racchiudono le informazioni relative al colore.

Riporto il link ad una applicazione rintracciata sul web molto interessante e semplice per interpretare i concetti:

<http://www.senchalabs.org/philogl/PhiloGL/examples/histogram/>.

In Figura 9 e in Figura 10 vengono mostrate le rappresentazioni rispettivamente degli spazi di colore RGB e HSV.

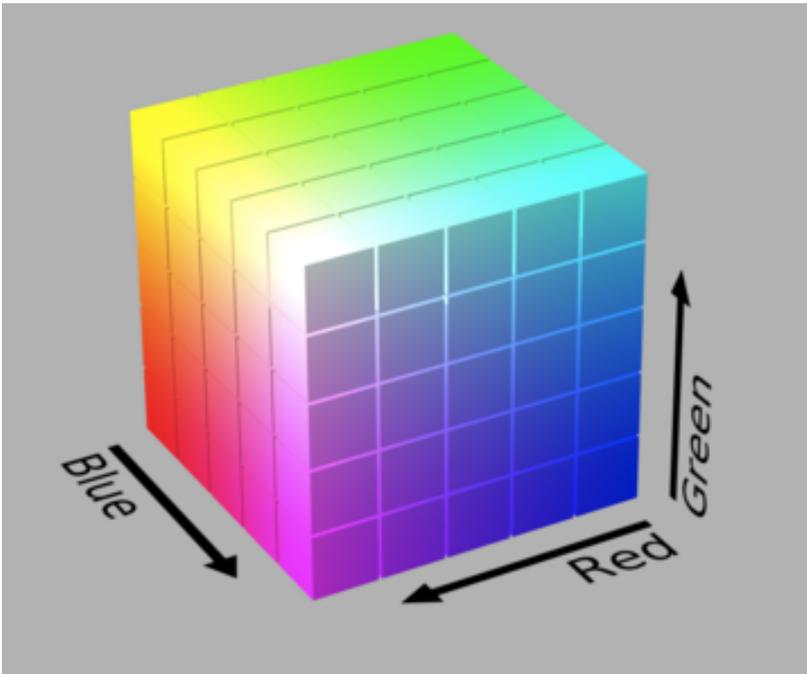


Figura 9: Rappresentazione grafica in 3D dello spazio dei colori RGB.

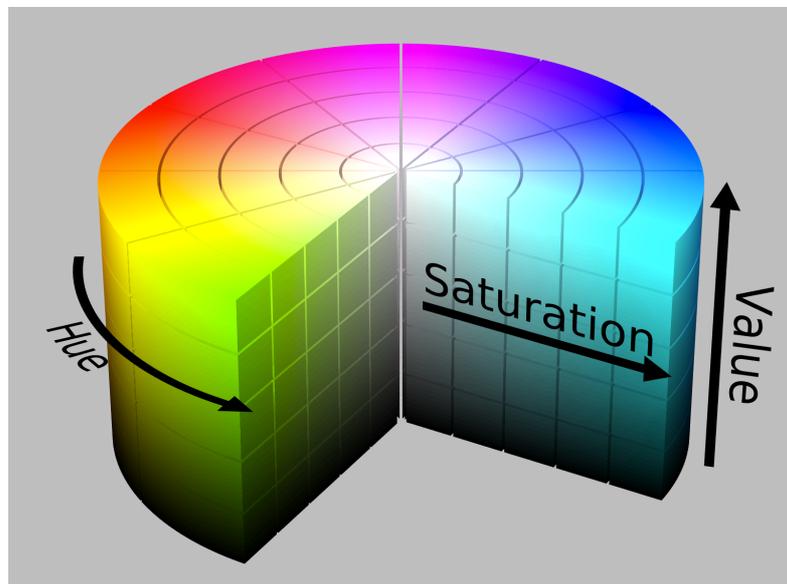


Figura 10: Rappresentazione grafica in 3D dello spazio dei colori HSV.

Di seguito sono riportati esempi di immagini con i relativi istogrammi RGB e/o HSV.

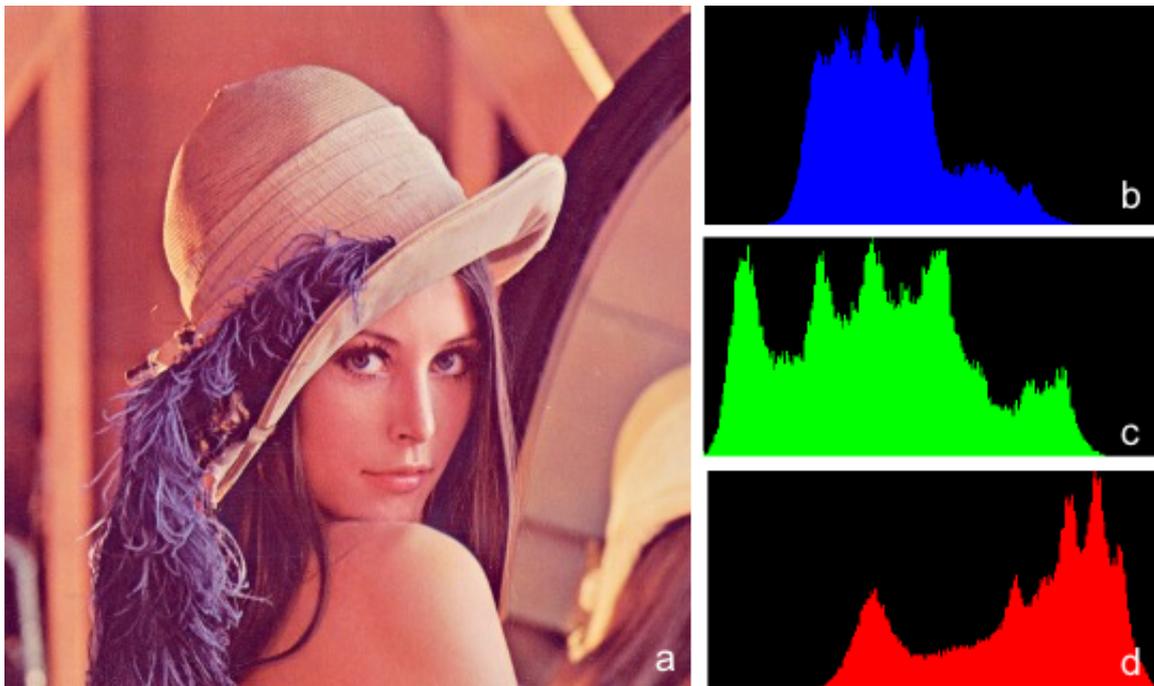


Figura 11: In figura è possibile vedere l'immagine di riferimento(a), dalla quale viene estratto l'istogramma del blu(b), del verde(c) e del rosso(d). Il colore di ogni pixel è dato dall'insieme dei singoli valori dei tre colori.

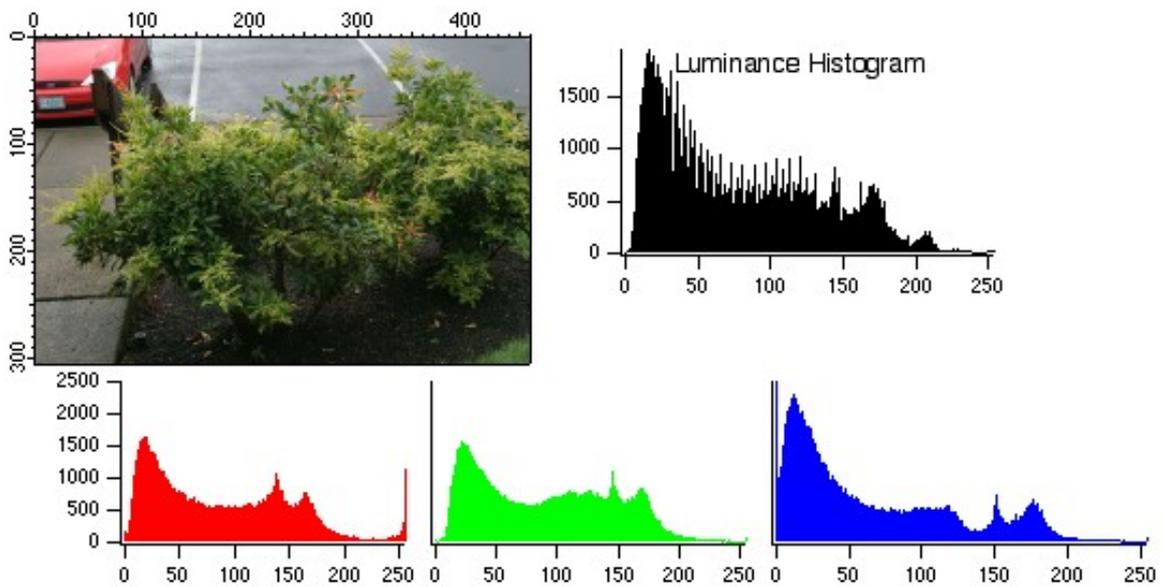


Figura 12: Altra immagine della quale sono riportati gli istogrammi RGB, inoltre in nero è riportato l'istogramma relativo alla luminosità dell'immagine.

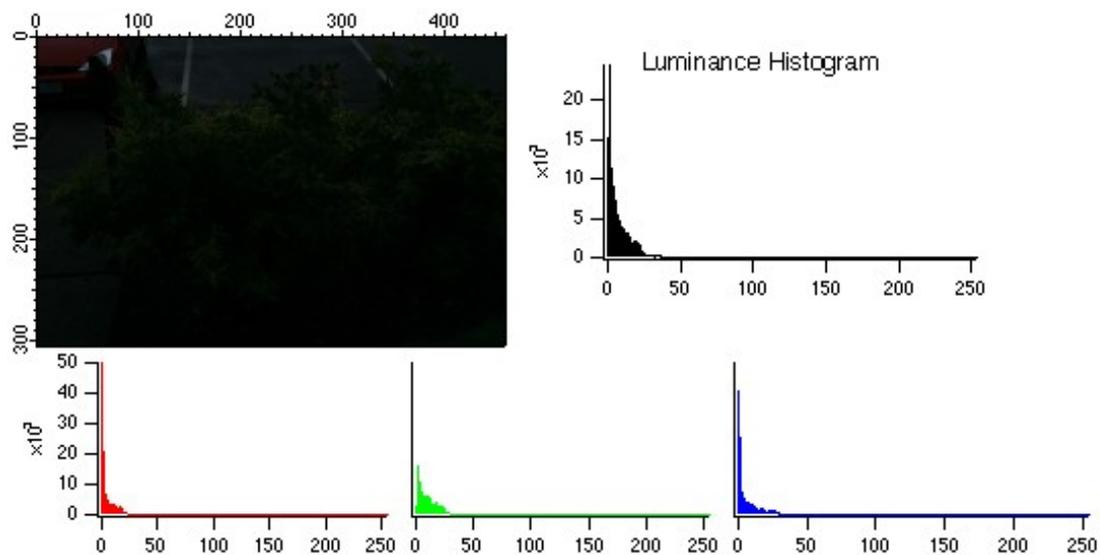


Figura 13: Immagine di Figura 12 riportata con caratteristiche di scarsissima luminosità, con i relativi istogrammi delle feature.

Osservando le immagini di Figura 13 e Figura 14 si può effettuare un confronto visivo per avere conferma del rapporto tra le caratteristiche riportate dagli istogrammi, infatti un'immagine scura restituisce valori di luminosità bassissimi, al contrario un immagine luminosa è caratterizzata da valori molto alti in ogni pixel.

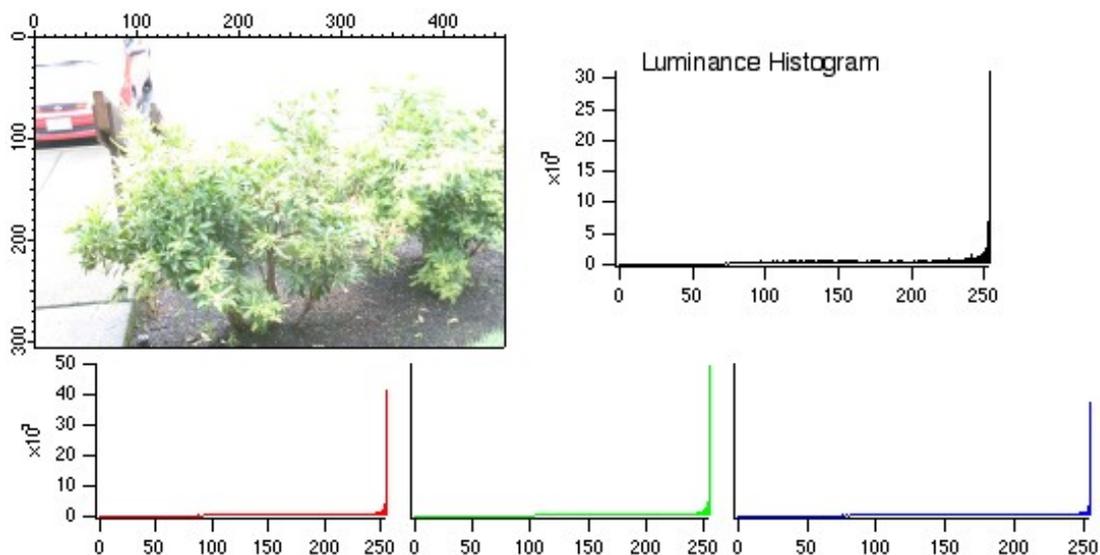


Figura 14: immagine di Figura 12 riportata con caratteristiche di altissima luminosità, con i relativi istogrammi delle feature.

L'immagine in Figura 15 mostra invece sia l'estrazione di feature di colore nello spazio RGB che in quello HSV, mentre in Figura 16 abbiamo l'estrazione dei parametri HSV tramite Matlab.

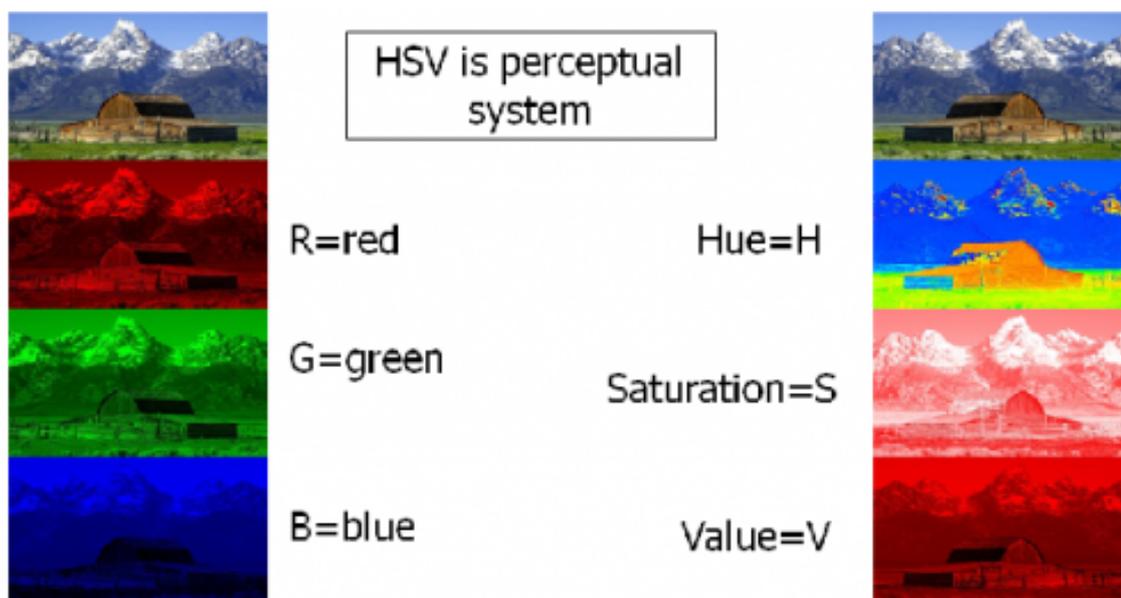


Figura 15: RGB e HSV della stessa immagine iniziale.

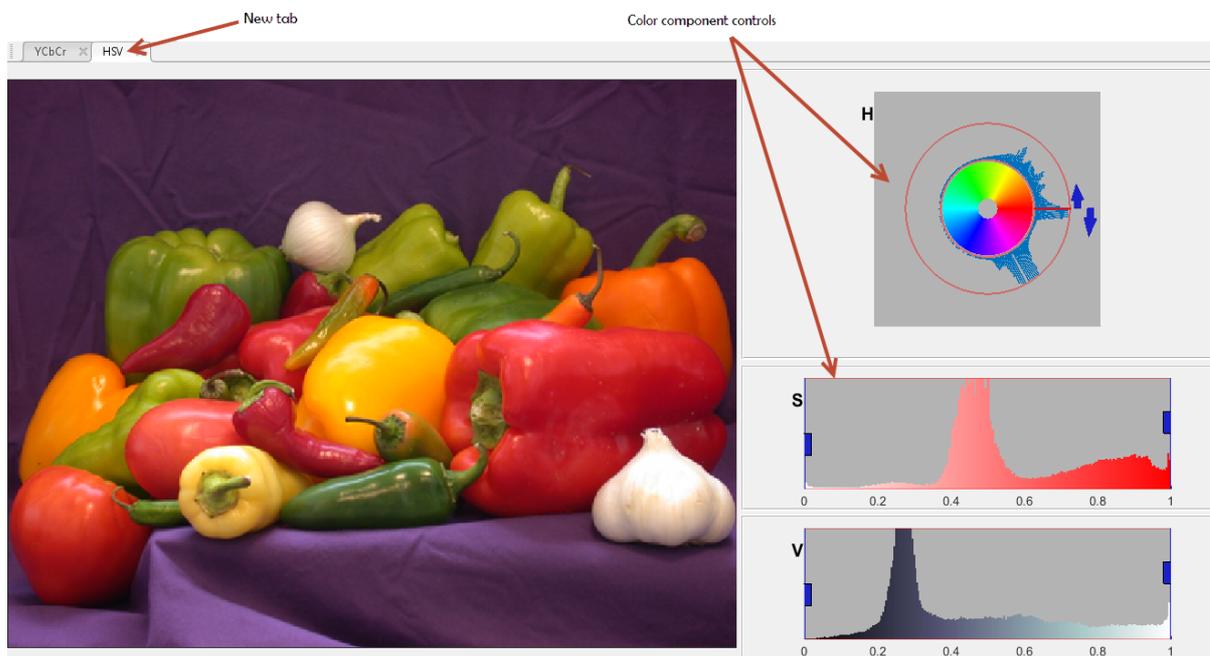


Figura 16: Istogrammi dei parametri HSV.

2.2.2 Color Coherence Vector

Intuitivamente, definiamo un'area con colore coerente come uno spazio nel quale l'insieme dei pixel è caratterizzato dallo stesso grado di colore. Definire tale spazio è molto importante perché permette di caratterizzare un'immagine. Il concetto è facilmente interpretabile osservando la Figura 17.



Figura 17: Rappresentazione di due immagini caratterizzate dalla stessa distribuzione di colore.

Infatti, se si provasse ad ottenere gli istogrammi dei colori descritti nel paragrafo precedente si avrebbero risultati simili nonostante la loro chiara diversa apparenza. Questo suggerisce che il colore rosso (ad esempio) appare in entrambe i casi approssimativamente nella stessa quantità. Nell'immagine di sinistra, il rosso è distribuito in maniera irregolare prevalentemente nell'area dei fiori, mentre l'immagine di destra presenta una regione di pixel rossi ben definita, coerente. L'idea di base di questo descrittore di colore è quindi quella di classificare ogni pixel come coerente o incoerente.

I pixel coerenti sono quelli appartenenti ad una regione misurabile e delimitata dell'immagine, al contrario non lo sono i pixel incoerenti. Un CCV è quindi un vettore che rappresenta questa classificazione per ogni colore presente nell'immagine in esame.

Questo metodo permette di evitare il matching tra pixel che in un'immagine si presentano come coerenti, mentre sono incoerenti nell'altra oggetto del confronto. La distinzione appena descritta, non può essere effettuata utilizzando istogrammi del colore [15].

2.3 Descrittori di Forma

Parliamo ora dei descrittori di Forma per un'immagine; per introdurli ipotizziamo di estrapolare all'interno di una foto, un elemento presente in essa come un monumento, una macchina, un albero o un qualsiasi altro oggetto che possa essere identificato come singola entità. Ovviamente il riconoscimento di oggetti e il CBIR sono campi strettamente correlati [16] [17] e per alcuni processi di recupero di immagini, partire dal metodo di riconoscimento di un oggetto potrebbe essere l'unica soluzione possibile. Per esempio ricercando immagini ritraenti una stessa persona, chiaramente un metodo di riconoscimento esclusivamente facciale ci darà il risultato migliore [18] [19].

2.3.1 Istogramma dei gradienti orientati

Il pensiero fondamentale dietro l'istogramma dei descrittori dei gradienti orientati è quello di ottenere una descrizione di una forma all'interno dell'immagine attraverso la distribuzione dei gradienti di intensità o di direzione. L'attuazione di questo descrittore può essere ottenuto dividendo l'immagine di partenza in piccole regioni di dimensioni prestabilite dette 'celle', e per ognuna delle celle computare un istogramma di direzioni. La combinazione di tutti i singoli istogrammi della cella costituisce il descrittore complessivo. Per aumentarne le prestazioni è consigliabile aumentare il contrasto dell'immagine in esame, operazione denominata *contrast stretching*. Il descrittore HOG mantiene alcuni vantaggi chiave rispetto ad altri metodi di descrizione di forma. Dal momento che esso opera su cellule localizzate, il metodo sostiene invarianza per trasformazioni geometriche e fotometriche.

In figura 18 e 19 le immagini mostrano i passaggi per ottenere l'istogramma dei gradienti e la visualizzazione di un'immagine attraverso l'orientamento dei gradienti.

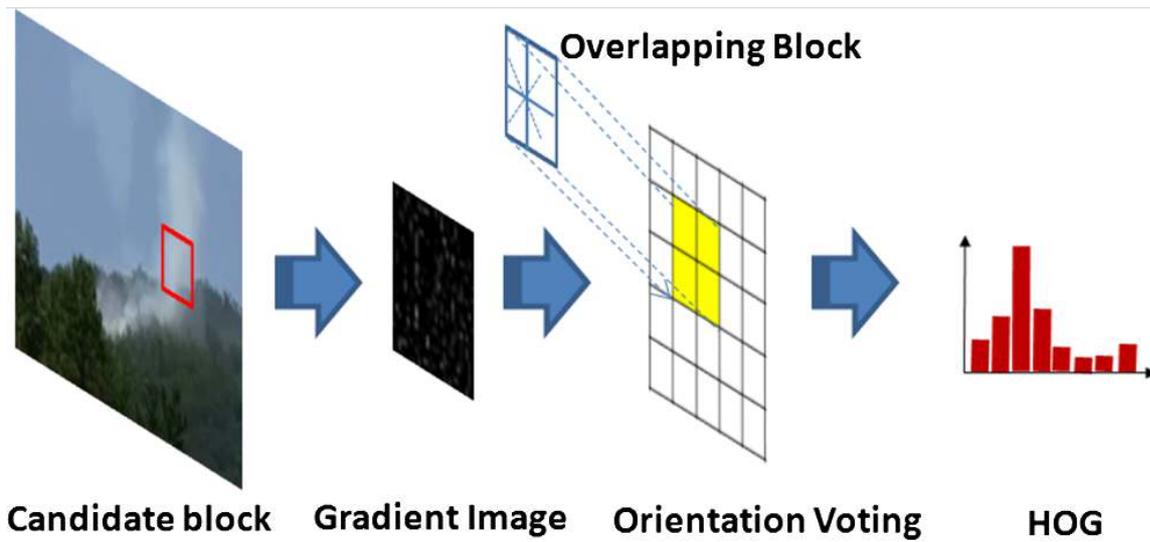


Figura 18: Rappresentazione grafica del procedimento effettuato per l'estrazione dell'istogramma delle direzioni da una cella.

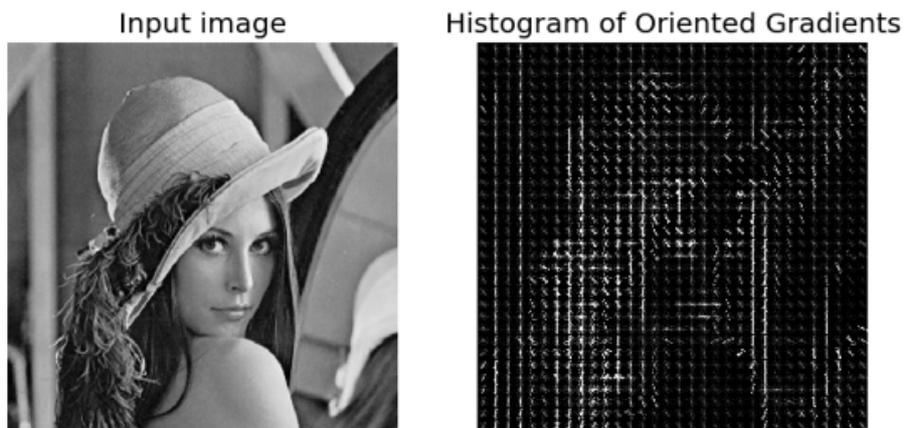


Figura 19: Esempio di come un'immagine viene rappresentata dalle direzioni delle sue celle con l'applicazione del descrittore HOG.

2.4 Descrittori di Tessitura

Parlando di descrittori di tessitura di un'immagine, risulta molto difficile dare una definizione formale per essi. Molti articoli in letteratura forniscono diverse definizioni, prendiamo ad esempio la definizione data da [20]: una tessitura può essere descritta dalle proprietà di percezione, di frequenza e da quelle spaziali.

In un certo senso la tessitura può essere interpretata come un insieme di variazioni di intensità dovuti alla ripetizione di determinate regole (pattern).

Differentemente dai descrittori di colore, le proprietà di tessitura non possono essere sempre estratte considerando i valori di un singolo pixel, infatti per ottenere informazioni valide è spesso necessario calcolare le variazioni in un contorno di pixel.

Molti descrittori di tessitura lavorano utilizzando scale di grigi e solamente pochi di essi hanno una descrizione intrinseca su come convertire l'immagine a colori per ottimizzare le performance.

Analizzando la tassonomia possiamo anche in questo caso dividere i descrittori di tessitura in più classi, la Figura 14 rappresenta la suddivisione data anche da [21]. Uno dei metodi più tradizionali per analizzare la distribuzione spaziale dei livelli di grigio è quello statistico, ad esempio computando la probabilità di co-occorrenza di valori di grigio in differenti distanze e orientamenti. Il metodo statistico può calcolare i valori dei singoli pixel (metodo statistico del primo ordine) oppure su coppie di pixel (metodo statistico del secondo ordine) [21].

Il metodo Geometrico effettua un'analisi avvalendosi delle proprietà geometriche delle primitive, come scala, forma, area o lunghezza e dopo averle identificate, ne estrae le regole utilizzando forme vettoriali. Questo metodo a volte risulta molto difficile nella descrizione di elementi naturali come un cielo con le nuvole, che possono avere caratteristiche simili ma le primitive come la forma o la dimensione sono molto varie; mentre risulta molto buona l'applicazione ad elementi come un muro di mattoni, dove gli elementi riportano primitive quasi costanti.

Un altro metodo è il Model-Based che raccoglie informazioni sintetiche della tessitura ed è dimostrato che lavora molto bene quando applicato ad immagini con una tessitura molto regolare. Infine il metodo Signal Processing caratterizza le informazioni di tessitura applicando dei filtri sopra all'immagine, i descrittori che sfruttano ad esempio i filtri di Gabor, seguono questo approccio.

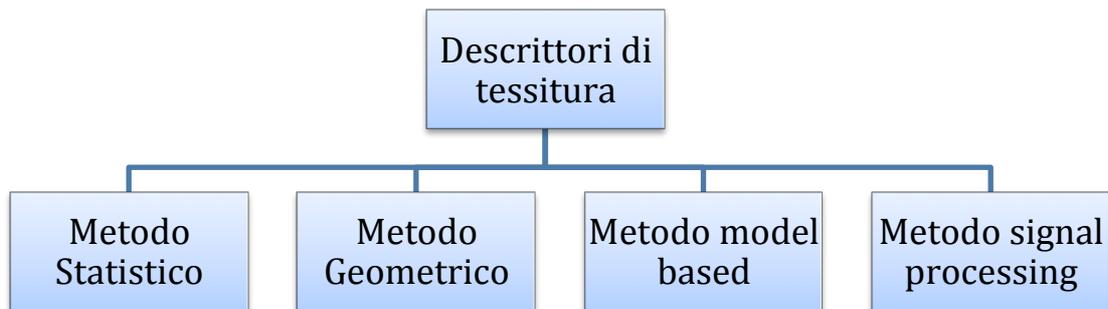


Figura 20: Tassonomia per i descrittori di tessitura.

2.4.1 Descrittori Globali di tessitura

In [11] è stato delineato un insieme composto da diversi attributi di tessitura dai quali è possibile ottenere un ottimo descrittore di tessitura: *Fractal Dimension*, misura la rugosità di una superficie, è calcolato utilizzando il metodo reticular cell counting; *Coarseness*, caratterizza la grossezza della grana dell'immagine, è calcolata utilizzando la varianza dell'immagine; *Entropy* dei pixel, che è usata come misura del disordine all'interno dell'immagine; *Spazial gray-level difference statistics*, descrive la relazione di luminosità tra i pixel all'interno dei bordi, è anche noto come analisi della matrice di co-occorrenza [22]. Il *circular Moran autocorrelation function*, misura la grossolanità della tessitura. Per il calcolo totale del descrittore viene utilizzato un set di funzioni di autocorrelazione [23].

Dalle caratteristiche sopra elencate e dalle funzioni si ottiene un vettore a 43 dimensioni, dove si ha un valore per la Fractal Dimension, uno per la Coarseness, uno per l'Entropy, 32 valori per la difference statistics e 8 per la Circular Moran autocorrelation function.

Questo descrittore è stato utilizzato con ottimi risultati per database contenenti immagini mediche [24].

2.4.1.1 Tamura Features

Nell'articolo [25] l'autore propone una struttura composta da sei fondamentali features di tessitura che riassumono la percezione visiva umana: *roughness contrast*, *directionality*, *line-likeness*, *regularity*, e *coarseness*. Test effettuati utilizzando questi descrittori hanno evidenziato che i primi tre elementi rappresentano i più importanti e questo porta ad utilizzarli per i confronti istogrammi composti da coarseness, contrast, directionality (grossolanità, contrasto, direzionalità) [11]. Questo genere di istogrammi viene molto utilizzato anche nei sistemi QBIC [26].

2.4.1.2 Average Hash

Questo descrittore estrae le features da un'immagine in bassa frequenza. Per farlo l'algoritmo deve rimuovere le alte frequenze e i dettagli dall'immagine iniziale per poi ridurre la risoluzione iniziale a un semplice quadrato di 64 pixel, ossia di dimensioni 8x8.

L'immagine ridimensionata e privata delle qualità dell'alta frequenza viene poi convertita dalla rappresentazione a colori nella rappresentazione in scala di grigi passando quindi da una rappresentazione realizzata da 64 pixel di rosso, 64 di verde e 64 di blu ad una molto più essenziale composta da soli 64 pixel totali di colore. Da questi ultimi si effettua la media dei singoli valori per poi decidere nel passaggio successivo se il valore di ogni singolo pixel è maggiore o minore della media precedentemente ottenuta.

In questo modo si ottiene una stringa di 64 bit che viene convertita in un intero a 64 bit che in definitiva rappresenta l'identificatore (il descrittore) del metodo Average Hash.

2.4.1.3 Perceptive Hash

Il Perceptive Hash utilizza un algoritmo più robusto del'Average Hash, estendendo il metodo precedente e introducendo l'utilizzo di un filtro DCT (discrete cosine transform) per ridurre la frequenza dell'immagine.

La risoluzione viene portata in questo caso a 32x32 in modo da poter utilizzare il filtro DCT applicato dopo che l'immagine è stata portata in scala di grigio. Dopo la computazione viene selezionato il quadrato 8x8 con frequenza minore per poi da esso eseguire il metodo della media e ricavarne l'intero a 64 bit come nel metodo visto precedente.

2.4.1.4 Different Hash

Osservando l'implementazione dell'algoritmo Different Hash si nota che la principale differenza rispetto ai metodi simili è l'utilizzo di una risoluzione per ridurre le alte frequenze di 9×8 , dove quindi il numero complessivo di pixel sotto esame sono 72 e non 64. Come visto in precedenza l'immagine viene convertita in una scala di grigio, in seguito viene computata la *differenza* comparando per ogni pixel quelli adiacenti e grazie all'utilizzo di una risoluzione 9×8 in definitiva si ottengono 64 comparati poiché il primo di ogni riga viene utilizzato solo come comparatore e non riceve l'assegnazione del bit. Si sottolinea che in questo caso la comparazione avviene osservando la luminosità, infine come nei due metodi precedenti si ottiene un intero a 64 bit che identificherà l'immagine.

2.4.1.5 Haralick Texture Features

Queste features sono utilizzate per la classificazione di un'immagine catturando informazioni salienti che emergono dai patterns della tessitura, ossia ripetizioni di regole che stabiliscono uno schema ripetitivo di una caratteristica. Questo tipo di caratteristiche sono calcolate utilizzando la matrice di co-occorrenza, metodo alquanto costoso sotto il profilo computazionale. Alla fine della computazione si ottiene una matrice nella quale sono contenute le 13 Haralick features, di seguito elencate [27]: *Energy, Correlation, Inertia, Entropy, Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Difference Average, Difference Variance, Difference Entropy, Information measure of correlation 1, Information measure of correlation 2*. Per un approfondimento delle singole feature si rimanda il lettore all'articolo originale [27].

2.4.2 Descrittori Locali di Tessitura

Un descrittore locale di tessitura a differenza del globale estrae le informazioni su un ‘intorno locale’ dei pixel dal quale vengono estratte informazioni che alla fine della computazione vengono unite a formare il descrittore dell’immagine.

2.4.2.1 Filtri di Gabor

I filtri di Gabor sono oggetti matematici caratterizzati da una funzione armonica moltiplicata per una funzione Gaussiana; essi trovano grandissima applicazione nel campo dell’elaborazione delle immagini e della biometria, soprattutto per problemi come il riconoscimento dell’iride e delle impronte digitali

Attraverso l’uso dei filtri di Gabor è possibile realizzare nuovi attributi e descrittori ampiamente utilizzati in ambito di analisi della tessitura [28] [29].

Uno dei principali impieghi prevede l’uso di dodici filtri simmetrici e circolari differenti ai quali è possibile applicare un’intensità variabile su una scala da zero a dieci, l’esecuzione restituisce un istogramma che contiene la media di ogni singolo filtro che caratterizza le informazioni di tessitura delle immagini. Questi istogrammi sono comparati solitamente utilizzando la JSD della quale si fornisce la formula matematica nel paragrafo relativo alle misure di similarità.

Una rappresentazione grafica di come è costituito uno di questi filtri è fornita in Figura 21.

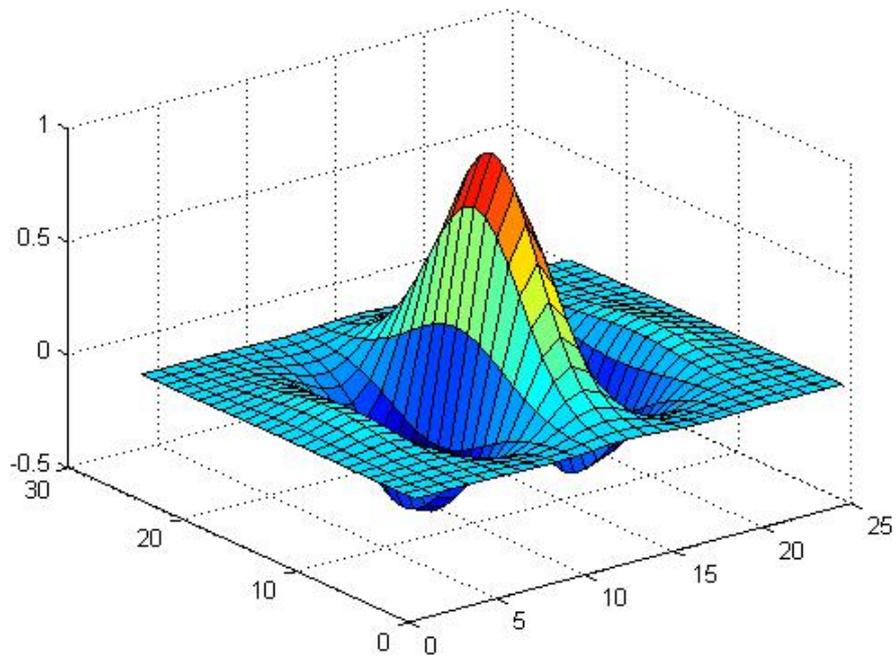


Figura 21: Filtro di Gabor.

Nell'esempio seguente vengono mostrati esempi dell'applicazione dei filtri di Gabor a differenti frequenze, potendo osservarne i comportamenti diversi sull'immagine iniziale.

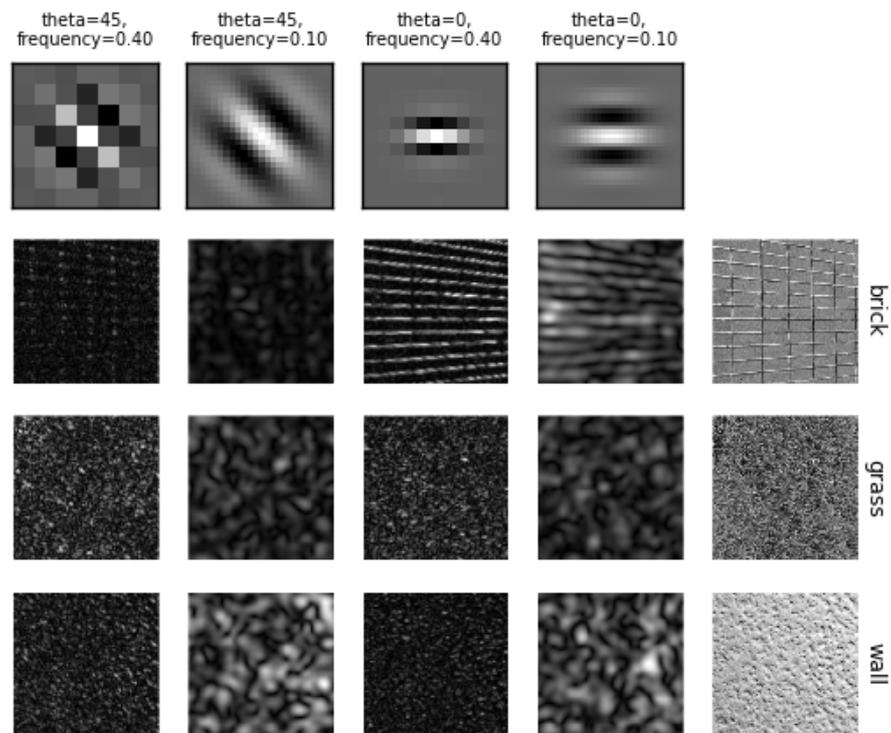


Figura 22: Diverso comportamento all'applicazione dei filtri di Gabor.

2.4.2.2 LBP

Il descrittore LBP (Local Binary Patterns) È un descrittore Model-Based la cui implementazione fa sì che un'immagine venga divisa in una griglia di celle composte da 9 pixel ed ogni pixel al centro della cella viene comparato con gli otto pixel che lo circondano, ai pixel che hanno valore maggiore rispetto al pixel centrale viene assegnato il valore binario '1' mentre per quelli il cui valore è minore rispetto al pixel centrale gli si assegna '0'. Otteniamo così una stringa binaria di otto valori per ogni cella, che verrà convertita in decimali e sarà fatta la media rispetto a quanti uni ci sono nella cella, Infine con i valori di tutte le celle che appartengono all'immagine si ottiene un istogramma completo.

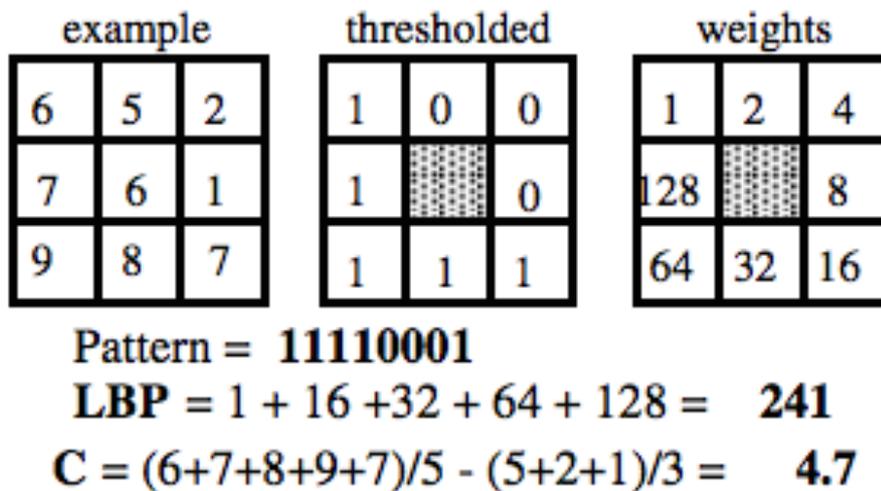


Figura 23: Procedimento del descrittore LBP su un pixel.

2.4.2.3 LPQ

Il descrittore LPQ (Local phase quantization), è stato originariamente proposto da Ojansivu e Heikkila come un metodo di estrazione di features di tessitura [30]. Il metodo si basa sull'invarianza di proprietà ottenute nello spettro di fase di Fourier. Utilizza le informazioni locali estratte con la trasformata di Fourier (operazione di convoluzione) computando su un contorno di pixel generalmente di forma rettangolare, su ogni pixel dell'immagine. Questo descrittore risulta particolarmente utilizzato in problemi relativi all'ambito medico.

In LPQ sono considerati solo quattro coefficienti corrispondenti alle frequenze 2-D.

2.5 Descrittori su Key Point (punti salienti)

Questo genere di descrittori, in una prima fase, si occupano della localizzazione dei Key Point (punti salienti), in particolare le regioni con un basso contrasto vengono scartate mentre al contrario quelle con orientamento del gradiente dominante vengono catalogate come Key Point. Scelti tali punti viene considerato un raggio di pixel dal quale verranno estratti i vettori di feature. Successivamente questi vettori vengono confrontati con quelli estratti dalla seconda immagine per testare quali fanno match (operazione di matching). Solitamente, la distanza tra i vettori viene calcolata utilizzando la distanza euclidea, viene poi scelto un valore di soglia e gli esperimenti realizzati da Lowe evidenziano che prendendo una soglia maggiore di 0,8 si possono eliminare il 90% dei falsi match perdendo soltanto il 5% di match corretti [31].

2.5.1 SIFT

Un esempio di descrittore locale molto importante poiché è utilizzato in gran parte delle sperimentazioni moderne e che tra le caratteristiche di forza ha anche quella di essere un descrittore invariante, ossia le feature raccolte rimangono costanti anche a fronte di trasformazioni affini come la rotazione, la traslazione o cambiamenti di scala, è il SIFT (Scalable Invariant Features Transform).

Quest'ultimo utilizza un algoritmo ideato per la prima volta da David G. Lowe [31] intorno al 1999 e perfezionato nel suo funzionamento sino al 2004. La qualità di tale algoritmo di estrazione deriva dal fatto di possedere un'ottima robustezza rispetto al rumore, alla luminosità e a variazioni del punto di vista di osservazione. Inoltre le feature sono invarianti ad ogni tipo di cambiamento di scala, zoom in o zoom out, a trasformazioni geometriche e ad ogni tipo di rotazione dell'immagine. Per ottenere questa robustezza, i punti dell'immagine dai quali estrarre i descrittori vengono selezionati in prossimità di regioni nelle quali vi è una forte variazione di contrasto, come ad esempio i bordi.

L'algoritmo SIFT essenzialmente trasforma un'immagine in un grande insieme di vettori di feature che hanno proprietà simili a quelle che hanno i neuroni della corteccia temporale inferiore dei primati, responsabili del riconoscimento degli oggetti.

La sua difficile implementazione e la complessità computazionale in fase di test ha fatto sì che questo descrittore, così come altri basati sui punti salienti, fossero momentaneamente esclusi dalla fase successiva di sperimentazione.

Una delle caratteristiche principali che rende questi descrittori così costosi sotto il profilo computazionale è costituita dalla necessità di utilizzare un algoritmo per effettuare il matching dei punti salienti. In letteratura viene spesso menzionato il metodo **RANSAC** che verrà descritto nel paragrafo successivo nonostante il suo inutilizzo in questo elaborato. Infatti per quanto risulti efficace l'algoritmo descritto sopra, in fase di matching tra le feature dei punti salienti non tiene conto della posizione dei keypoint nell'immagine ma ne considera esclusivamente la similarità tra i descrittori.

2.5.1.1 Misure di Similarità applicabili al SIFT

Ottenuti i vettori e le relative distanze con match positivo, dobbiamo utilizzare tali distanze per ottenere una vera e propria similarità tra le immagini quantificandola numericamente sulla base dei risultati fino ad ora registrati.

Per farlo si hanno a disposizione soluzioni differenti tra cui:

- ❖ Riordinare il vettore delle distanze tra match e considerare la media dei primi n elementi, dove n è un valore parametrizzabile.

$$AVGdistance = \frac{1}{n} \sum_{i=1}^n imatches$$

- ❖ Effettuare il rapporto tra il numero complessivo di matches ottenuti e il minimo numero di keypoint tra quelli della prima e della seconda immagine. In questo caso il parametro parametrizzabile per effettuare il test è la soglia utilizzata in fase di match per determinarne o meno l'ambiguità.

$$NMATCHdistance = \frac{|matches|}{\min(|nkeypoints_a|, |nkeypoints_b|)}$$

2.5.1.2 RANSAC come metodo per il matching dei Key Point

RANdom SAMple Consensus è un algoritmo pubblicato per la prima volta da Fischler e Bolles nasce con l'obiettivo di realizzare una trasformazione affine composta da operazioni ricorsive di roto-traslazione, in modo da ottenere in un numero finito di iterazioni, la sovrapposizione dei keypoint delle immagini a confronto che hanno dato match positivo.

Per fare questo, l'algoritmo si basa sulla distinzione di due sottoinsieme di valori: gli inliers, dati la cui distribuzione può essere spiegata utilizzando un modello parametrico, e gli outliers, sottoinsieme invece dei valori sporadici che sono discostati dal trend dei valori generali ai quali appartengono gli inliers, ottenuti prevalentemente da errori di vario genere.

L'algoritmo viene reiterato un numero predefinito di volte generando ogni volta un nuovo modello e un diverso sottoinsieme di inliers, poi attraverso l'utilizzo di un valore di valutazione viene selezionato il modello migliore e i restanti vengono scartati.

In Figura 24 è possibile visualizzare graficamente come opera RANSAC in un insieme di dati in uno spazio bidimensionale, l'immagine mostra come dopo un numero n di iterazioni vengano selezionati gli inliers (in blu) mentre gli outliers (in rosso) vengono scartati dal modello, in modo da non influenzarne la stima.

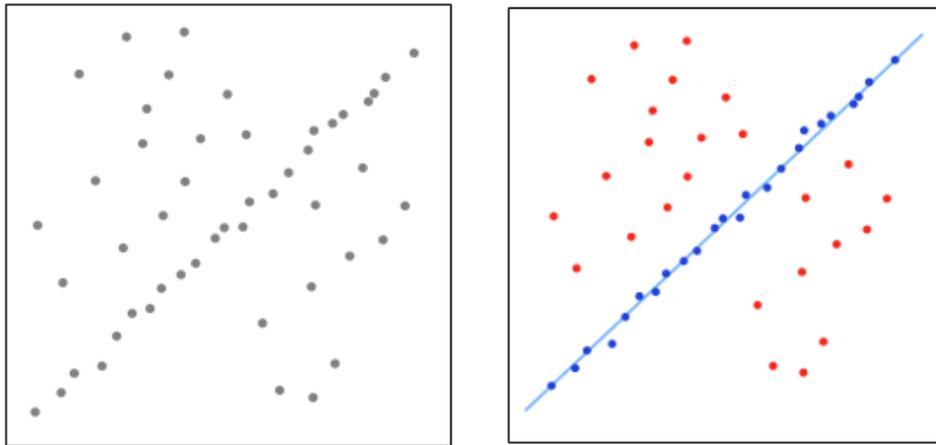


Figura 24: Selezione degli inliers (blu) dopo n iterazioni del metodo.

Grazie alle sue qualità, RANSAC risulta dunque un buon algoritmo per filtrare i match che vengono generati ad esempio dall'algoritmo SIFT.

Una volta terminata la sua applicazione, è necessario quantificare numericamente la similarità delle immagini; per farlo come sempre abbiamo a disposizione un numero ampio di scelte, di seguito sono mostrate tre misure semplici e intuitive:

- ❖ È possibile calcolare il rapporto tra il numero di inliers che fanno parte del sottoinsieme finale ottenuto dalla completa computazione di RANSAC e il numero di match iniziale dati dal SIFT:

$$N_{inliersdistance} = \frac{|match_inliers|}{|match_SIFT|}$$

- ❖ Considerare la media delle distanze euclidee quadratiche di tutti i match facenti parte del sottoinsieme finale degli inliers prodotti dal RANSAC:

$$AVG_{inliersdistance} = \frac{\sum inliers_euclidean_distance}{|inliers|}$$

- ❖ Un altro metodo di distanza nasce dal presupposto che le immagini in esame abbiano lo stesso orientamento e che quindi le linee che congiungono i key point con match positivo non dovrebbero intersecarsi, per questo è ragionevole calcolare il rapporto tra le linee che si intersecano e il numero totale di intersezioni possibili:

$$INTERSECTION = \frac{\textit{intersezioni}}{\textit{possibili intersezioni}}$$

Questi metodi di distanza sono stati riportati qui poiché strettamente correlati con l'algoritmo RANSAC, per la visualizzazione delle misure che abbiamo adottato si rimanda l'attenzione al paragrafo 2.6.

2.5.2 SURF

L'approccio SURF, acronimo di Speeded-Up Robust Features, è una valida alternativa all'utilizzo del metodo SIFT.

Implementato nel 2006 da Herbert Bay, Tinne Tuytelaars e Luc Van Gool [32], nasce proprio con l'obiettivo di migliorare le performance e la robustezza dell'algoritmo SIFT e di raggiungere una maggiore velocità computazionale.

Per rilevare i Key Point all'interno dell'immagine, SURF utilizza la matrice Hessiana, mentre il descrittore utilizzato raccoglie informazioni dalla distribuzione della risposta alla wavelet di Haar all'interno di un intorno dei punti salienti. Proprio come l'algoritmo SIFT, questo metodo utilizza unicamente informazioni riguardanti i toni di grigio dell'immagine senza utilizzare proprietà legate al colore.

La maggiore velocità di tale algoritmo si ottiene quando SURF sfrutta una rappresentazione intermedia dell'immagine da processare, chiamata *immagine intermedia*, a differenza di altri metodi che usano il *Box filter*. L'estrazione dei punti salienti si basa sulla *matrice Hessiana* che grazie a determinate approssimazioni risulta avere un'elevata accuratezza e ottima performance computazionale a discapito di qualche problema di perdita di reperibilità a fronte di alcuni tipi di rotazioni. Infine si ottiene il descrittore lavorando sugli orientamenti.

Come mostra la Figura 24 possiamo elaborare una delle differenze sostanziali tra SIFT e SURF; nel primo caso l'immagine iniziale viene più volte campionata e scalata mentre nel secondo caso grazie all'utilizzo delle immagini integrali è possibile incrementare o diminuire le dimensioni del filtro mantenendo costanti quelle dell'immagine.

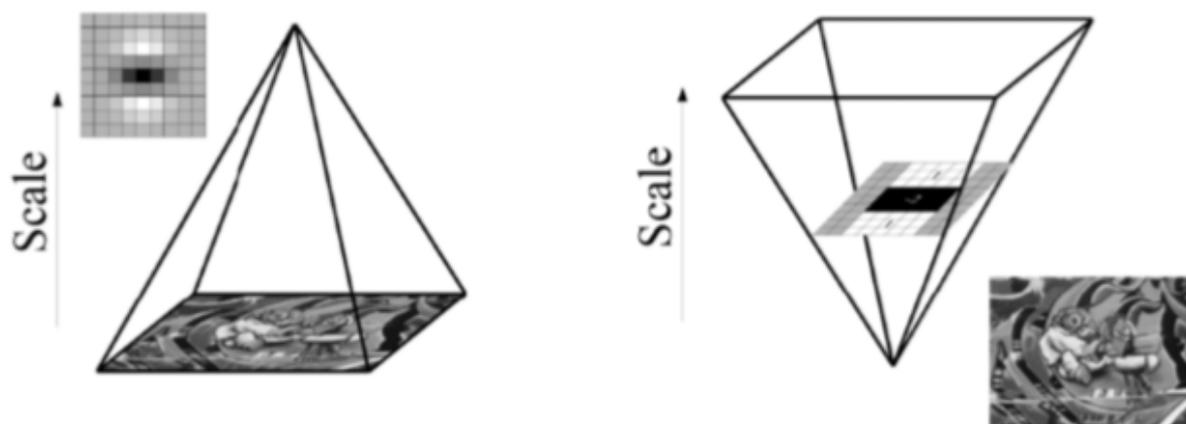


Figura 25: Differenza nella variazione delle dimensioni dell'immagine: sulla sinistra è riportato il SIFT, sulla destra il SURF.

2.6 Misure di Similarità

Quando parliamo di CBIR e utilizziamo un'immagine come query, l'obiettivo è quello di calcolare la similarità tra l'immagine iniziale e ogni singola immagine presente nel dataset e a seconda del valore che si ottiene da tale comparazione il sistema è in grado di interpretare le corrispondenze tra le immagini.

Fino ad ora abbiamo parlato di features estratte dalle immagini che vanno a costituire i descrittori; le misure di similarità utilizzano questi vettori di informazioni per estrapolarne la distanza. La letteratura offre un archivio vastissimo di algoritmi utilizzabili suggerendo quali distanze supportano meglio il particolare descrittore.

Di seguito viene riportato un elenco delle misure più conosciute e applicate negli esempi ed articoli analizzati [33], per capire al meglio le formule matematiche si sottolinea che i vettori \mathbf{x} e \mathbf{y} appartengono rispettivamente alle immagini oggetto della comparazione, mentre i vettori contrassegnati da una barretta rappresenteranno la media dei valori contenuti nel valore stesso. I vettori \mathbf{X} e \mathbf{Y} rappresenteranno la distribuzione di probabilità del vettore stesso, dove la sommatoria degli elementi è uguale all'unità, infine il valore μ sarà la media matematica dei vettori \mathbf{x} e \mathbf{y} , $\mu = \frac{x+y}{2}$.

❖ Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

❖ Chi-Square Distance

$$d(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^n \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

❖ **Cosine Distance**

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}^T}{\sqrt{\mathbf{x} \cdot \mathbf{x}^T} * \sqrt{\mathbf{y} \cdot \mathbf{y}^T}}$$

❖ **CityBlock Distance**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|$$

❖ **Infinity Distance**

$$d(\mathbf{x}, \mathbf{y}) = \max_{i=1}^n |x_i - y_i|$$

❖ **Jeffrey Divergence**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n x_i \log \frac{x_i}{\mu_i} + y_i \log \frac{y_i}{\mu_i}$$

Quando consideriamo una funzione il cui risultato è una distanza $d(\mathbf{x}, \mathbf{y})$ è sottointeso che la *similarità* si ottiene semplicemente ponendo $1 - d(\mathbf{x}, \mathbf{y})$ nell'ipotesi in cui la distanza vari tra 0 e 1, altrimenti tale valore deve essere normalizzato.

Nei test che seguiranno, e per i descrittori basati su una rappresentazione ad istogrammi si è scelto di utilizzare sempre la misura **Chi-Square Distance**, mentre ad alcuni descrittori di tessitura si è applicata la **Euclidean Distance**. Un'altra misura spesso utilizzata in letteratura su descrittori di tessitura è rappresentata dalla Jeffrey Divergence. Per il descrittore di forma abbiamo usato la **Cosine Distance**.

All'interno del framework è stata inoltre introdotta l'implementazione della Hamming Distance, applicata ai metodi Hash, che opera molto semplicemente contando il numero di elementi diversi tra due vettori considerandone la posizione

Capitolo 3

Indicatori di Prestazione

3.1 Introduzione ai Performance Indicator

In questo lavoro i diversi descrittori sono stati valutati sperimentalmente nell'ambito di un problema di "instance recognition": differentemente da un CBIR classico in cui la valutazione delle performance avviene analizzando la lista delle immagini restituite dal sistema in risposta ad un'immagine query, l'instance recognition consiste in un problema di classificazione a 2 classi in cui si valuta il sistema in base alla sua capacità di predire se le due immagini confrontate appartengono o meno alla stessa classe. Con indicatori di prestazione intendiamo quegli algoritmi che restituiscono valori per interpretare il grado di risposta positiva che ha ottenuto il descrittore che abbiamo utilizzato con la relativa misura di similarità nello specifico dataset.

In letteratura, sono stati proposti molti indicatori di performance, prima di proseguire con una descrizione teorica ed applicativa dei più frequentemente utilizzati, come l'Area Under Curve (AUC), l'Accuracy delle classificazioni del KNN (K Nearest Neighbors), la curva ROC o il grafico Genuine-Impostor, saranno illustrati alcuni dei modelli di classificazione:

- **Statistici**, memorizzano parametri legati alle distribuzioni di probabilità relative ad ogni singola classe e attributo e per stimare l'appartenenza di un oggetto ad una specifica classe si calcola la probabilità di appartenenza ad una di esse.

- **Basati su esempi**, memorizzano tutti gli esempi che gli vengono sottoposti, poi la classificazione avviene assegnando la classe all'oggetto in base alla somiglianza con gli esempi memorizzati, come ad esempio il K-nearest neighbor, di cui ci occuperemo di seguito.
- **Matematici**, che riescono a creare, grazie agli esempi di training, una funzione matematica di classificazione. Essa viene utilizzata sui nuovi dati di test per effettuare correttamente la catalogazione.
- **Logici**, la classificazione avviene semplicemente applicando delle condizioni logiche sugli attributi estratti dai dati.

3.1 Receiver Operator Characteristic

La curva che otteniamo con il Receiver Operator Characteristic (ROC) è utilizzata per ottenere una rappresentazione grafica della qualità di un classificatore binario. L'applicazione del metodo risulta molto intuitiva.

Per semplificare al massimo la spiegazione, immaginiamo due insiemi A e B e l'insieme C dato dall'intersezione di A e di B. Per esempio ad A appartengono immagini di monumenti e a B foto di paesaggi e definiamo U la somma dei due insiemi che può rappresentare il nostro dataset di immagini di monumenti e paesaggi.

Il nostro obiettivo è quello di classificare un oggetto come monumento o come paesaggio (problema binario). Immaginiamo ora di disporre di un algoritmo in grado di dividere l'insieme U nelle due categorie di fotografie cercando di separare i due insiemi nel modo migliore. Ovviamente l'occhio umano non avrebbe alcun problema ad interpretare le immagini e etichettarle nel corretto insieme ma purtroppo l'esperienza dimostra che è impossibile dividere perfettamente i due insiemi, poiché le features estratte da ogni singola immagine attraverso gli algoritmi non sono sufficienti per la distinzione. In poche parole possiamo avere un piccolo monumento nel quale è presente una parte secondaria di paesaggio e il sistema non è in grado

di identificare correttamente la categoria. Sarà quindi necessario introdurre una soglia, che venga utilizzata dal classificatore per decidere a quale categoria una immagine deve appartenere. Il concetto può essere comparato con un problema digitale, dove difficilmente si hanno segnali elettrici uguali a 1 o 0 e di conseguenza si introducono valori di soglia per far interpretare correttamente i valori ai sistemi

Prendendo una soglia β , il classificatore può trovarsi davanti a 4 casi:

- $x > \beta$ & la foto è un monumento, abbiamo un “True Positive (TP)”;
- $x > \beta$ & la foto è un paesaggio, abbiamo un “False Positive (FP)”;
- $x < \beta$ & la foto è un paesaggio, abbiamo un “False Negative (TN)”;
- $x < \beta$ & la foto è un monumento, abbiamo un “True Negative (FN)”;

Prima di procedere e calcolare ulteriori valori è utile normalizzare i dati ottenuti; per fare ciò si prendono i quattro valori descritti qui sopra e ognuno viene diviso per il numero di elementi dell’insieme di appartenenza: i valori relativi ai monumenti saranno divisi per il numero totale di elementi dell’insieme A mentre i valori relativi ai paesaggi saranno divisi per il numero degli elementi di B.

Con i nuovi dati normalizzati possiamo facilmente trovarci valori sperimentali fondamentali che in letteratura appaiono spesso come *Precision*, *Sensitivity*, *Accuracy*, *Specificity* e che ritroveremo anche nella definizione di altri performance indicator:

$$Precision = \frac{TP}{(TP + FP)}$$

$$Sensitivity = \frac{TP}{(TP + FN)}$$

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN}$$

$$Specificity = \frac{TN}{(TN + FP)}$$

$$Error Rate = 1 - Accuracy$$

Al variare di β , ossia aumentando o diminuendo la soglia, otteniamo dati diversi; ad esempio prendendo un $\beta' > \beta$ possiamo ottenere un valore di True Negative (TN) normalizzato uguale a 1, avendo una massima specificità e contemporaneamente aumenterebbero i False Negative (FN); allo stesso modo se prendessimo un $\beta'' < \beta$ otterremmo l'effetto contrario.

Con i dati ottenuti variando il valore della soglia possiamo stabilire la curva Roc, ottenuta graficando i valori dello schema di classificazione su un diagramma bidimensionale il cui grafico varia a seconda del valore della soglia.

Analizzando il grafico identifichiamo sull'asse delle ascisse il valore dato da *1-Accuratezza* mentre sull'asse delle ordinate avremo il valore della *Sensibilità*.

Ogni punto $P(x,y)$ del grafico è quindi identificato come segue:

$$P(x,y) = 1 - \frac{TN}{(TN+FP)}, \frac{TP}{(TP+FN)} ;$$

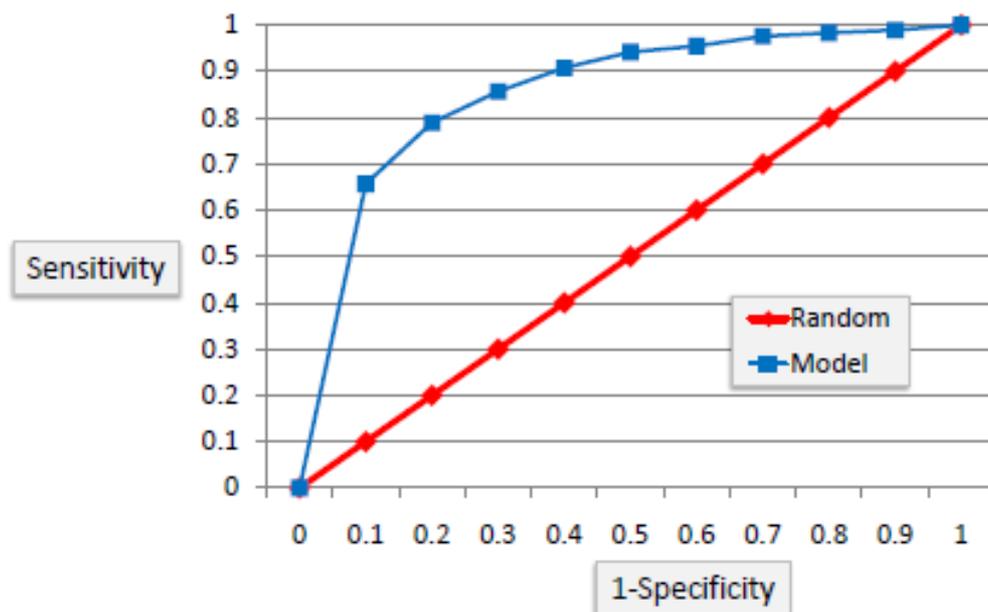


Figura 26: Grafico della curva ROC.

La curva risultante dall'utilizzo di tutti i possibili valori di soglia si dice ROC del sistema dati/classificatore.

3.2 Area Under Curve

Attraverso questa curva è possibile identificare un dato di fondamentale importanza, rappresentato dall'area al di sotto della curva che definiremo AUC (Area Under Curve).

La AUC rappresenta un indicatore di performance della bontà della curva ROC e del classificatore indipendente dalla soglia: un valore di AUC prossimo a 1 denota un classificatore quasi ottimale, un valore pari a 0.5 denota il classificatore random, mentre un valori inferiori a 0.5 indicano che il classificatore “inverte le risposte”. E' stato inoltre dimostrato che l'AUC rappresenta la probabilità che il classificatore assegni a un esempio positivo un valore di confidenza maggiore rispetto a un esempio negativo, ovvero la probabilità che un elemento casuale venga catalogato correttamente dal sistema.

I valori di soglia che distano maggiormente dalla diagonale principale del grafico forniscono una probabilità maggiore di classificazione corretta. Nel caso limite in cui la soglia si trovi esattamente sulla diagonale principale, il valore dell'area sarà 0.5, questo equivale ad avere una scelta di classificazione totalmente casuale.

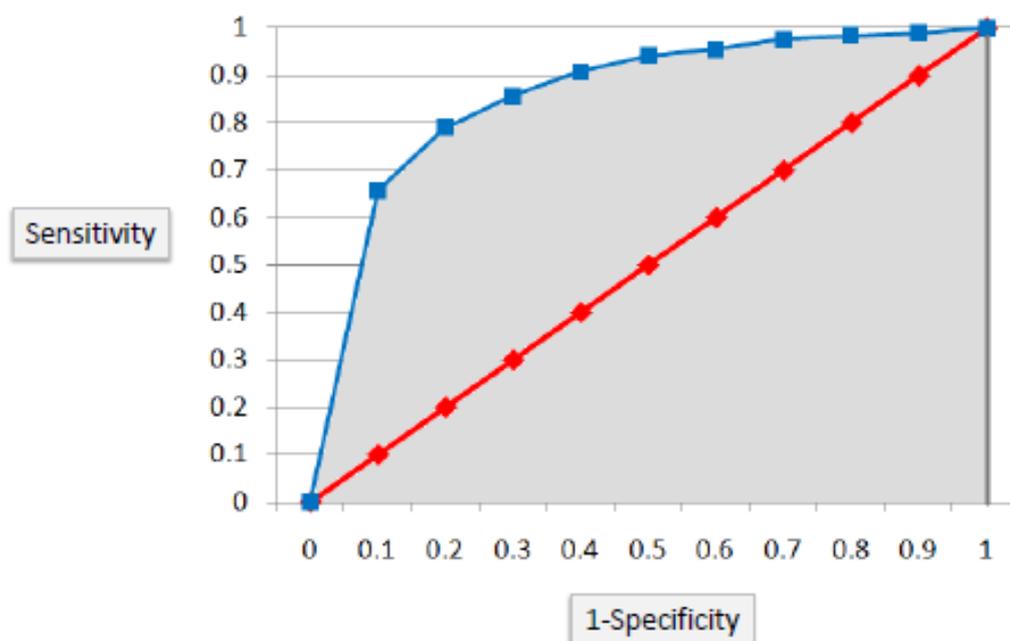


Figura 27: Rappresentazione dell'Area Under Curve ottenuta dalla curva ROC di Figura 26.

3.3 Accuracy ottenuta con il K-Nearest-Neighbors

Il K-Nearest-Neighbors è un algoritmo che classifica un determinato oggetto basandosi esclusivamente sulle caratteristiche dei K oggetti più simili a quello in esame.

L'oggetto sarà identificato con la classe degli oggetti maggiormente presenti nell'intorno dei k elementi; è quindi importante la scelta del parametro k che varierà a seconda dei dati in esame. La scelta di k diversi può infatti portare a diversi risultati. Ad esempio, preso il valore di k uguale a uno, l'oggetto verrà classificato con la stessa classe dell'oggetto più vicino ad esso. La qualità della classificazione si può misurare utilizzando l'Accuracy, ovvero la percentuale di immagini correttamente classificate.

La rappresentazione grafica in Figura 1 rappresenta un esempio dell'applicazione dell'algoritmo K-NN con un $k=3$, infatti il nostro oggetto (punto verde) verrà classificato come triangolo rosso in quanto nel suo intorno di k-esimi elementi essi sono i più ricorrenti.

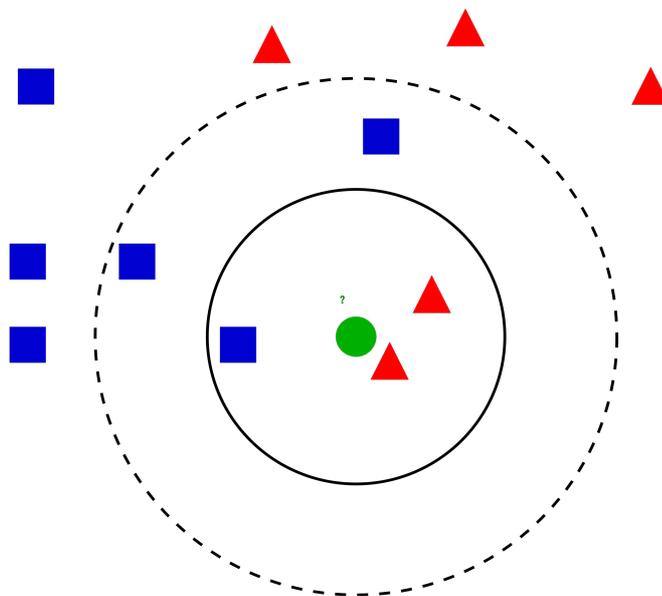


Figura 28: Idealizzazione grafica del comportamento del K-NN.

3.4 Grafico Genuine Impostor

Il grafico Genuine-Impostor è una trasposizione in ambito CBIR di una misura usata nei sistemi biometrici: definendo “Genuine” tutte le coppie di match di immagini della stessa classe e “Impostor” le coppie di match di immagini di classe diverse, in grafico rappresenta la distribuzione di tali valori in modo da verificare visivamente se le due curve sono separabili. Ovviamente per semplificare la lettura del grafico è necessario che i valori di similarità (o distanza) tra due immagine siano normalizzati tra 0 e 1 dove il valore 1 significa che le due immagini sono congruenti e il valore 0 incongruenti. Il grafico Genuine-Impostor rappresenta sullo stesso piano la densità delle distanze ottenute dai match genuine e la densità delle distanze dei match impostor. Osservando tale grafico si noterà un punto di intersezione tra le due curve e proprio questo valore o un suo intorno, può essere preso come valore di soglia al fine di massimizzare l’accuracy del sistema, precedentemente vista. È importante ricordare che la scelta della soglia dipende spesso anche dagli scopi applicativi del nostro sistema e dai margini di sicurezza ai quali l’applicazione richiede di attenersi; ad esempio in un problema biometrico di riconoscimento facciale o di impronte digitali, il valore di soglia viene preso molto alto in modo da assicurarci che il sistema scarti ogni possibile impostor e classificando come tale anche qualche genuine. Al contrario, un valore di soglia basso è del tutto accettabile in un problema di instance recognition in cui non è grave l’accettazione di falsi positivi.

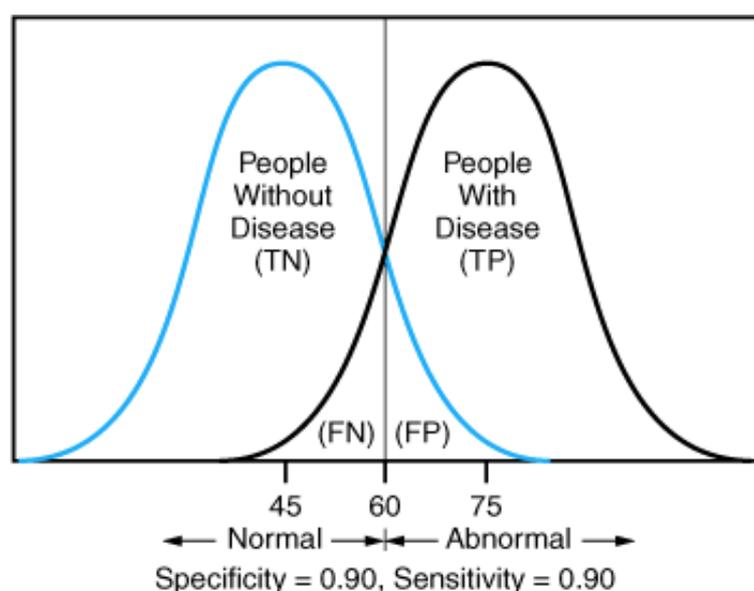


Figura 29: Grafico Genuine-Impostor relativo a un problema medico, sul quale è possibile visualizzare le porzioni che identificano i TP, TN, FP e FN.

Capitolo 4

Risultati Sperimentali

4.1 Introduzione ai test

In questo capitolo vengono raccolte le attività sperimentali svolte dopo l'analisi del problema e la raccolta di informazioni teoriche che compongono l'elaborato. Per la realizzazione di test sono stati scelti alcuni dataset, le cui caratteristiche sono riportate nel paragrafo seguente insieme ad una tabella che ne racchiude le informazioni principali. Tutta l'attività di test è stata svolta utilizzando l'applicazione MatLab, piattaforma che offre una buonissima libreria preesistente per la gestione delle immagini e soprattutto permette un'ottima possibilità di lavorare attraverso la gestione di matrici ed elaborare grafici in modo da rendere possibile la realizzazione di informazioni facilmente leggibili. Essi sono consultabili nei paragrafi successivi suddivisi per dataset e saranno seguiti da una breve analisi

Viene riportato il link alla pagina web di MatLab relativa all'intera documentazione della libreria disponibile per l'Image Processing mentre i link ai dataset di cui ci siamo forniti saranno disponibili in Bibliografia.

MatLab Image Processing Toolbox: <http://it.mathworks.com/help/images/index.html>.

4.2 Descrizione dei Dataset utilizzati

Per la fase di testing sono stati utilizzati 7 dataset con caratteristiche eterogenee al fine di ottenere una panoramica completa del comportamento del software a seconda dei casi in esame. Di seguito verranno elencati i dataset e verrà fornita una descrizione sintetica delle proprietà e del contenuto.

La tabella seguente sintetizza le informazioni principali dei dataset descritti in precedenza.

Dataset	N° immagini	N° classi	Immagini per classe	Risoluzione
Caltech Buildings	250	50	5	2.048x1.536
ZuBud	1005	201	5	640x480
Shieffield Buildings	4179	40	variabili	160x120
ZuBud Objects	265	53	5	320x240
WangDB	1.000	10	100	384x256
ItaLa	1.423	40	variabili	variabile
ColorBrodatz	448	112	4	320x320

Tabella 1: Riassunto delle informazioni dei dataset.

➤ **Caltech buildings** [35]

Costituito da un totale di 250 fotografie di edifici appartenenti al campus del California Institute of Technology. Sono presenti 50 edifici e per ognuno vi sono 5 di fotografie scattate da angolature differenti garantendo diversi prospetti dello stesso. Tutte le foto sono state effettuate più o meno nello stesso momento della giornata garantendo così la stessa luminosità per ogni immagine. La risoluzione, 2048x1536 per ognuna delle immagini, risulta decisamente alta, di conseguenza in fase di testing si preferisce utilizzare delle semplificazioni del Caltech buildings in modo di ridurre il tempo di esecuzione dei test. In ogni caso i risultati sperimentali che seguiranno sono relativi al Caltech Buildings originale.



Figura 30: Esempio del contenuto del dataset Caltech Buildings.

➤ **ZuBud** [36]

Il Dataset ZuBud contiene ben 1005 immagini, tutte raffiguranti edifici della città di Zurigo. Per ognuno dei 201 elementi fotografati sono presenti 5 foto; le caratteristiche sono le stesse del Caltech Buildings, ossia fotografie scattate da angolazioni diverse e con la stessa luminosità del giorno, inoltre però le foto possono presentare anche una rotazione differente, cioè non sono tutte orientate nel verso corretto; il pregio dello ZuBud è quello di contenere un numero molto grande di elementi diversi e con una risoluzione inferiore al Caltech di 640x480.



Figura 31: Esempio del contenuto del dataset ZuBud.

➤ **Sheffield Buildings** [37]

Questo dataset contiene 4.178 file, sono rappresentati 40 edifici fotografati da angolazioni diverse ma a differenza delle due raccolte precedenti, le fotografie sono state effettuate in momenti diversi della giornata, garantendo una diversa tipologia di illuminazione. Non vi è un numero costante d'immagini per elemento, la quantità per singolo edificio varia dalle 50 a più di 300 acquisizioni. Queste caratteristiche lo rendono sicuramente il dataset più eterogeneo. La bassa risoluzione delle immagini, 160x120, garantisce un'ottimizzazione dei tempi di esecuzione degli algoritmi durante la fase di svolgimento dei test.

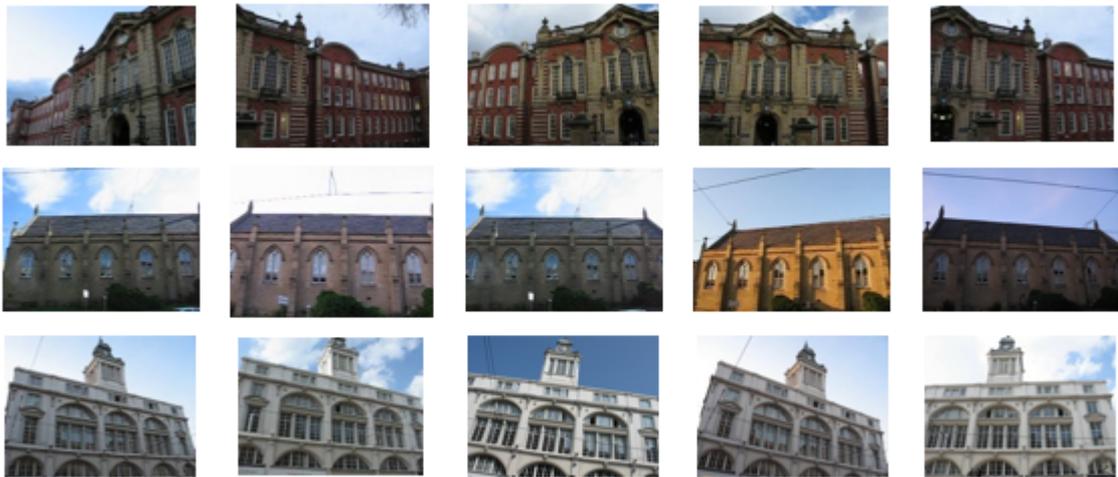


Figura 32: Esempio del contenuto del dataset Sheffield Buildings.

➤ **ZuBud Objects** [38]

Il dataset contiene immagini di 53 semplici oggetti fotografati da diverse angolazioni, per un totale di 5 immagini cadauno. La bassa risoluzione a 320x240, come già sottolineato influirà positivamente sui tempi di testing. L'utilizzo di raccolte fotografiche diverse da monumenti ed edifici è data dal fatto di voler osservare il comportamento degli algoritmi su diverse tipologie di elementi analizzabili.



Figura 33: Esempio del contenuto del dataset ZuBud Objects.

➤ **WangDB [39]**

Dataset composto da 1000 immagini che raffigurano 10 diverse tipologie di elementi, abbiamo 100 foto per ogni tipologia con una risoluzione standard di 384x256 o nel caso si trovi in posizione verticale 256x384.

In questo caso quando parliamo di elemento non ci si riferisce alla ripetizione dello stesso soggetto ma all'appartenenza alla stessa classe, quindi immagini di monumenti diversi apparterranno comunque alla stessa classe.

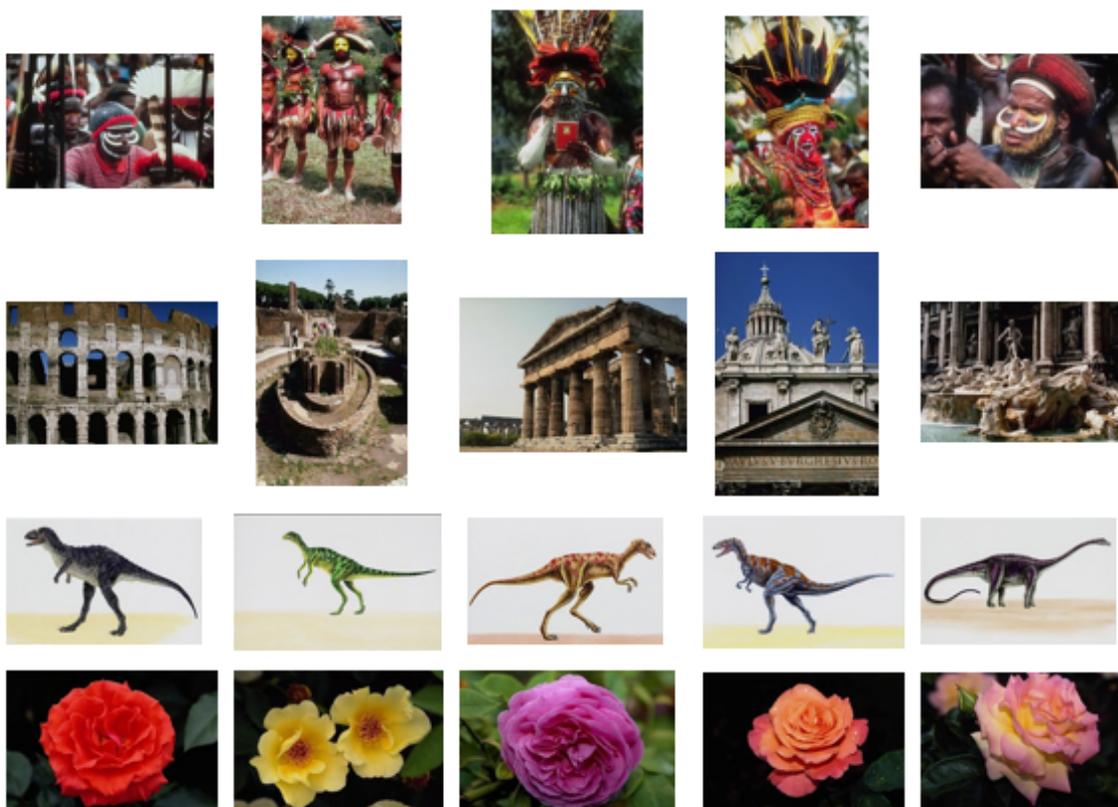


Figura 34: Esempio del contenuto del dataset WangDB.

➤ **ItaLa** [40]

Come suggerisce il nome stesso di questa raccolta di immagini, l'ItaLa ingloba un numero considerevole di immagini provenienti dai monumenti presenti sul territorio italiano. In totale sono presenti 40 monumenti e 36 per ognuno di essi. La risoluzione delle foto non è standardizzata così come sono diverse le raffigurazioni del soggetto, che variano sia riguardo l'angolatura, sia per gli effetti di luce dovuti alle condizioni atmosferiche.



Figura 35: Esempio del contenuto del dataset ItaLa.

➤ **ColorBrodatz** [41]

Questo dataset è stato scelto per poter testare i vari metodi non solo in immagini raffiguranti elementi definiti e ben precisi ma estendere l'esperienza a fotografie ritraenti un tema di tessitura. Il Color Brodatz a differenza di moltissimi database di tessitura che contengono solamente immagini in bianco e nero, è composto da fotografie colorate, permettendo l'uso di tutti i metodi a nostra disposizione.

Il dataset contiene 448 immagini, 4 per ognuna delle 112 classi e tutte presentano la stessa risoluzione costante.

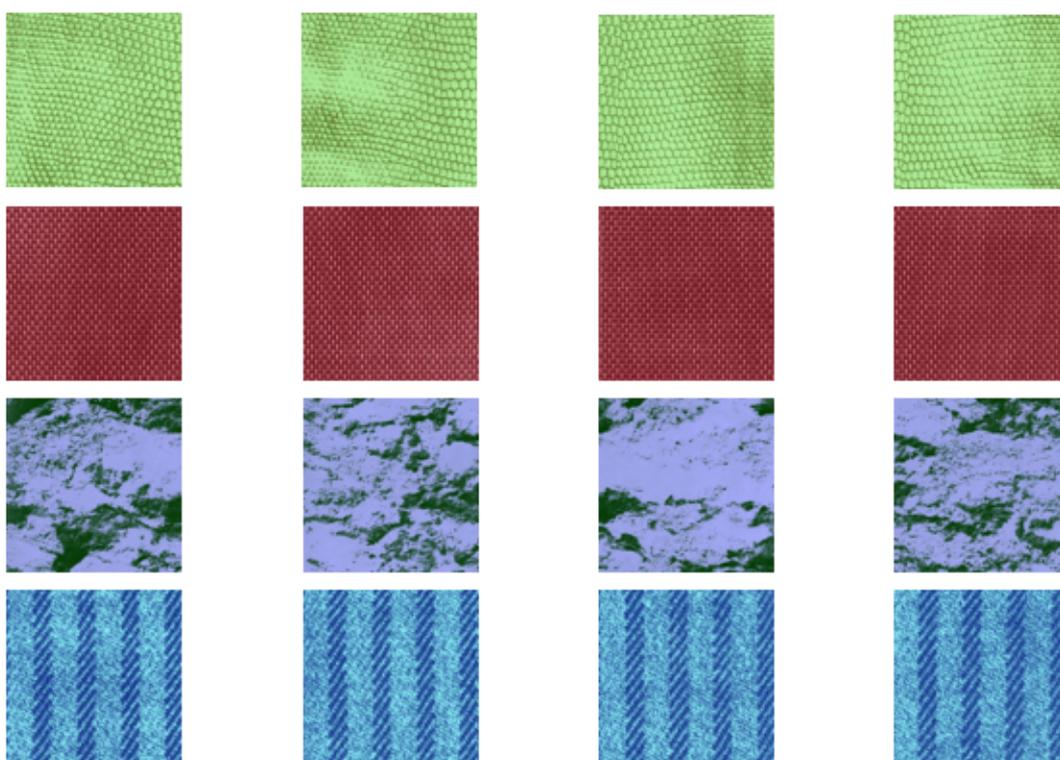


Figura 36: Esempio del contenuto del dataset ColorBrodatz.

4.3 Dati sperimentali ottenuti

Di seguito sono riportate le Tabelle che incorporano i risultati ottenuti attraverso l'applicazione dei diversi descrittori con la misura di similarità più indicata dopo l'analisi effettuata in fase preliminare; ogni accoppiamento è stato preso in esame sui dataset descritti. Le performance relative alle capacità dei metodi sono state misurate in termini di AUC e di Accuracy calcolata usando il K-Nearest Neighbors con valore di K pari a 5; la scelta di utilizzare questo valore è stata dettata in quanto alcuni dei dataset presi in esame avevano esattamente 5 foto per ogni "classe", di conseguenza se avessimo utilizzato un valore più elevato di K, concettualmente avremmo commesso un errore in quanto è impossibile classificare correttamente più di 5 immagini.

Nei paragrafi successivi seguiranno le raccolte dei grafici suddivisi per dataset dove sarà possibile osservare i comportamenti delle curve ROC e le relazioni Genuine-Impostor relative ad ogni algoritmo. Faccio notare che nei grafici Genuine-Impostor abbiamo inserito tutti i risultati a parte quello dell'algoritmo PHash, sia per ragioni di spazio sia perché si è osservato che il risultato ottenuto corrisponde quasi perfettamente a quello del DHash. Inoltre si è scelto di riportare l'implementazione e il comportamento di un'estensione dei metodi che ho implementato soltanto nel caso del dataset Caltech Buildings (paragrafo 4.3.1) poiché si è constatato un miglioramento netto ed evidente in confronto agli altri tentativi.

Tra le altre considerazioni preliminari si evidenzia il comportamento dei valori dell'Accuracy sul dataset Shieffield Buildings dove la cardinalità delle classi di immagini (come riportato in Tabella 1) è variabile e mai inferiore a 60 fotografie, di conseguenza per le ragioni precedentemente osservate il valore dell'Accuracy sarà molto più elevato in quanto per ogni classe sono presenti un numero considerevole di immagini molto simili tra di loro, dove l'angolo di inquadratura varia di pochi gradi.

Nei grafici Genuine-Impostor sarà sempre riportata la curva Genuine identificata con il colore blu e la curva Impostor identificata in rosso.

AUC		Dataset									
Descrittore	Distanza	Caltech	ZuBud	Shieffield	ObjZuBud	Wang	ItaLa	ColorBrodatz			
HSV	DistHist	0.9323	0.9679	0.8096	0.8415	0.8119	0.6280	0.9978			
RGB	DistHist	0.8768	0.9425	0.7577	0.8354	0.7742	0.5983	0.9942			
CCV	DistHist	0.8779	0.9525	0.7440	0.8481	0.7368	0.6135	0.9938			
HoG	DistHoG	0.8122	0.8608	0.8169	0.7537	0.6263	0.6848	0.8272			
AHash	DistPHash	0.8293	0.8634	0.6375	0.7135	0.6359	0.6025	0.6822			
Phash	DistPHash	0.7105	0.7973	0.5881	0.6912	0.5590	0.5761	0.6534			
DHash	DistPHash	0.7849	0.8248	0.6928	0.7258	0.5870	0.6209	0.6642			
Tamura	DistEuc	0.7850	0.8654	0.7135	0.8024	0.7000	0.6257	0.8743			
Haralick	DistEuc	0.7696	0.7853	0.6195	0.7841	0.6195	0.5630	0.9157			
LBP	DistHist	0.8596	0.8852	0.8004	0.7745	0.7905	0.6707	0.9874			

Tabella 2: Valori di AUC ottenuti dai test.

Accuracy with K-NN		Dataset									
Descrittore	Distanza	Caltech	ZuBud	Shieffield	ObjZuBud	Wang	ItaLa	ColorBrodatz			
HSV	DistHist	66.40%	78.01%	97.34%	57.74%	76.70%	31.14%	95.76%			
RGB	DistHist	42.40%	61.09%	94.95%	52.45%	67.70%	30.79%	94.64%			
CCV	DistHist	43.60%	61.99%	95.17%	63.40%	69.40%	31.64%	84.15%			
HoG	DistHoG	32.00%	53.63%	96.74%	16.23%	43.00%	60.79%	23.88%			
AHash	DistPHash	24.00%	30.75%	70.30%	10.19%	35.70%	30.79%	7.14%			
PHash	DistPHash	9.20%	20.40%	79.78%	7.92%	29.90%	31.07%	1.79%			
DHash	DistPHash	17.60%	27.66%	84.47%	8.68%	31.20%	31.86%	2.23%			
Tamura	DistEuc	8.80%	9.85%	58.07%	14.72%	31.90%	24.93%	15.18%			
Haralick	DistEuc	9.60%	8.26%	69.41%	19.62%	41.60%	24.64%	48.44%			
LBP	DistHist	48.00%	58.51%	96.19%	22.64%	73.90%	38.21%	89.96%			

Tabella 3: Valori di Accuracy ottenuti nei test.

4.3.1 Caltech Building

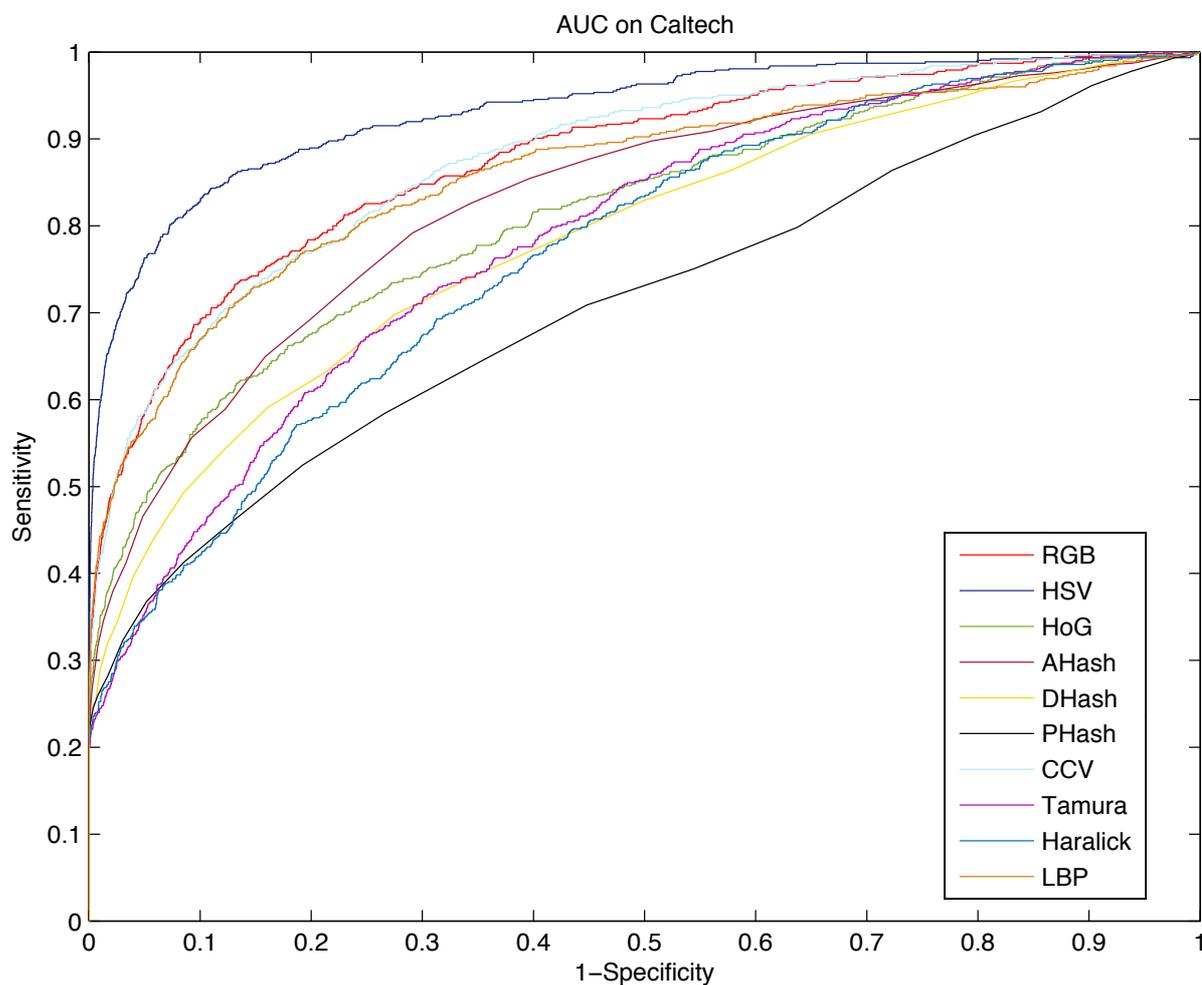


Figura 37: Rappresentazione delle curve ROC ottenute sul dataset Caltech Buildings

Si intuisce che la curva ROC migliore si è ottenuta tramite l'applicazione dell'istogramma HSV mentre, sempre per quanto riguarda gli istogrammi di colore, hanno fornito risultati quasi equivalenti RGB e CCV. Tra i descrittori di tessitura indubbiamente LBP ha dato un ottimo risultato.

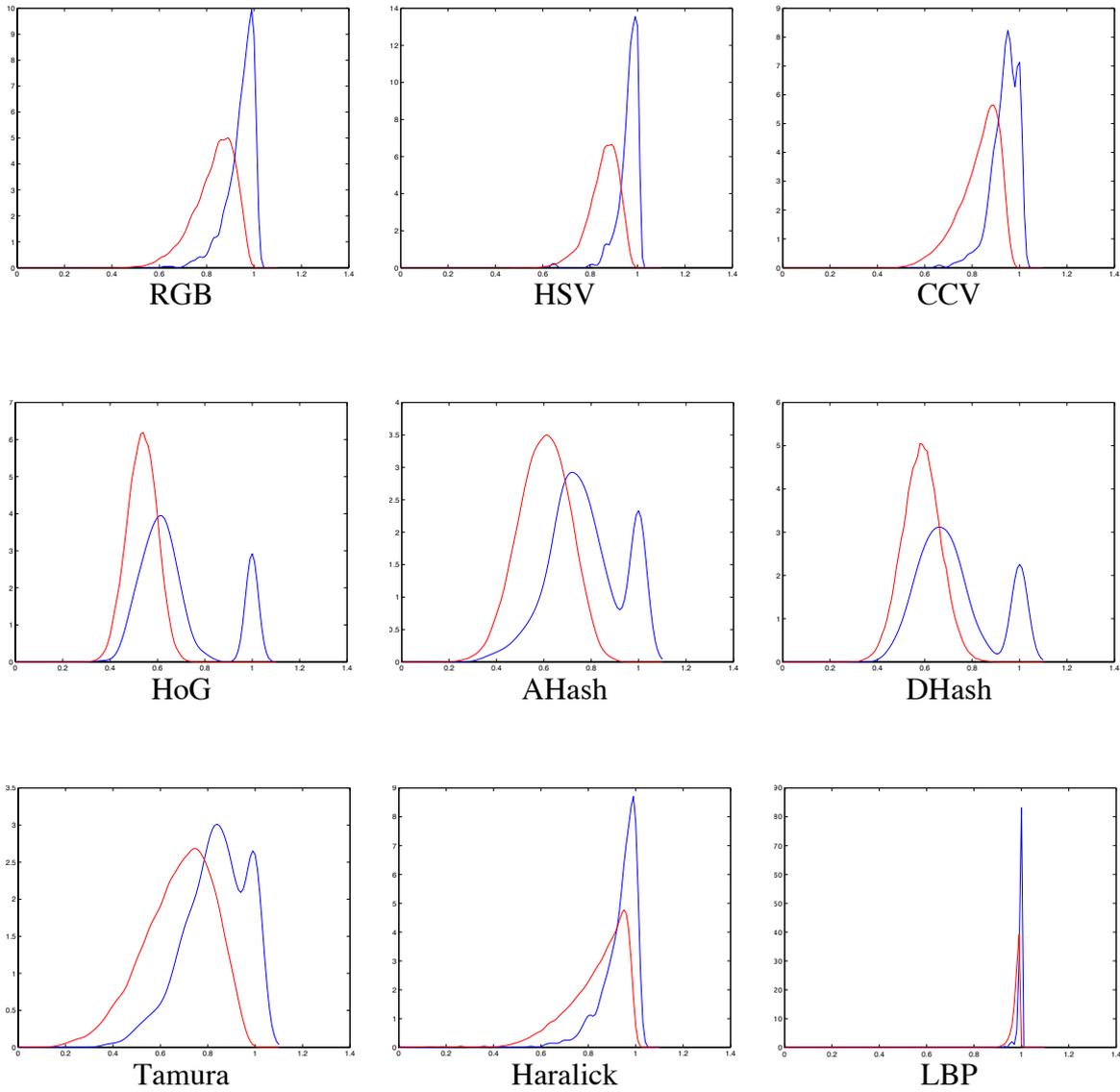


Figura 38: Grafici Genuine-Impostor per il dataset Caltech Buildings

Anche questi grafici Genuine/Impostor suggeriscono come i descrittori di colore abbinati alla Chi-Square distance rappresentino la soluzione migliore.

4.3.2 ZuBud

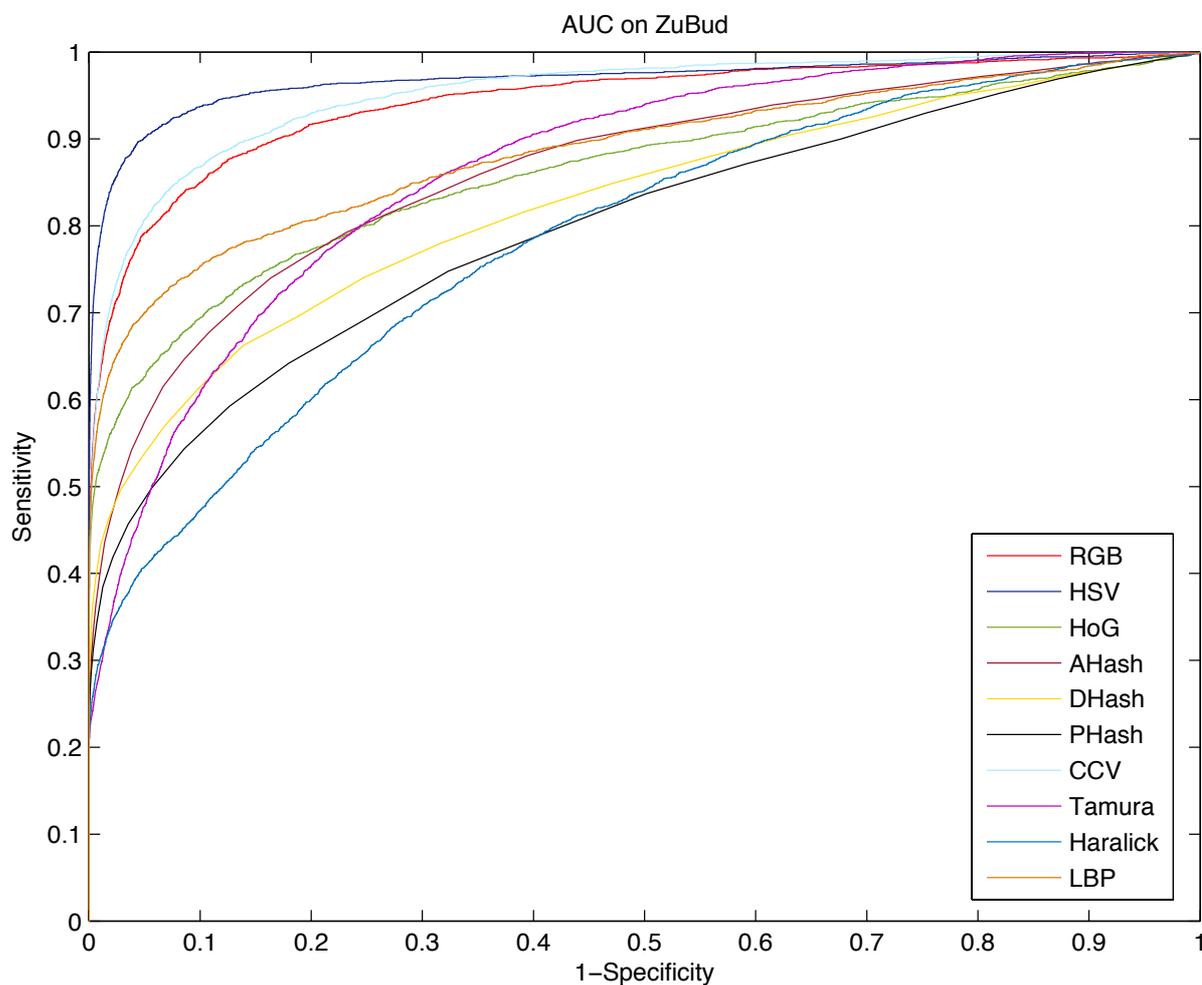


Figura 39: Rappresentazione delle curve ROC ottenute sul dataset ZuBud.

Anche in questo caso il descrittore di colore HSV risulta molto efficace rispetto ai contendenti, seguito da RGB e CCV. Rispetto al caso precedente si osserva un lieve miglioramento dei descrittori Tamura e HoG.

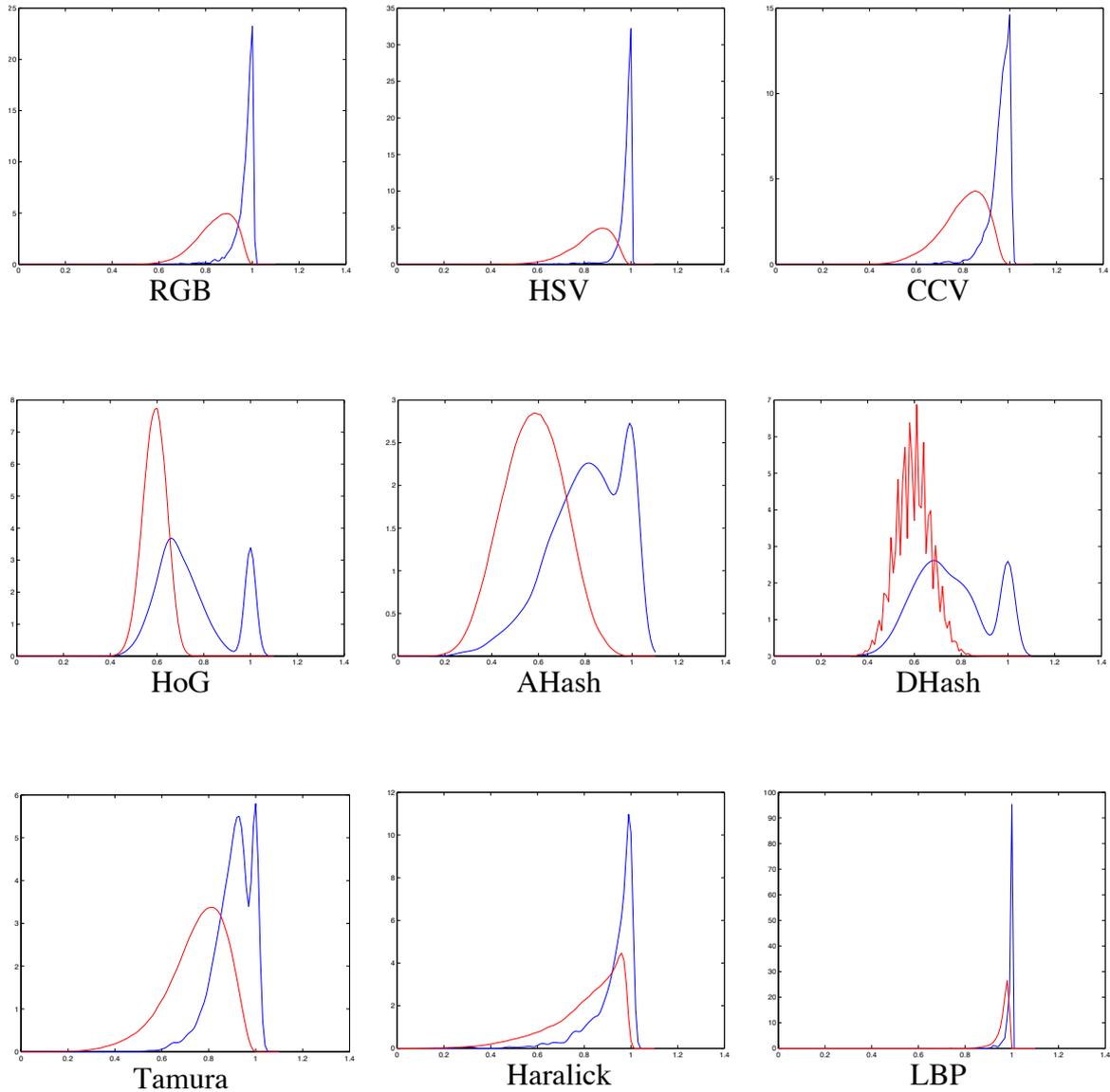


Figura 40: Grafici Genuine-Impostor per il dataset ZuBud.

Anche sul dataset ZuBud è osservabile che i descrittori di colore forniscono i risultati migliori, in effetti questo è in linea con il fatto che i due dataset risultano abbastanza simili come caratteristiche.

4.3.3 Sheffield

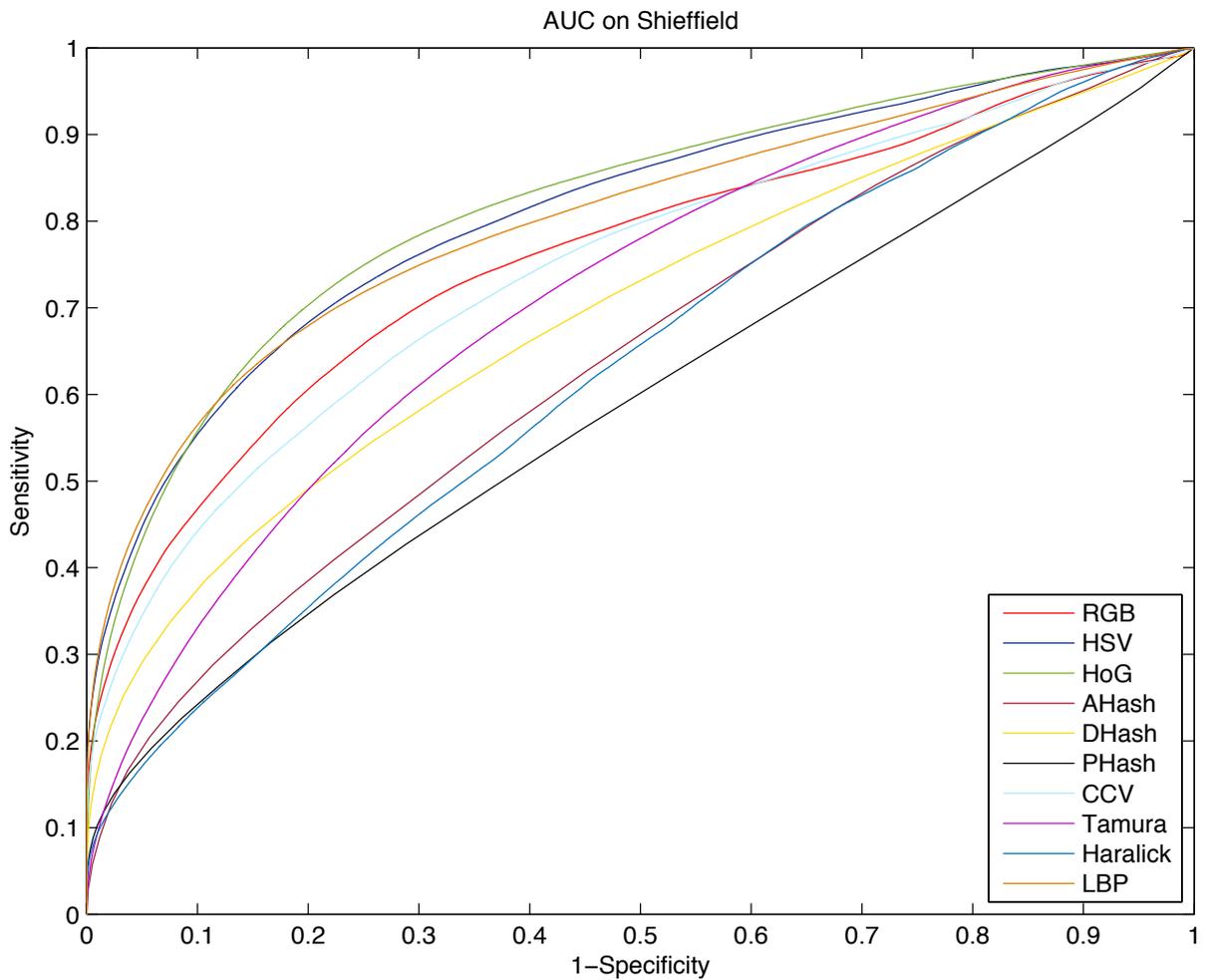


Figura 41: Rappresentazione delle curve ROC ottenute sul dataset Sheffield.

Osservando le curve ROC dello Sheffield si vede subito la grossa diminuzione di performance da parte dei descrittori di colore RGB, CCV, di HSV e LBP (tessitura) che sono molto simili, mentre il più performante sembra essere il descrittore di forma HoG.

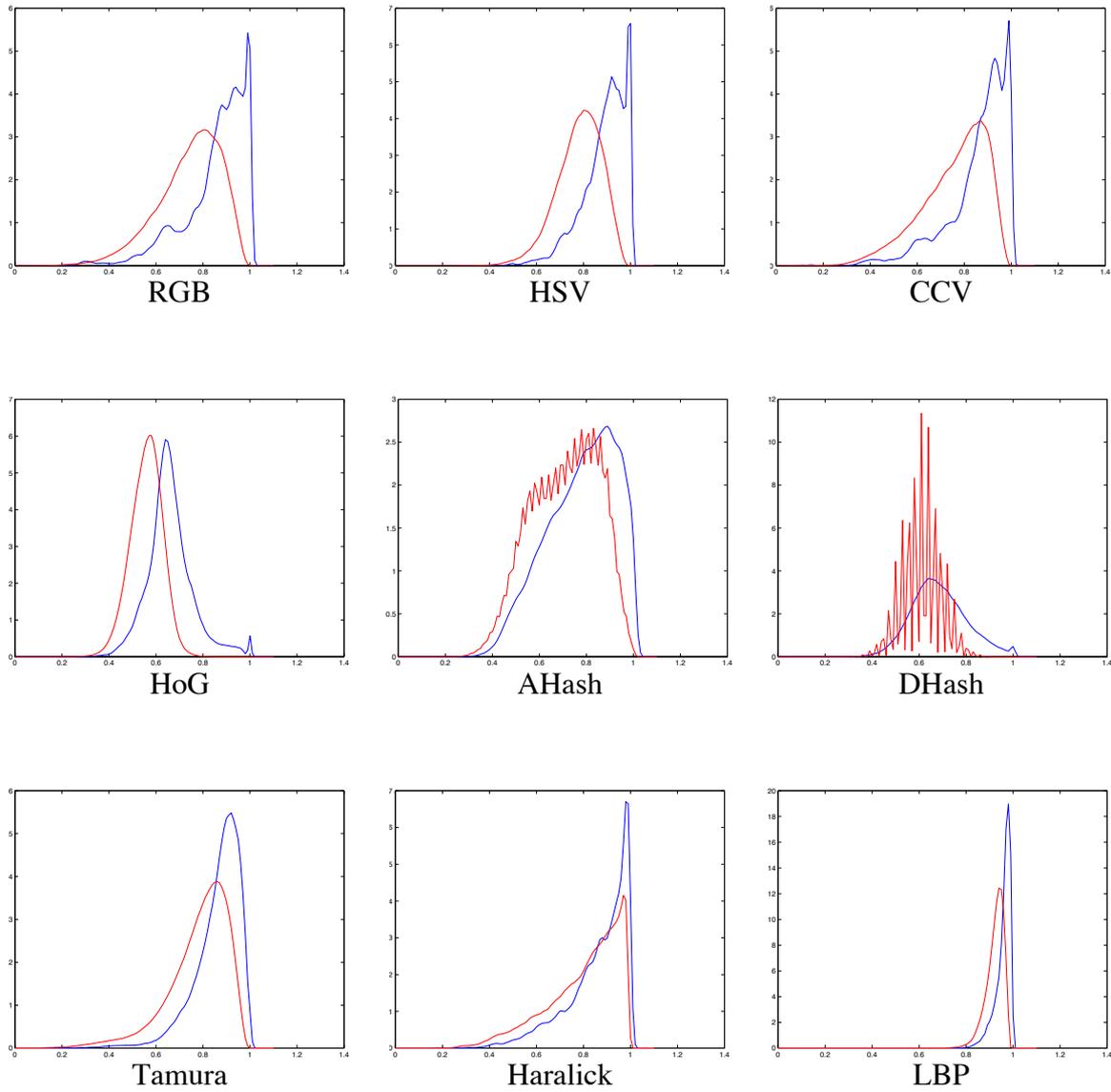


Figura 42: Grafici Genuine-Impostor per il dataset Sheffield Building.

Anche i Genuine-Impostor evidenziano che il descrittore HoG fornisce una separazione delle due gaussiane leggermente più accentuata.

4.3.4 Object ZuBud

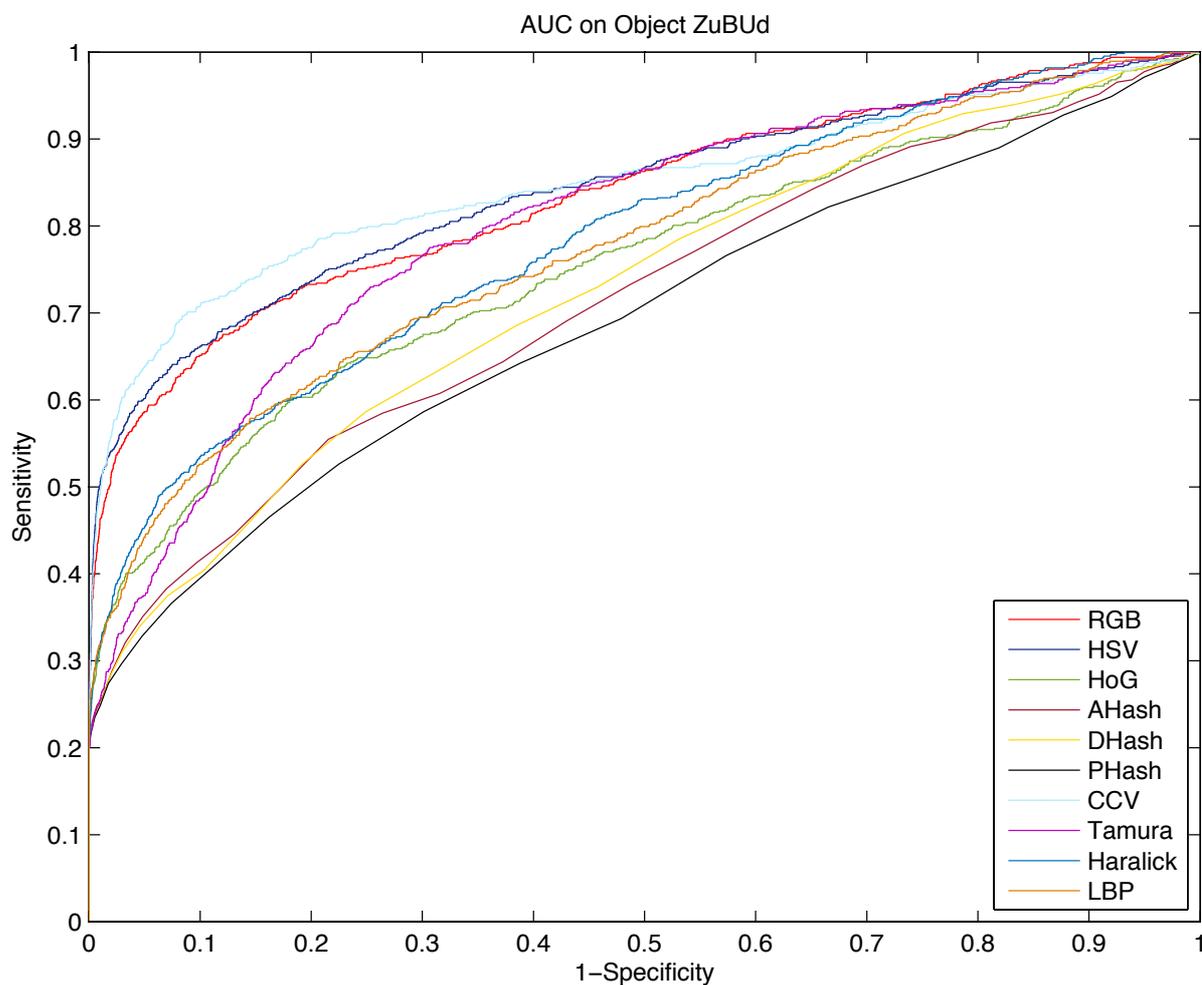


Figura 43: Rappresentazione delle curve ROC ottenute sul dataset Object ZuBud.

Il dataset contenente solo immagini di oggetti ha mostrato un buon comportamento di CCV, seguito da RGB, HSV e per quanto riguarda la tessitura il descrittore Tamura.

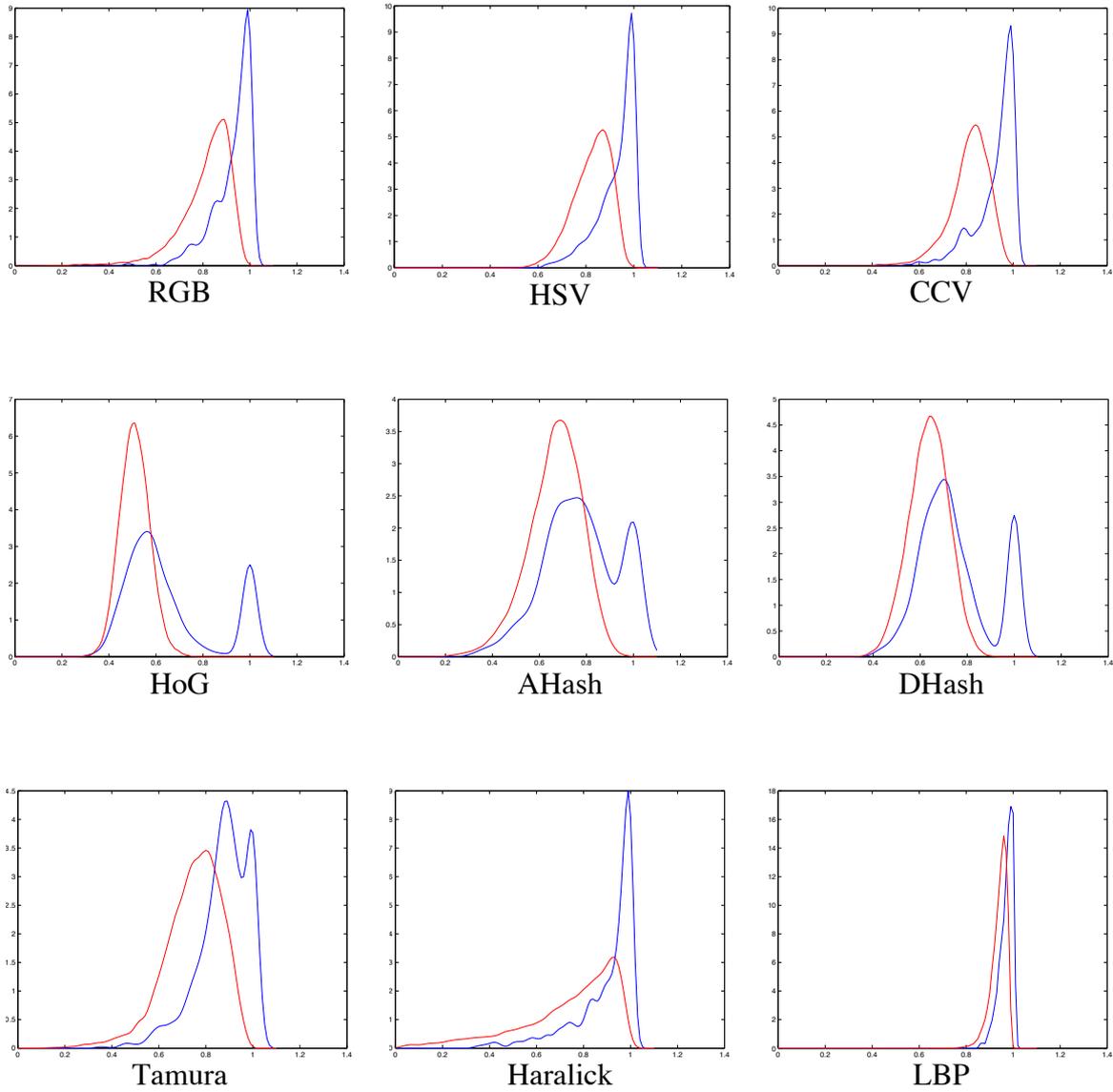


Figura 44: Grafici Genuine-Impostor ottenuti sul dataset Object ZuBud.

Anche osservando i Genuine-Impostor è evidente che la miglior separazione si è ottenuta con il descrittore CCV e la misura Chi-Square Distance.

4.3.5 WangDB

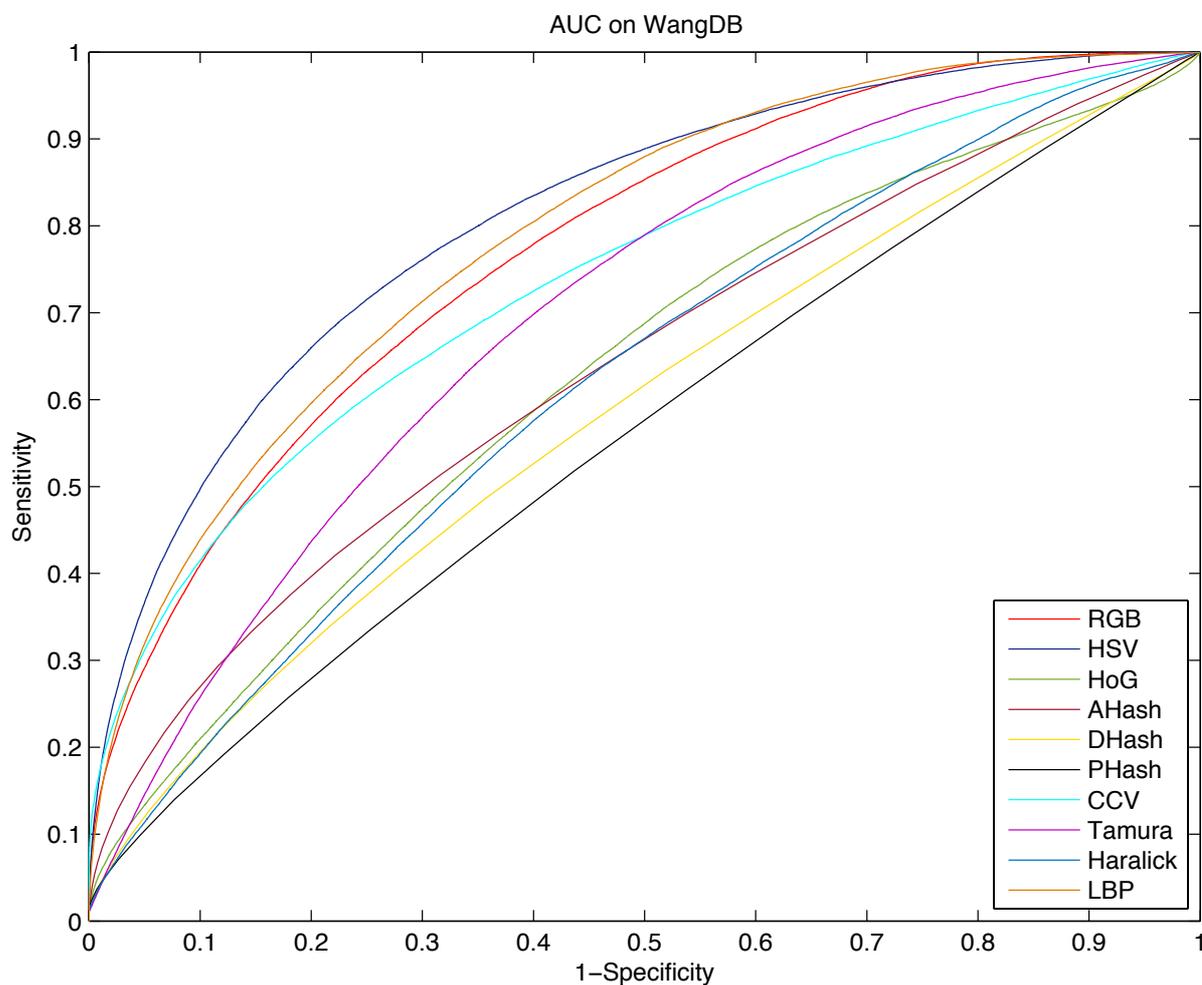


Figura 45. Rappresentazione delle curve ROC ottenute sul dataset WangDB.

Le curve ROC sul dataset Wang vedono nuovamente come miglior descrittore HSV seguito da RGB, LBP e CCV praticamente alla pari.

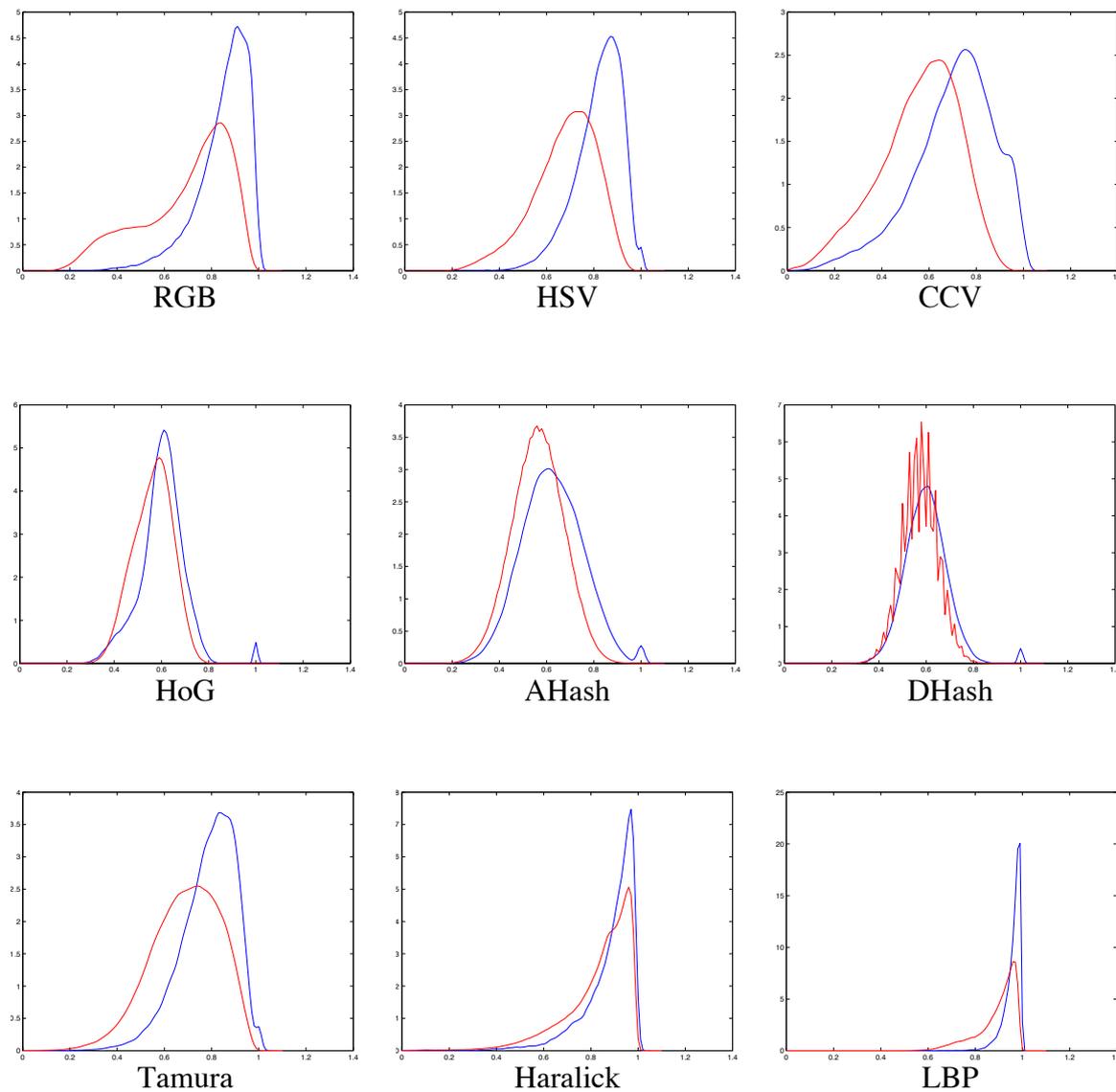


Figura 46: Grafici Genuine-Impostor ottenuti sul dataset Wang.

I risultati non molto alti visibili sulle curve ROC sono confermati da grafici Genuine-Impostor dove la sovrapposizione delle gaussiane è molto elevata.

4.3.6 ItaLa

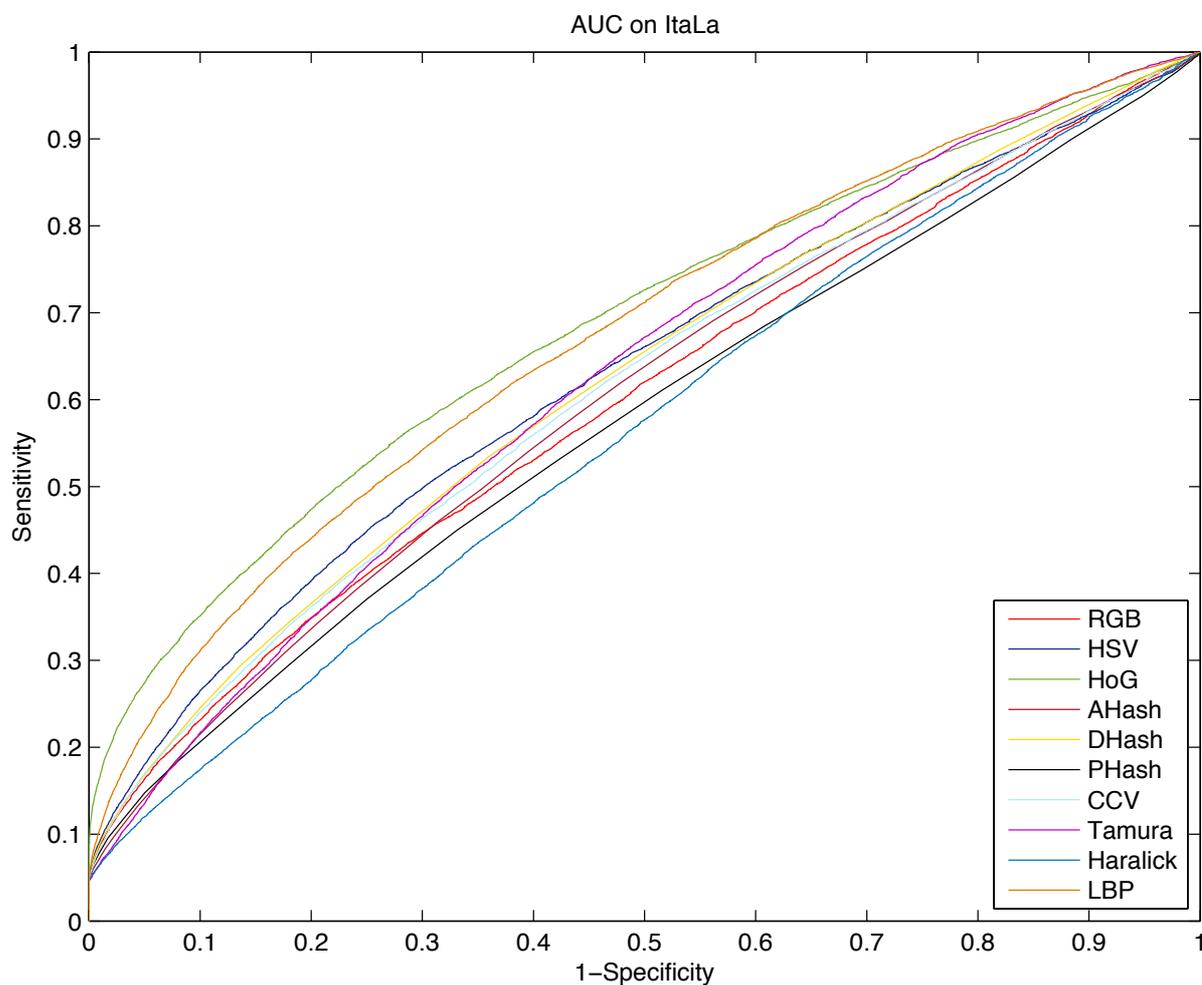


Figura 47: Rappresentazione delle curve ROC ottenute sul dataset ItaLa.

L'abbassamento generale delle curve suggerisce una difficoltà maggiore riscontrata nell'analizzare il dataset ItaLa, dove comunque la varietà di immagini e la differenza di circostanze temporali nel quale sono state ricavate porta il descrittore di forma HoG ad avere il risultato migliore, seguito non di molto distante da LBP.

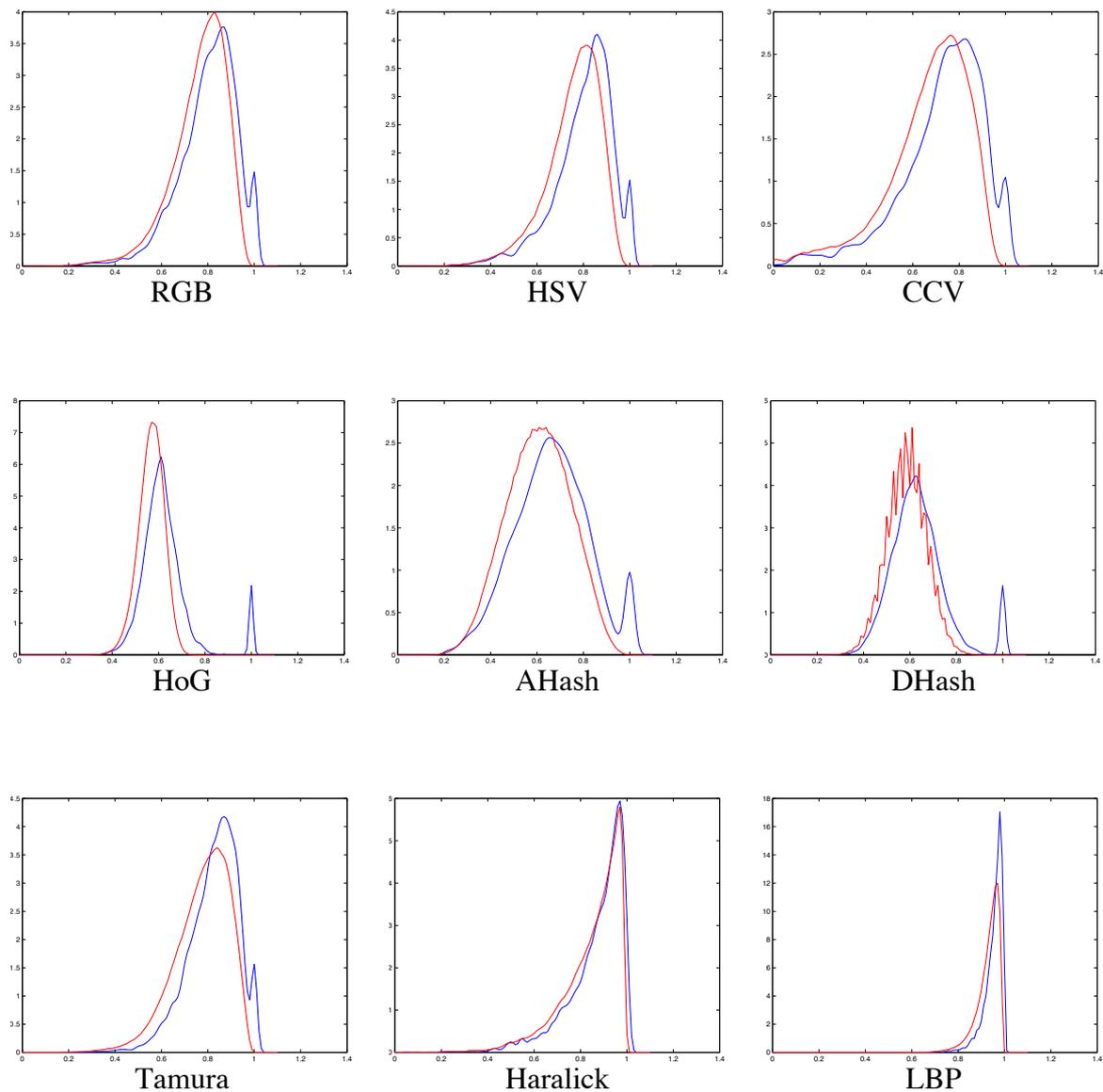


Figura 48: Grafici Genuine-Impostor ottenuti sul dataset ItaLa,

La quasi totale sovrapposizione delle gaussiane generata da tutti gli algoritmi è in linea con ciò che abbiamo osservato in precedenza, evidenziando nuovamente la netta difficoltà di questo dataset.

4.3.7 ColorBrodatz

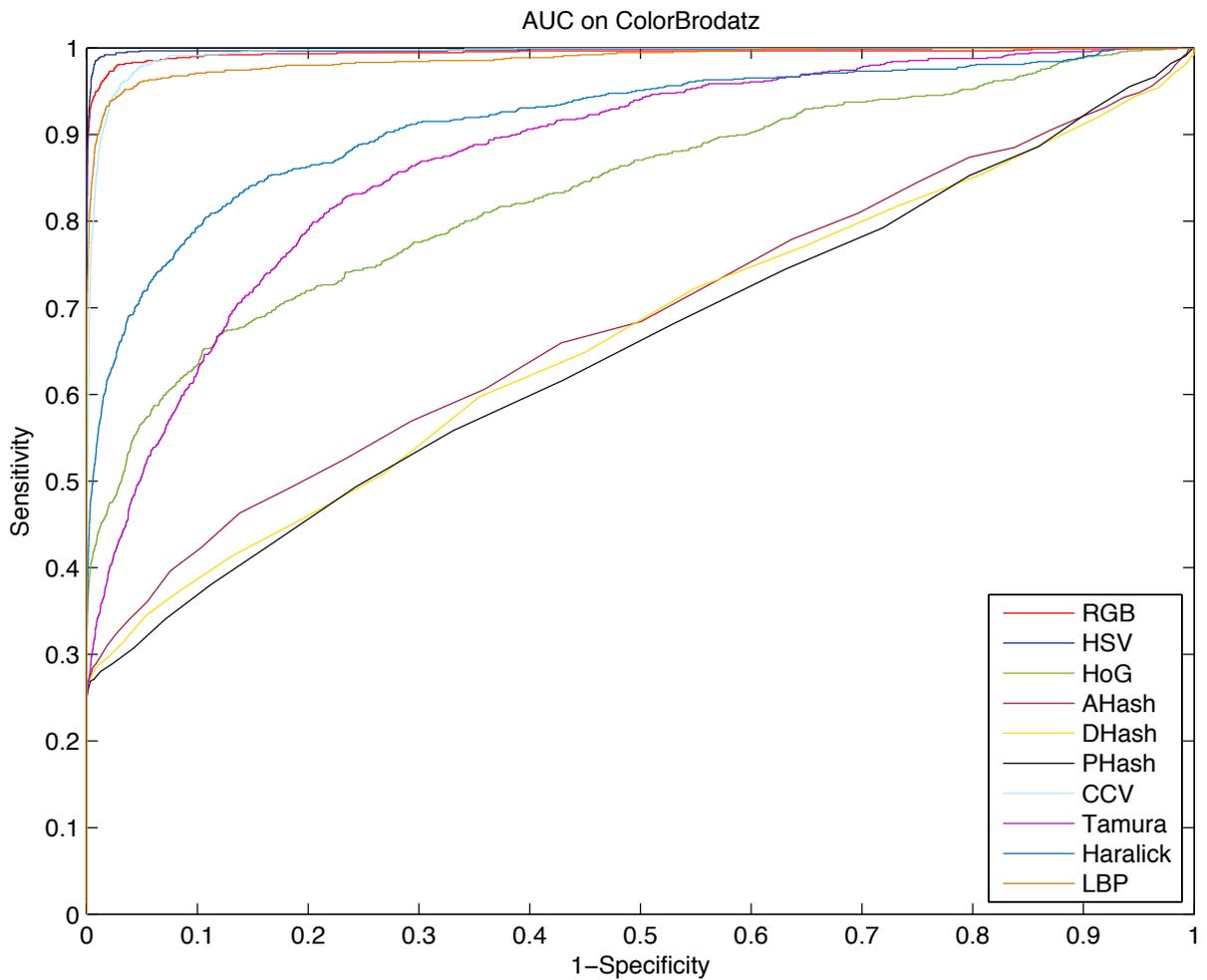


Figura 49: Rappresentazione delle curve ROC ottenute sul dataset ColorBrodatz.

L'unico dataset di tessitura testato mostra risultati ovviamente formidabili per i descrittori di colore a causa delle caratteristiche intrinseche delle immagini. Importante osservare come tra i descrittori di tessitura LBP sia risultato in questo caso nettamente migliore di Tamura e Haralick.

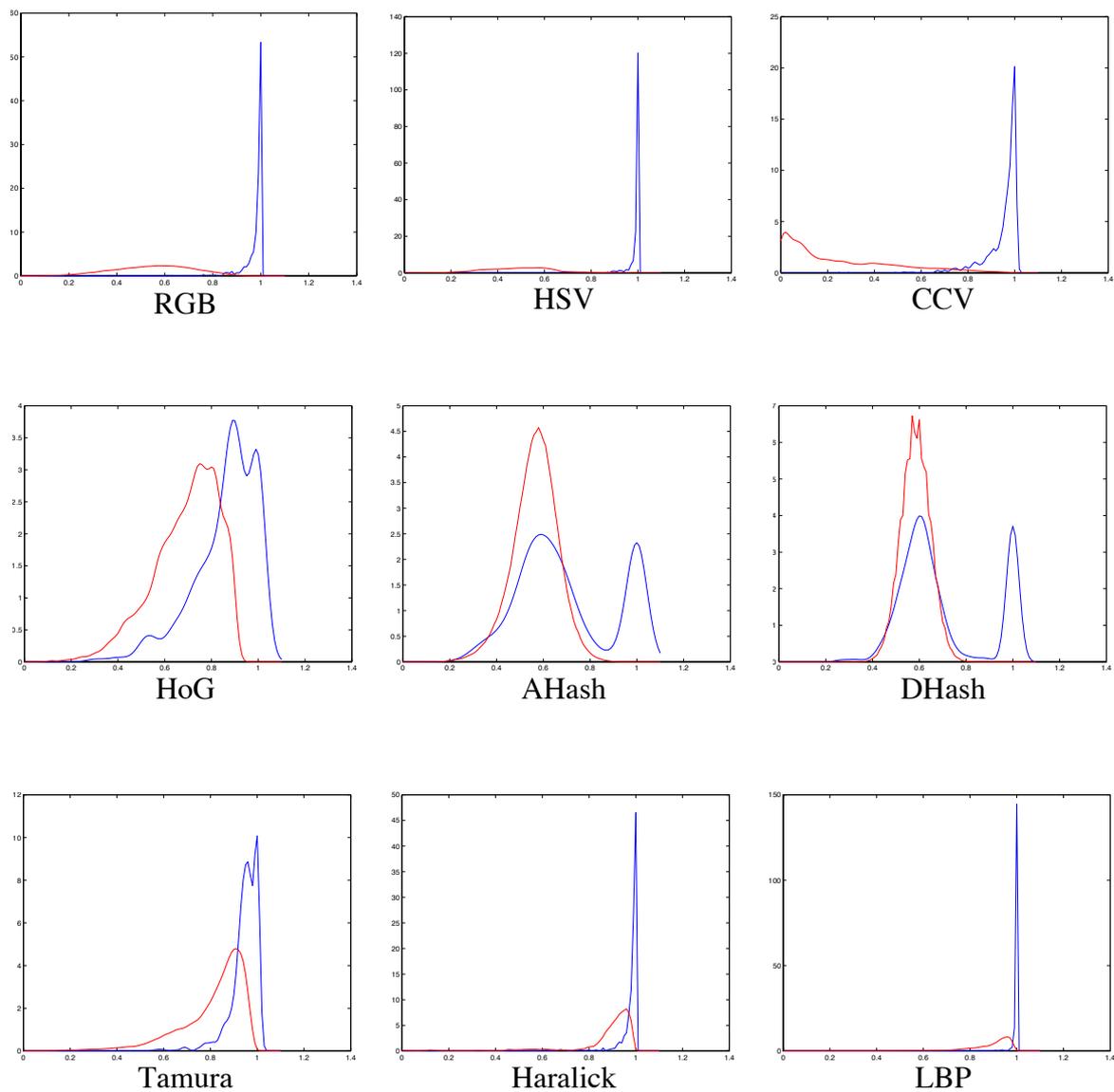


Figura 50: Grafici Genuine-Impostor ottenuti sul dataset ColorBrodatz.

Tralasciando il quasi perfetto comportamento di RGB e HSV, osserviamo anche tra i Genuine-Impostor di tessitura come indubbiamente LBP abbia dato i risultati migliori.

4.3.8 Risultati del metodo 3×3

Come introdotto precedentemente, uno degli aspetti implementativi affrontati ha portato alla realizzazione della suddivisione delle immagini in 9 celle e successiva estrazione dei descrittori dalle singole. In questo paragrafo vengono riportati i risultati dati dall'applicazione della divisione dell'immagine in una griglia 3×3, confrontati con i dati che avevamo ottenuto tramite il normale descrittore. Ogni estensione è stata applicata ad ognuno dei dataset per i descrittori RGB e HSV.

AUC	Descrittore			
Dataset	RGB	RGB3×3	HSV	HSV3×3
Caltech Buildings	0.8768	0.9139	0.9323	0.9525
ZuBud	0.9425	0.9434	0.9679	0.9641
Shieffield	0.7577	0.7769	0.8096	0.8156
ZuBud Object	0.8354	0.8024	0.8415	0.8328
Wang	0.7742	0.7697	0.8119	0.8184
ItaLa	0.5983	0.6320	0.6280	0.6612
ColorBrodatz	0.9942	0.9872	0.9978	0.9960

Tabella 4: Confronto tra l'AUC ottenuta con metodi locali e globali (RGB, HSV) sui dataset.

In Tabella 4 sono stati evidenziati i risultati dei quali si è scelto di riportare ulteriori grafici per mostrare anche graficamente l'effetto dell'applicazione del 3×3. Di seguito è quindi possibile osservare i miglioramenti su Caltech Buildings, ItaLa e Object ZuBud, anche se su quest'ultimo i miglioramenti si sono visti solo per HSV.

Nei grafici sottostanti si può osservare come avendo suddiviso l'immagine in 9 sottoimmagini si è ottenuto un sensibile miglioramento dell'algoritmo di partenza sia nel caso del descrittore RGB che in quello HSV. In particolare il metodo HSV3×3 ha fornito un valore di AUC di **0.9525** mentre HSV di **0.9323**.

Con RGB siamo passati invece da **0.8768** ad un valore di AUC con RGB3×3 di **0.9139**.

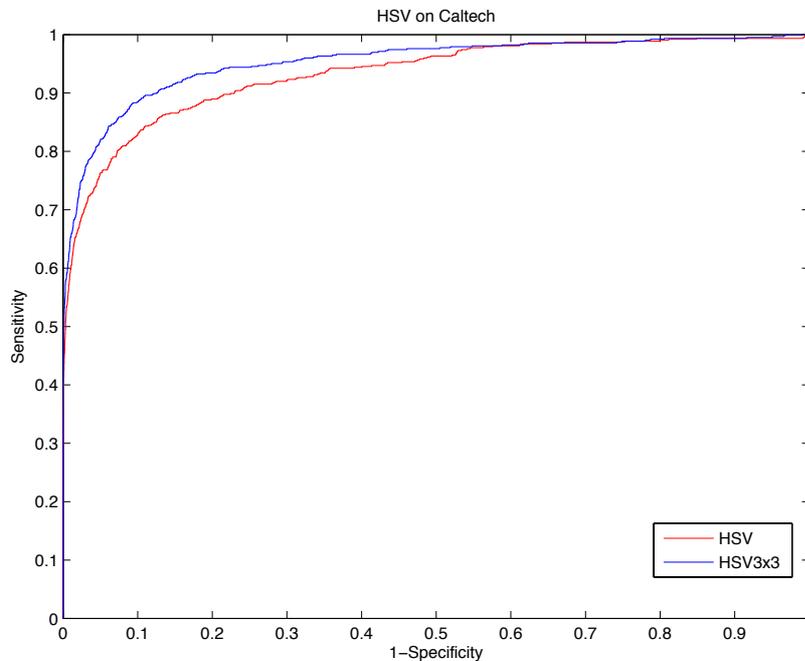


Figura 51: Confronto della curva ROC tra HSV e HSV3×3 su Caltech Buildings.

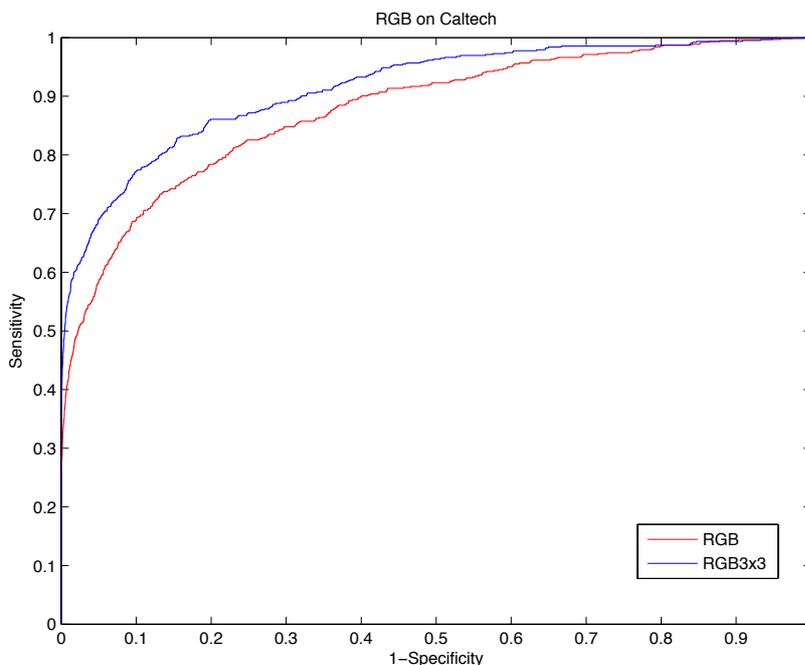


Figura 52: Confronto della curva ROC tra RGB e RGB3×3 su Caltech Buildings.

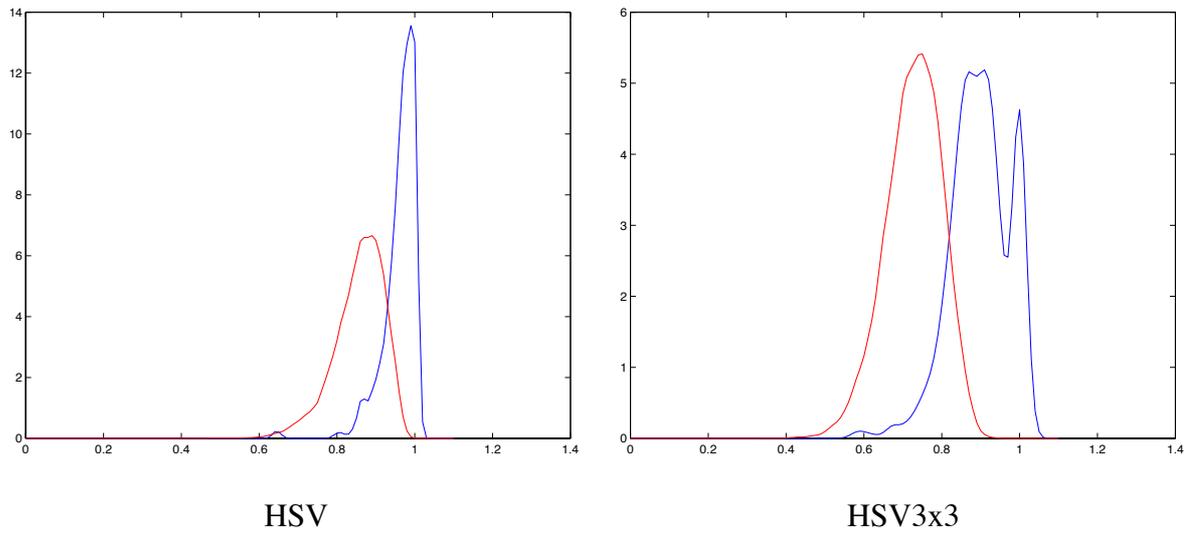


Figura 53: Confronto dei Genuine-Impostor tra HSV e HSV3x3 nel Caltech Buildings.

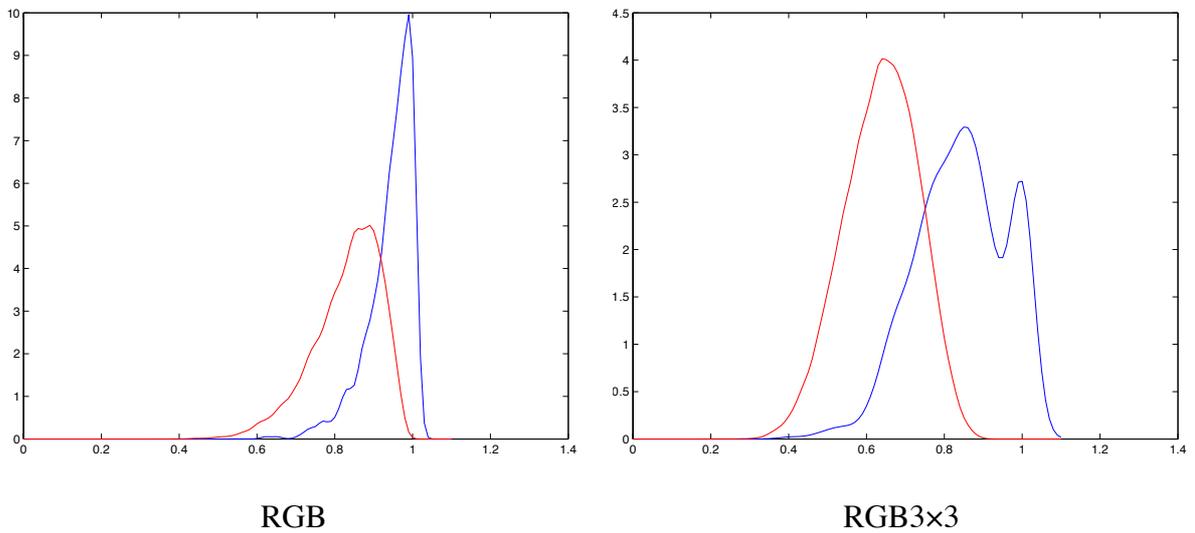


Figura 54: Confronto dei Genuine-Impostor tra RGB e RGB3x3 nel Caltech Buildings.

Per quanto riguarda l'Accuracy, nel primo caso relativo ad HSV si è passati da **66.40%** con il metodo normale a **78.00%** utilizzando l'HSV 3x3. Nel secondo caso invece il descrittore RGB è migliorato passando da **42.40%** (metodo normale) a **64.40%** con il metodo 3x3.

Sullo Shieffield un sensibile miglioramento si è visto sullo spazio RGB.

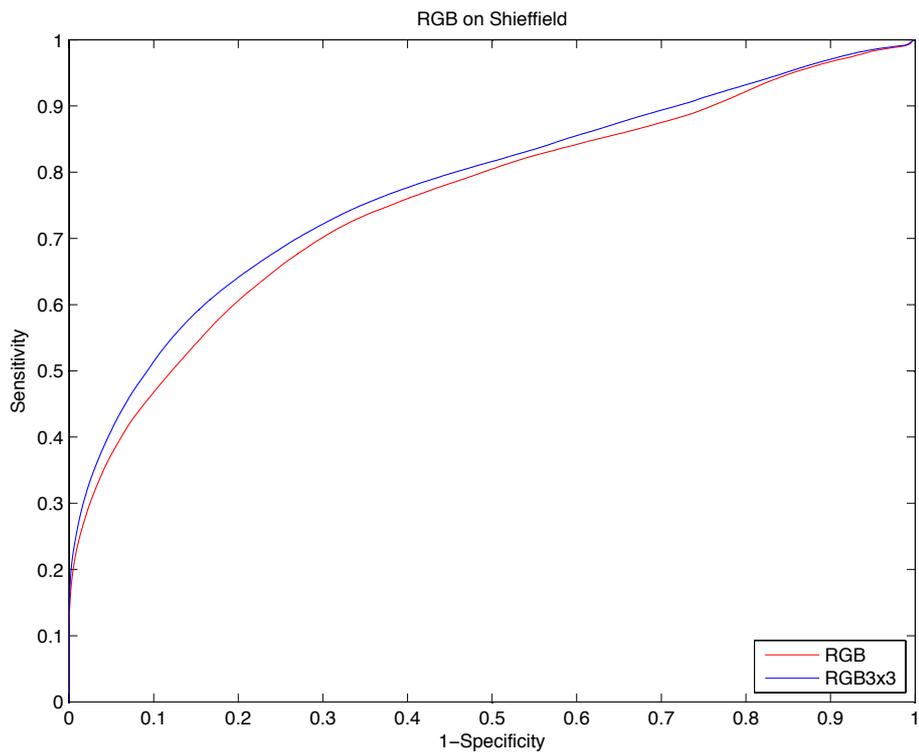


Figura 55: Confronto della curva ROC tra RGB e RGB3x3 su Shieffield.

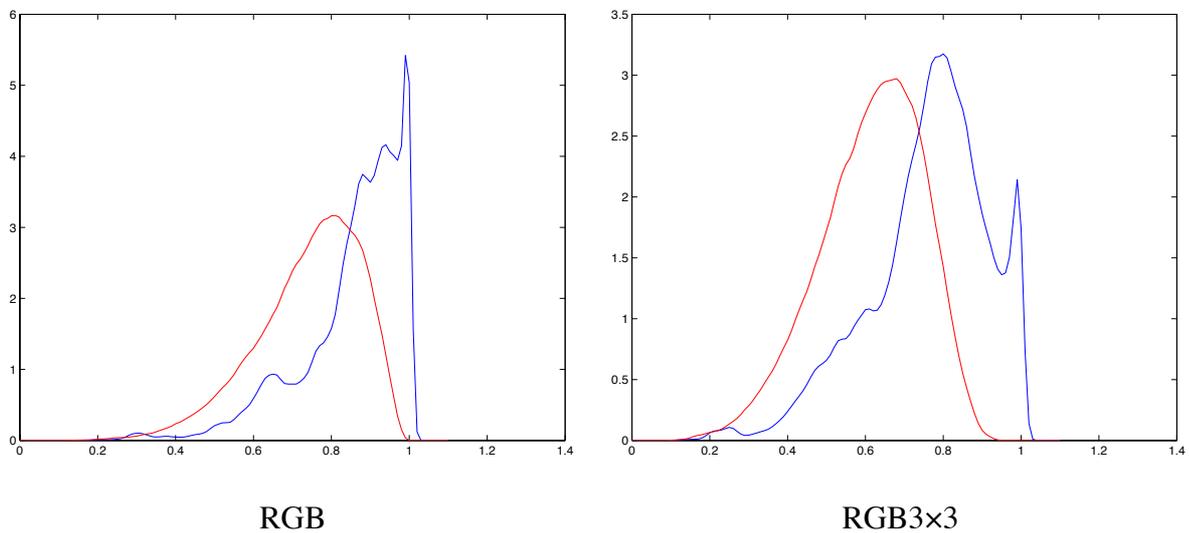


Figura 56: Confronto dei Genuine-Impostor tra RGB e RGB3x3 sullo Shieffield.

In questo caso l'Accuracy è passata da un valore di **94.95%** a **97.92%**.

Su Object Zubud possiamo invece osservare un peggioramento dei metodi HSV e RGB.

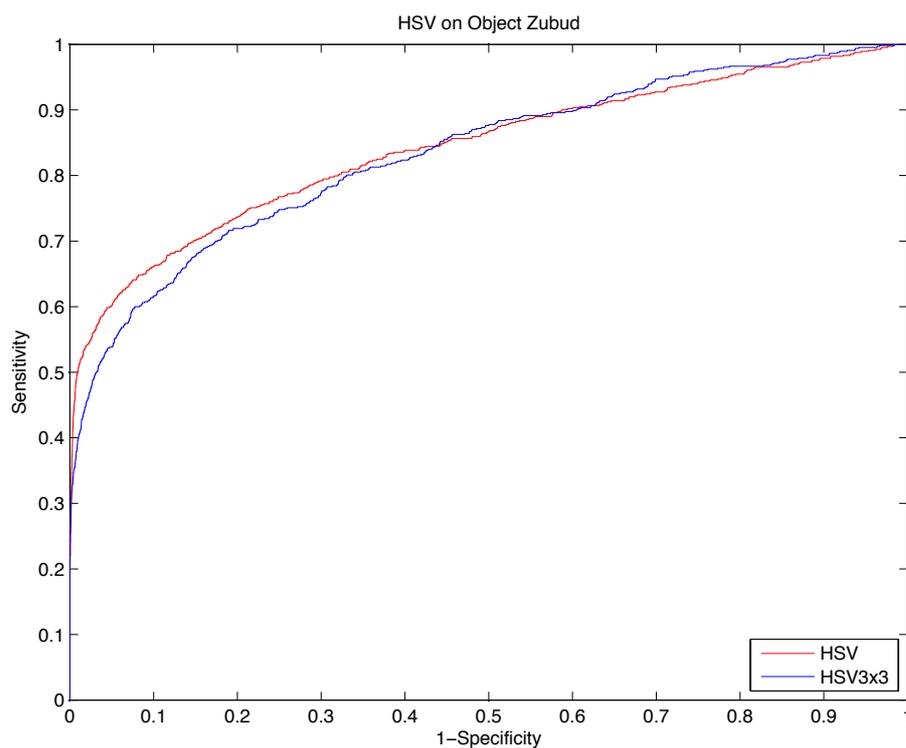


Figura 57: Confronto delle curve ROC ottenute con HSV e HSV3x3 su Object ZuBud.

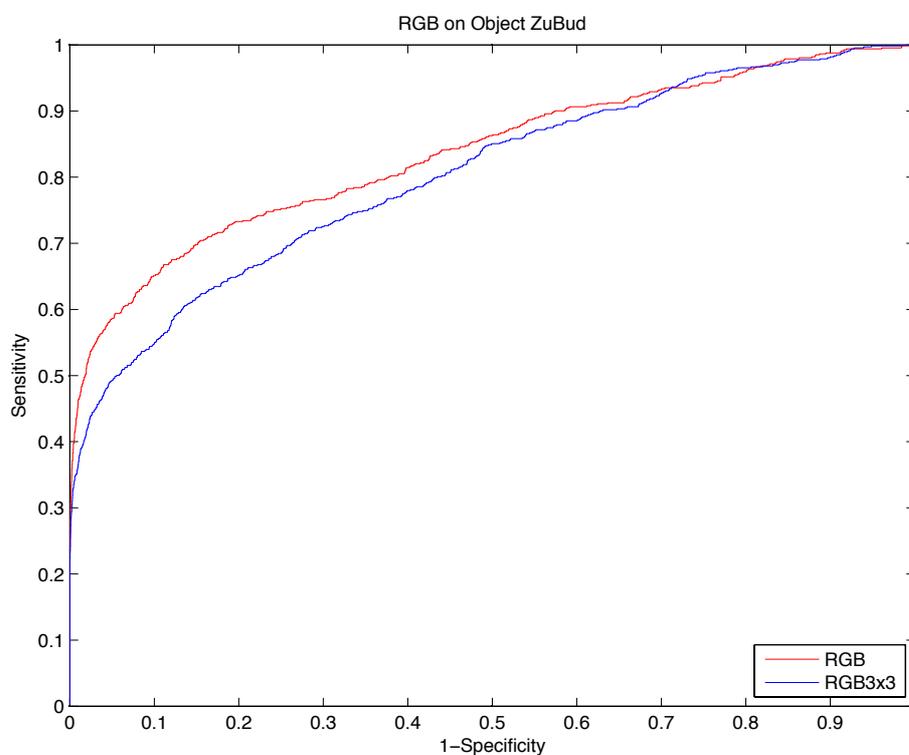


Figura 58: Confronto delle curve ROC ottenute con RGB e RGB3x3 su Object ZuBud.

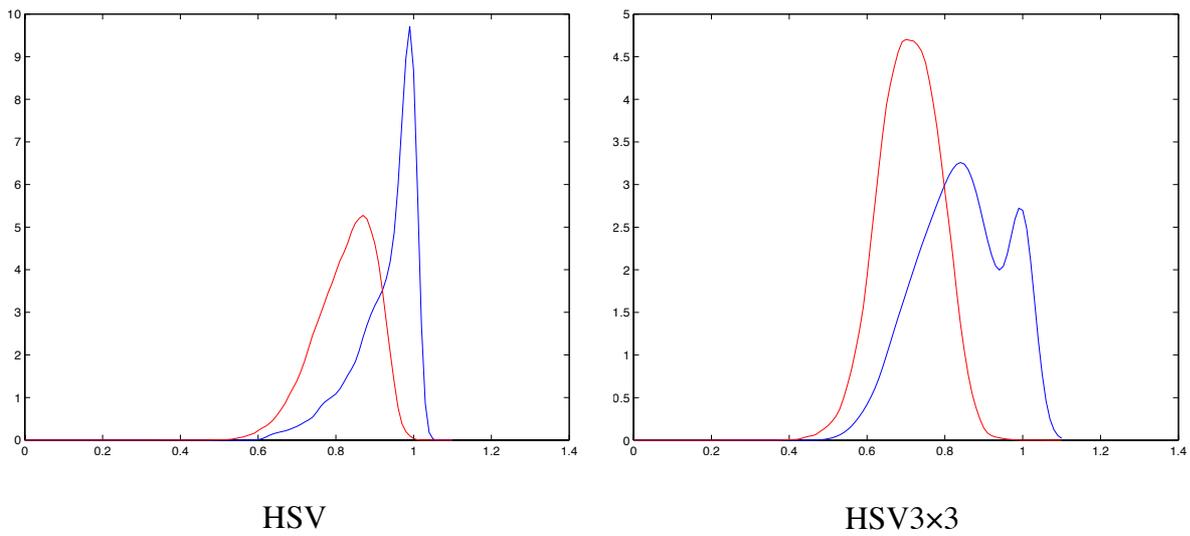


Figura 59: Confronto dei Genuine-Impostor tra HSV e HSV3x3 su Object ZuBud.

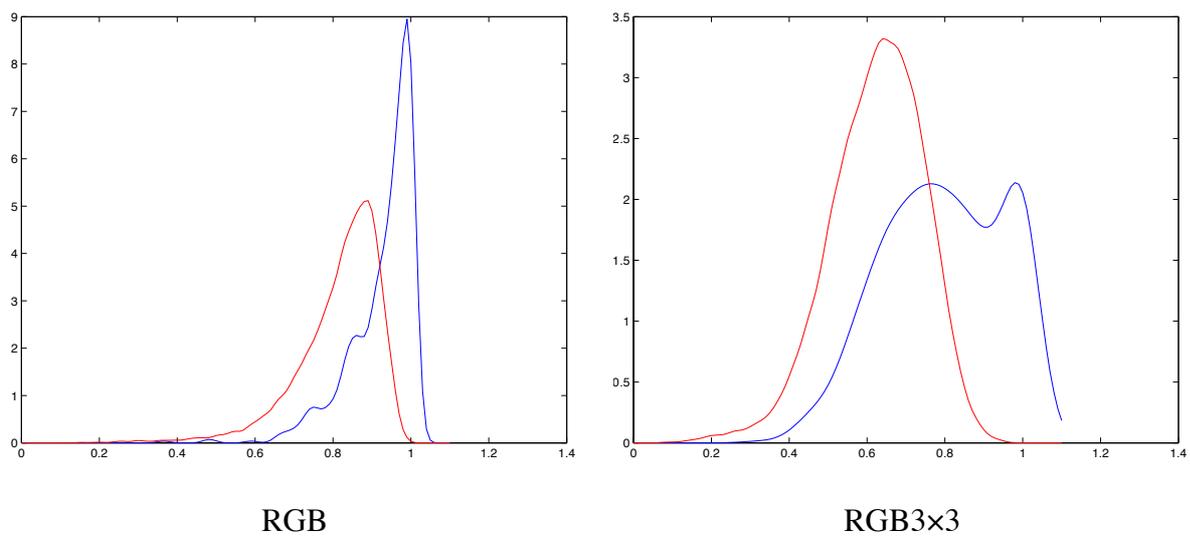


Figura 60: Confronto dei Genuine-Impostor tra RGB e RGB3x3 su Object ZuBud.

L'Accuracy è passata nel caso di HSV dal **57.74%** al **44.91%** mentre con RGB è scesa da **52.45%** a **33.21%**. Il netto peggioramento è anche visibile dalla minore separazione che si è ottenuta sulle curve Genuine-Impostor.

Anche ItaLa ha mostrato discreti miglioramenti grazie al nuovo metodo sia per quanto riguarda HSV che RGB.

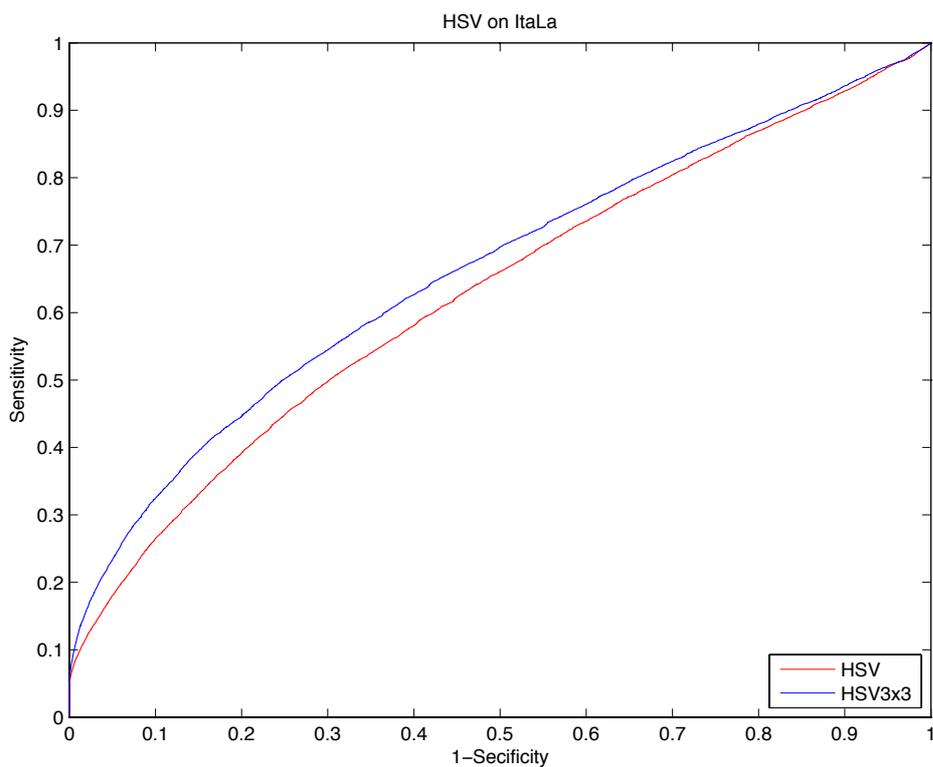


Figura 61: Confronto delle curve ROC ottenute con HSV e HSV3x3 su ItaLa.

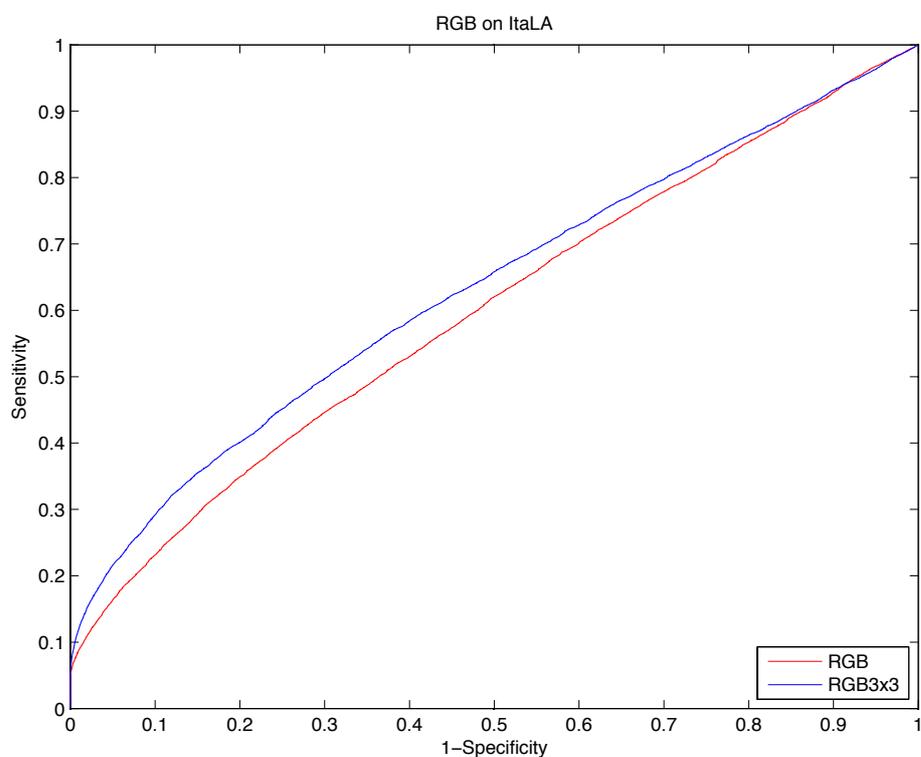


Figura 62: Confronto delle curve ROC ottenute con RGB e RGB3x3 su ItaLa.

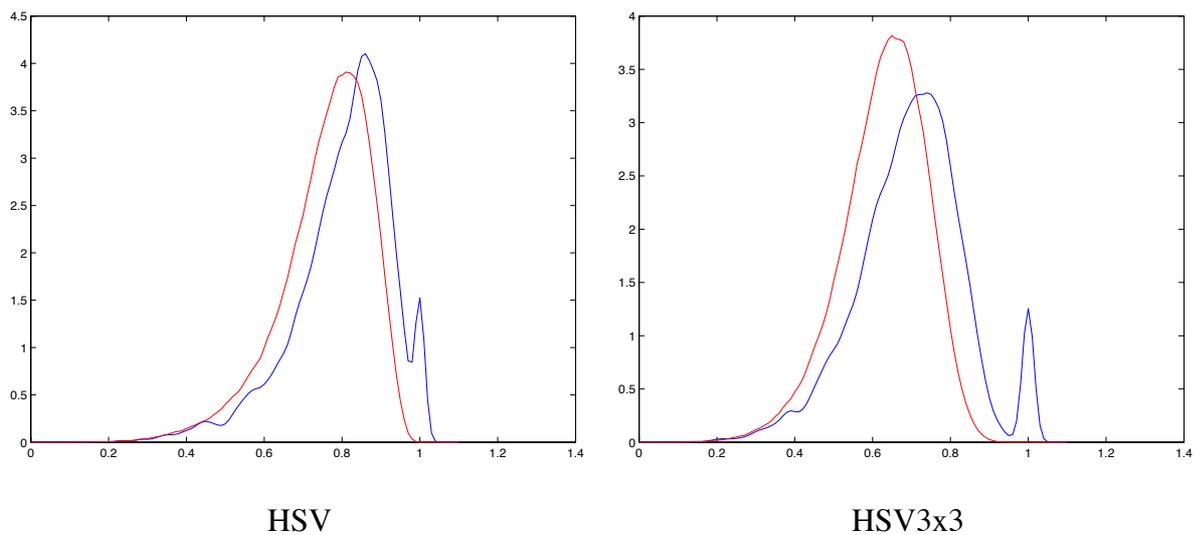


Figura 63: Confronto dei Genuine-Impostor tra HSV e HSV3x3 su ItaLa.

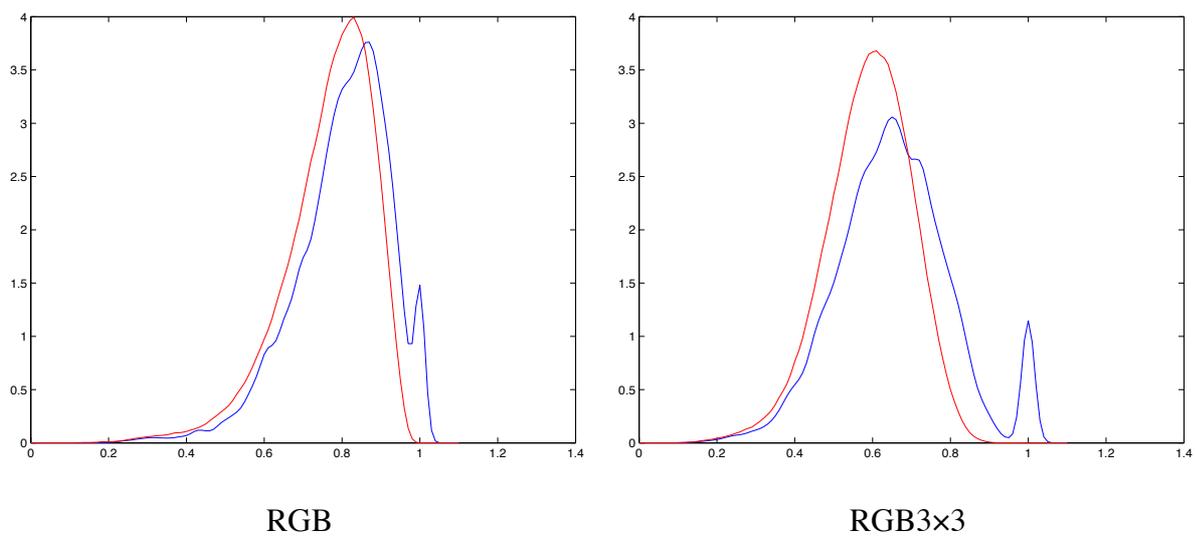


Figura 64: Confronto dei Genuine-Impostor tra RGB e RGB3x3 su ItaLa.

Nel caso HSV l'Accuracy è passata da **31.14%** a **44.79%**, mentre con RGB siamo passati da un'Accuracy di **30.79%** a **30.79%**.

4.4 Valutazioni dei dati sperimentali

Analizzando i dati rilevati, confrontando i grafici ottenuti e tenendo in considerazione le caratteristiche dei singoli dataset sottoposti ai test, possiamo affermare che nei casi che vedono l'analisi di immagini appartenenti a edifici, ritratti nello stesso momento senza quindi nessun tipo di variazioni di luce (Caltech Buildings, ZuBud), HSV risulta il miglior descrittore mentre, come già affermato in precedenza, esso risulta poco robusto per problemi nei quali abbiamo a che fare con cambi di luminosità, contrasto e tono di colore, nonostante ciò rimane comunque un ottimo metodo di classificazione in quanto si è dimostrato adatto anche a identificare foto non solo di edifici ma anche di oggetti, anche se in quest'ultima situazione è probabilmente più consigliato l'utilizzo del CCV.

Nei casi di utilizzo dei dataset con forti variazioni di luce date dal fatto di ritrarre lo stesso soggetto in momenti della giornata diverse ed in situazioni climatiche differenti (Sheffield Buildings, ItaLa) il comportamento migliore, seppur non contraddistinto da risultati elevati, è dato dal descrittore di forma HoG.

Per quanto riguarda i descrittori di tessitura implementati, ossia Haralick, Tamura e LBP. Proprio LBP risulta in tutti i casi esaminati il miglior algoritmo di estrazione di feature associato alla distanza Chi-Square distance. I metodi PHash, DHash e AHash hanno mostrato bassi risultati collegabili possibilmente alla grossissima riduzione delle immagine e all'esclusione dei dettagli e delle altre frequenze.

Al fine di migliorare ulteriormente il framework utilizzato in questo elaborato ed ampliare le possibilità di test e valutazione oltre all'implementazione di nuovi descrittori e nuove distanze, si può pensare di parametrizzare la costruzione degli istogrammi stessi. Nei test qui presentati si è scelto di mantenere valori statici per la dimensione dei bucket dell'istogramma: per il canale della tinta si è suddiviso il range dei valori in 16 bucket, per la saturazione e per la luminosità 4 bucket; si è fatta questa scelta in quanto era nostra intenzione dare una maggior rilevanza alla tinta piuttosto che alla luminosità o alla saturazione in quanto è il canale che, al variare dell'illuminazione e contrasto, rimane pressoché costante.

I miglioramenti raggiunti invece con l'implementazione della suddivisione dell'immagine in nove celle può essere un buon punto di partenza per ampliare i test parametrizzando il numero di celle sul quale applicare i descrittori.

Capitolo 5

Conclusioni

Lo scopo di questa tesi è rappresentato dall'analisi dei sistemi Content Based Image Retrieval. Questo campo dell'information retrieval è tutt'oggi argomento di ricerca ed il suo studio risulta appassionante non solo nell'analizzare articoli ma anche nell'implementare, nonostante le difficoltà, nuovi algoritmi e metodi che possano essere fonte di test o la base per applicazioni utili e di interesse comune.

Al fine di soddisfare gli obiettivi prefissati, ho analizzato alcuni dei metodi più conosciuti e descritti negli articoli che ho analizzato, al fine di apprenderne i funzionamenti e le caratteristiche che differenziano gli uni dagli altri, in seguito ho contribuito ad aumentare le capacità del framework a nostra disposizione inserendo nuovi descrittori tra cui LBP, aggiungendo il metodo per la divisione di un'immagine in 9 celle dalle quali ricavare ulteriori vettori di feature e implementando le funzioni per ricavare i grafici relativi alle curve ROC, molto utili per un ulteriore confronto dei risultati e una valutazione attenta dei dati numerici che sono stati convogliati in un primo momento all'interno di tabelle. Successivamente ho elaborato tali valori sotto forma di grafici che mi hanno fornito un'ulteriore verifica del mio lavoro.

In definitiva, ho raggiunto gli scopi prefissati, inoltre come già affermato in precedenza il lavoro eseguito potrà essere ancora ampliato in possibili sviluppi futuri, introducendo nuovi metodi, in particolare su forma e punti salienti, nuove misure di similarità e nuovi dataset con caratteristiche differenti per estendere il campo di test e affinare le decisioni su quali coppie di descrittori e misure di distanza possono favorire un determinato problema CBIR.

Bibliografia

- [1] Huang, Y. Rui and T. S., «image retrieval: Current techniques, promising directions and open issues,» *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39-62, 1999.
- [2] Flickner M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, «Query by image and video content: the QBIC system,» *Computer*, vol. 28, n. 9, pp. 23-32, 1995.
- [3] Mylonas, P., Spyrou, E., Avrithis, Y., Kollias, S., «Using Visual Context and Region Semantics for High-Level Concept Detection,» *IEEE Transactions on Multimedia*, vol. 11, n. 2, pp. 229-243, 2009.
- [4] Eakins J., M. Graham, J. Eakins, M. Graham, and T. Franklin, «Content-based image retrieval,» *Library and Information Briefings*, vol. 85, pp. 1-15, 1999.
- [5] Miguel Arevalillo-Herráeza, Mario Zacarésb, Xaro Benaventc, Esther de Vesa, «A relevance feedback CBIR algorithm based on fuzzy sets,» *Signal Processing: Image Communication*, vol. 23, pp. 490-504, 2008.
- [6] Otávio A.B. Penattia, Eduardo Vallea, Ricardo da S. Torresa, «Comparative study of global color and texture descriptors for web image retrieval,» *Journal of Visual Communication and Image Representation*, vol. 23, pp. 359-380, 2012.
- [7] Nanni L., A. Lumini, S. Brahmam, «Ensemble of different local descriptors, codebook generation methods and subwindow configurations for building a reliable computer vision system,» *Journal of King Saud University*, pp. 1-5, 2014.
- [8] Joshi K.D., S. N. Bhavsar, R. C. Sanghvia, «Image Retrieval System using Intuitive Descriptors,» *Elseiver*, pp. 535-542, 2014.
- [9] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R., «Content-based image retrieval at the end of the early years,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n. 12, p. 1349–1380, 2000.

- [10] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., & Petkovic, D., et al., «Efficient and effective querying by image content,» *Journal of Intelligent Information Systems*, vol. 3, p. 231–262, 1994.
- [11] Deselaers, T., «Features for image retrieval. Master's thesis. Aachen: Human Language Technology and Pattern Recognition Group,» *RWTH Aachen University*, 2003.
- [12] Puzicha, J., Rubner, Y., Tomasi, C., & Buhmann, J., «Empirical evaluation of dissimilarity measures for color and texture,» *In International Conference on Computer Vision*, vol. 2, p. 1165–1173, 1999.
- [13] Swain, M. J., & Ballard, D. H., «Color indexing,» *International Journal of Computer Vision*, vol. 7, n. 1, pp. 11-32, 1991.
- [14] Smith, J. R., & Chang, S. F., «Tools and techniques for color image retrieval,» *In SPIE Storage and Retrieval for Image and Video Databases*, vol. 2670, p. 426–437, 1996.
- [15] Greg Pass, Ramin Zabih, Justin Miller, «Comparing images using Color Coherence Vectors,» in *Proceedings of the fourth ACM international conference on Multimedia*, New York, 1996.
- [16] Vailaya, A., Figueiredo, M. A. T., Jain, A. K., & Zhang, H. -J., «Image classification for content-based indexing,» *IEEE Transactions on Image Processing*, vol. 10, n. 1, p. 117–130, 2001.
- [17] Antani, S., Kasturi, R., & Jain, R., «A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video,» *Pattern Recognition*, vol. 35, p. 945–965, 2002.
- [18] Pentland, A., Picard, R., & Sclaroff, S., «Photobook: Content-based manipulation of image databases,» *International Journal of Computer Vision*, vol. 18, n. 3, p. 233–254., 1996.
- [19] Deselaers, T., Rybach, D., Dreuw, P., Keyers, D., & Ney, H., «Face-based image retrieval-one step toward object-based image retrieval,» in *H. Müller & A. Hanbury (Eds.), MUSCLE/ImageCLEF Workshop on Image and Video Retrieval Evaluation*, Vienna, 2005.

- [20] Alberto Del Bimbo, Visual information retrieval, San Fransisco, CA: Morgan Kaufmann Publishers, 1999.
- [21] Tuceryan, M. & Jain, A.K., Texture analysis. In: Chec, C.H., Pau, L.F., and Wang, P.S.P., editors, Handbook of Pattern Recognition and Computer Vision, River Edge, NJ, USA, 1993, pp. 235-276.
- [22] Haralick, R. M., Shanmugam, B., & Dinstein, «Texture Features for image classification,» *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, n. 6, p. 610–621, 1973.
- [23] Gu, Z. Q., Duncan, C. N., Renshaw, E., Mugglestone, M. A., Cowan, C. F. N., & Grant, P. M., «Comparison of techniques for measuring cloud texture in remotely sensed satellite meteorological image data.,» *Radar and Signal Processing*, vol. 136, n. 5, p. 236–248, 1989.
- [24] T. M. G. M. -O. D. T. K. D. S. H. S. K. e. a. Lehmann, «Automatic categorization of medical images for content-based retrieval and data mining,» *Computerized Medical Imaging and Graphics*, vol. 29, n. 2, p. 143–155, 2005.
- [25] Tamura, H., Mori, S., & Yamawaki, T., «Textural features corresponding to visual perception,» *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 8, n. 6, p. 460–472, 1978.
- [26] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., & Petkovic, D., et al., «Efficient and effective querying by image content,» *Journal of Intelligent Information Systems*, vol. 3, p. 231–262, 1994.
- [27] Rekhil M Kumar et al, «A Survey on Image Feature Descriptors,» *International Journal of Computer Science and Information Technologies*, vol. 5, n. 6, pp. 7668-7673, 2014.
- [28] Park, M., Jin, J. S., & Wilson, L. S., «Fast content-based image retrieval using quasi-gabor filter and reduction of image feature,» in *Southwest Symposium on Image Analysis and Interpretation*, Santa Fe, 2002.
- [29] Squire, D. M., Müller, W., Müller, H., & Raki, J., «Content-Based query of image databases, inspirations from text retrieval: Inverted files, frequency-based weights and

relevance feedback,» in *Scandinavian Conference on Image Analysis*, Kangerlussuaq, 1999.

[30] Ojansivu, V., and Heikkila, J., «Blur insensitive texture classification using local phase quantization,» *ICISP*, 2008.

[31] D. G. Lowe, «Distinctive image features from scale-invariant keypoints,» *International Journal of Computer Vision*, vol. 60, n. 2, p. 91–110, 2004.

[32] Bay H., Ess A., Tuytelaars T. and Van Gool L., «Speeded-up robust features(surf),» *Computer Vision Image Understanding*, vol. 110, pp. 349-359, 2008.

[33] John Collins, Kazunori Okada, «A Comparative Study of Similarity Measures for Content-Based Medical Image Retrieval,» *Medical CBIR application context*, pp. 4-8, 2012.

[34] Squire D.M., Wolfgang Müller, Henning Müller, Raki J., «Content-Based Query of Image Databases, Inspirations From Text Retrieval: Inverted Files, Frequency-Based Weights and Relevance Feedback,» in *The 11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, 1999.

Riferimenti ai dataset:

[35] Caltech Buildings: <http://www.vision.caltech.edu/malaa/datasets/caltech-buildings>;

[36] ZuBud: <http://www.vision.ee.ethz.ch/showroom/zubud>;

[37] Sheffield Buildings: <https://www.sheffield.ac.uk/eee/research/iel/research>;

[38] ZuBud Objects: <http://www.vision.ee.ethz.ch/showroom/zubud>;

[39] WangDB: <http://wang.ist.psu.edu/docs/related>;

[40] ItaLa: <http://bias.csr.unibo.it/lumini/download/dataset/ItaLa.zip>.

[41] ColorBrodatz: http://multibandtexture.recherche.usherbrooke.ca/colored%20_brodatz;

Ringraziamenti

Grazie di cuore alla professoressa Alessandra Lumini per il tempo dedicatomi, la disponibilità e i suggerimenti che hanno reso possibile la stesura di questa tesi.

Grazie ai compagni di università, per gli anni passati assieme a seguire giornate intere di lezioni spesso incomprensibili e a realizzare progetti dei quali non capivamo niente, senza dimenticare le notti in bianco per le consegne mattiniere delle relazioni. In particolare ringrazio Mancio e Lunghe, per le giornate di studio e pause accompagnate dalla condivisione dei nostri hobby, senza un po' di *provocazione*, qualche *arena* e una *sbustata* ogni tanto, passare elaborazione sarebbe stato molto più impegnativo.

Grazie alla Ninni, una persona fantastica, leale, sincera, un libro aperto che bisogna avere il coraggio di leggere, ma che in ogni pagina può insegnarti qualche cosa e farti diventare una persona migliore. Una centrale nucleare di determinazione e volontà, una donna splendida, bella ed anche una cuoca straordinaria alla quale auguro di realizzare ogni suo singolo sogno.

Grazie ai miei genitori, per avermi sempre appoggiato nelle mie scelte, per essermi sempre vicino, e per desiderare ogni giorno il meglio per me.

Infine ringrazio tutti i miei nonni, per avermi cresciuto e avermi insegnato ad essere la persona che sono, in particolare il nonno Giorgio, al quale mando un angelico abbraccio e il nonno Guglielmo, al quale dedico questo successo.