



**UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA**



**DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE**

**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE**  
**CORSO DI LAUREA IN INGEGNERIA INFORMATICA**

**“Feature per la classificazione del movimento artistico dei dipinti”**

**Relatore: Prof. Loris Nanni**

**Laureando: Riccardo Rosalen**

**ANNO ACCADEMICO 2020–2021**

**Data di laurea: 22/09/2021**



# Indice

<b>INTRODUZIONE.....</b>	<b>5</b>
<b>RAPPRESENTAZIONE DELLE IMMAGINI .....</b>	<b>7</b>
1.1.    RAPPRESENTAZIONE RGB .....	7
1.2.    RAPPRESENTAZIONE HSV .....	8
1.3.    RAPPRESENTAZIONE L*A*B* .....	8
<b>CLASSIFICAZIONE DELLE IMMAGINI .....</b>	<b>11</b>
2.1.    SUPPORT VECTOR MACHINE.....	11
2.2.    SVM MULTICLASSE .....	11
<b>CLASSIFICAZIONE DELLE IMMAGINI .....</b>	<b>13</b>
3.1.    EME-LBP FEATURE .....	13
3.2.    COLOR RATIO FEATURE .....	14
3.3.    MINIMUM COLOR DISTANCE .....	15
3.4.    GIST .....	16
3.5.    EME-GIST .....	17
3.6.    ORGLBP.....	17
<b>TEST DEI METODI E RISULTATI .....</b>	<b>19</b>
4.1.    DATASET UTILIZZATI.....	19
4.2.    CLASSIFICAZIONE “ONE VS ONE” BINARIA .....	20
4.3.    CLASSIFICAZIONE “ONE VS ALL” GLOBALE .....	23
4.4.    MODIFICA DEI PARAMETRI DI SVM .....	23
4.5.    TEST CON DATASET ESTESO .....	24
<b>CONCLUSIONI .....</b>	<b>27</b>
<b>BIBLIOGRAFIA .....</b>	<b>29</b>



# Introduzione

Nell'ambito didattico lo studio della storia dell'arte è tutt'oggi molto importante. Studiarla infatti permette di comprendere meglio ciò che è accaduto in passato, visto che è influenzata da avvenimenti che hanno segnato la società nella storia. Molti sono stati i movimenti artistici che nelle varie epoche hanno contraddistinto la produzione artistica, arrivando ad un'esplosione di movimenti nell'ultimo millennio in quanto a nascite di nuove correnti, le une correlate alle altre, talvolta difficilmente distinguibili.

Data l'evoluzione tecnologica, l'esigenza di un supporto all'occhio umano può essere risolta dall'aiuto dell'intelligenza artificiale e più nello specifico da classificatori e reti neurali. In questo documento il focus riguarda prevalentemente i primi.

Per prima cosa vengono discussi alcuni metodi già esistenti per l'estrazione di feature utili alla classificazione dei dipinti, sono poi descritte delle varianti di alcuni di essi ed infine sono illustrati i risultati ottenuti da tali metodologie su due differenti dataset.

Le immagini utilizzate raffigurano solo ritratti: innanzitutto perché in un dataset si trovano solo questo tipo di immagini, poi perché la rappresentazione dei volti è un elemento comune di moltissimi dipinti, anche se differente a seconda della corrente artistica di appartenenza.



# Capitolo 1

## Rappresentazione delle immagini

Per riconoscere il movimento artistico a cui è associabile un quadro, se non possiede una risposta immediata o mnemonica, l'essere umano procederà attuando un ragionamento basato sui dati del quadro (nome dell'opera, artista, periodo storico in cui è stata realizzata) o osservandone le caratteristiche visive. Tuttavia, un software ha uno svantaggio: se non possiede i metadati del quadro, allora disporrà solo di una matrice di pixel contenente il colore di ognuno e da questi dovrà ricavare le relazioni attraverso metodi matematici. In parole povere, quindi, non possiede quindi la visione dell'immagine nella sua interezza.

Per un elaboratore ci sono diverse rappresentazioni del colore e, a seconda di ognuna, è possibile ricavare proprietà differenti in relazione all'immagine. In seguito, verranno descritte solo le più rilevanti ai fini della realizzazione delle feature trattate.

### **1.1. Rappresentazione RGB**

Il modello RGB [1] (Red, Green, Blue) è il metodo per la rappresentazione dei colori più diffuso nell'ambito delle immagini digitali. Si tratta di un modello di tipo additivo, in cui il colore di ogni pixel è dato dalla somma del valore dei tre piani che lo compongono: rosso, verde e blu.

Dunque, un'immagine in formato RGB ha una rappresentazione matriciale del tipo “Altezza×Larghezza×3”, ovvero per ogni pixel ci sono i 3 valori sopra descritti.

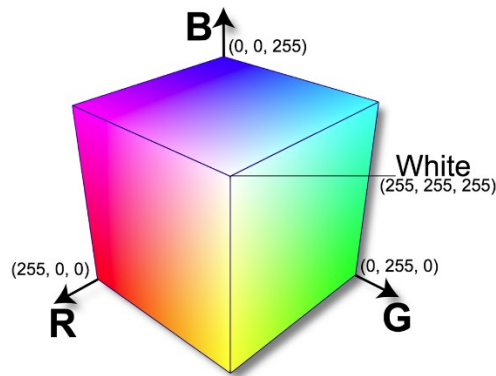


Figura 1.1: Rappresentazione tridimensionale del modello RGB.

## 1.2. Rappresentazione HSV

Il modello HSV (Hue, Saturation, Value) è anch'esso un modello di tipo additivo e come RGB associa ad ogni pixel tre valori: il primo valore è la tonalità, rappresentata attraverso un angolo intorno all'asse verticale a cui sono associati rosso ( $0^{\circ}$ - $119^{\circ}$ ), verde ( $120^{\circ}$ - $239^{\circ}$ ), blu ( $240^{\circ}$ - $359^{\circ}$ ); il secondo valore è la saturazione, i cui valori spaziano tra 0 e 1, e rappresenta l'intensità della tonalità del pixel; il terzo valore è l'intensità, anch'essa compresa tra 0 e 1, in cui gli estremi sono rispettivamente nero e bianco.

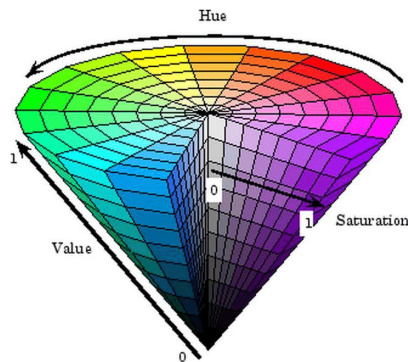


Figura 1.2: Rappresentazione tridimensionale del modello HSV.

## 1.3. Rappresentazione $L^*A^*B^*$

Il modello  $L^*A^*B^*$ , o più precisamente, CIELAB, è uno standard per la rappresentazione dei colori. Anche in questo caso ad ogni pixel sono associate tre dimensioni: la luminosità ( $L^*$ ), con valori compresi tra 0 (Nero) e 100 (Bianco) e le dimensioni colore-opponente ( $A^*$  e  $B^*$ ), che rappresentano rispettivamente la posizione del colore tra rosso e verde (se



negativo è più vicino al verde, altrimenti è più vicino al rosso) e la posizione tra giallo e blu (se negativo è più vicino al blu, altrimenti è più vicino al giallo). Questo standard è progettato per essere conforme alla percezione della vista umana ed è utile perché enfatizza le differenze tra colori simili, cosa che in seguito si rivelerà molto utile.

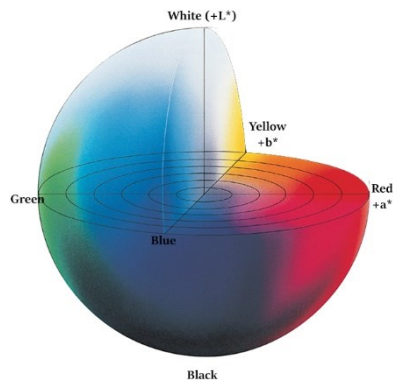


Figura 1.3: Rappresentazione tridimensionale del modello  $L^*A^*B^*$



# Capitolo 2

## Classificazione delle immagini

I metodi per classificare le immagini sono diversi, come ad esempio, “K Nearest Neighbors” (K-NN) o metodi probabilistici, ma per valutare la bontà delle feature estratte grazie ai metodi oggetto di questo documento, si utilizza Support Vector Machine (SVM). Il motivo di questa scelta è che SVM è più “agile” nei casi in cui gli elementi sono difficilmente distinguibili. A questi algoritmi vengono forniti come input una serie di vettori, detti “feature vector” o descrittori, i quali rappresentano nelle loro dimensioni determinate caratteristiche distintive dell’immagine.

### 2.1. Support Vector Machine

SVM [3] è una tipologia di classificatori specializzata nella classificazione binaria. L’obiettivo di un’entità di questo tipo è infatti quello di cercare un piano di separazione tra i vettori delle due classi, il quale permetta di classificare nuovi elementi in base alla loro posizione rispetto ad esso. Se nello spazio in cui sono rappresentati i vettori non è possibile raggiungere tale obiettivo, entrano in funzione gli SVM non lineari che attra-

verso  
delle funzioni, dette “funzioni kernel” [3], permettono di trovare un iperpiano in uno spazio vettoriale a dimensione superiore. Una volta trovato, viene riproiettato nello spazio originario.

Questa versione di SVM è la stessa utilizzata nel paragrafo 4.2 per la prima fase dei test.

### 2.2. SVM Multiclasse

Nel caso in esame le classi, ovvero i movimenti artistici, sono più di due, e per utilizzare SVM in situazioni di questo tipo ci sono due strategie: One vs One e One vs All. Nel primo caso, One vs One, con  $N$  classi viene creato un classificatore per ogni coppia di classi, per un totale di  $M=N \times (N-1)/2$  classificatori. Ogni elemento del test set, ovvero delle immagini utilizzate per valutare la precisione, viene classificato da ognuna delle  $M$

entità, ognuna delle quali fornisce come output la classe di appartenenza tra le due su cui è stato costruito. Al termine di queste operazioni ad ogni elemento sono associati i “voti” ricevuti da ogni modello e il movimento più votato è quello assegnato all’immagine. One vs All è la medesima strategia utilizzata per la valutazione globale dei descrittori descritta al paragrafo 4.3, e consiste nell’eseguire una classificazione SVM per ogni classe. L’altra classe con cui vengono comparati gli elementi è la classe “altro” contenente elementi di tutte le altre. L’output di tale classificazione è uno score e non un voto, ovvero la distanza dall’iperpiano. Viene scelta la classe che fornisce la maggiore distanza, perché significa che l’appartenenza dell’elemento è “maggiore” per quella.

# Capitolo 3

## Classificazione delle immagini

In questa sezione sono descritti i metodi base per l'estrazione delle feature utilizzati che si dividono in tre tipologie: quelli che estraggono feature relative alla texture delle immagini, quelli basati sull'analisi del colore dei pixel e infine quelli che utilizzano la configurazione globale dell'immagine. Inoltre, in alcuni casi, è stato possibile realizzare versioni diverse degli stessi descrittori, con l'utilizzo di differenti modelli di rappresentazione dei colori.

### 3.1. EME-LBP feature

Il primo metodo, il cui nome deriva da “Enhancement Measures” (Misure di miglioramento) e dall'algoritmo Local Binary Pattern [4], utilizza il contrasto relativo dei valori dei pixel, esaminando finestre  $F$  di dimensione  $3 \times 3$  attorno ad ognuno. In tali finestre ricava il pixel con valore massimo e quello con valore minimo e in base ad essi viene calcolato un nuovo valore da assegnare a tale elemento della matrice dell'immagine.

Il procedimento di calcolo affronta le seguenti fasi per ogni pixel e per ognuno dei tre piani del modello RGB:

1. Ricerca del massimo e del minimo nei pixel circostanti
2. Calcolo del valore detto “contrasto locale”, con la seguente formula:

$$ContrLocale = \frac{Max_F}{Min_F + K}$$

Il valore di  $K$  serve ad evitare divisioni per 0, mentre  $Max_F$  e  $Min_F$  sono i valori di massimo e minimo trovati al punto precedente.

3. Calcolo del nuovo valore del pixel:

$$P'(x, y) = A \times [1 - ContrLocale^B] + (1 - A) * P(x, y)$$

In cui  $A$  e  $B$  sono rispettivamente 0.4 e 0.1,  $P(x,y)$  è il valore precedente del pixel. Il risultato finale è una matrice, alla quale viene applicato l'algoritmo LBP, che permette di ottenere come output un array di dimensione 256, corrispondente al feature vector dell'immagine esaminata.

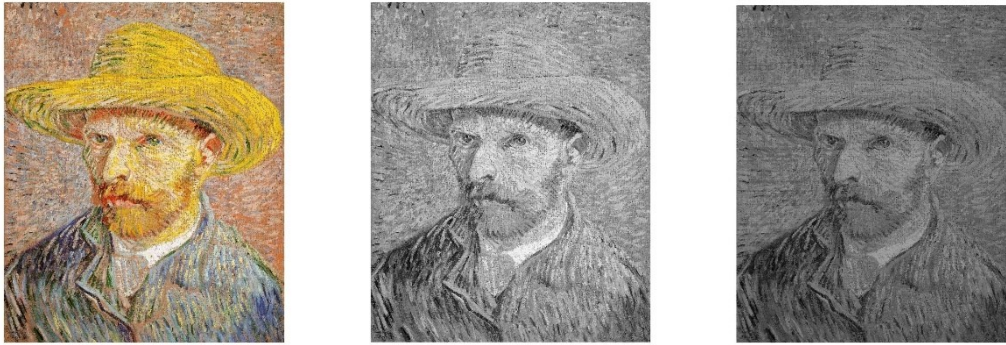


Figura 3.1: Comparazione (da sinistra a destra) tra l'immagine originale, la versione in scala di grigi e la versione che ha subito le modifiche della feature EME.

### 3.2. Color Ratio Feature

Color Ratio è una feature focalizzata sul colore. I modelli più efficaci sono RGB e HSV, infatti ne sono state implementate entrambe le versioni. Questo metodo si basa sulla relazione che c'è fra i valori dei pixel nei vari piani. In questo caso la finestra centrata su ogni pixel ha dimensione  $5 \times 5$  e il procedimento per ottenere il feature vector è il seguente, previa conversione nei colori HSV, se necessario:

1. Vengono create 3 matrici, una per ogni rapporto, delle dimensioni dell'immagine (ad esempio RG, RB e GB);
2. Per ogni pixel vengono effettuate tre sommatorie, una per ogni piano, dei pixel che la compongono la finestra;
3. Per ognuna delle matrici al punto 2, viene effettuato il rapporto tra le sommatorie dei piani a cui si riferiscono, generando il valore nella posizione corrispondente al pixel;

Dopo aver effettuato i passaggi descritti sopra si applica LBP alle tre matrici, formando tre array di dimensione 256, i quali vengono uniti in un unico feature vector di dimensione 768, che rappresenta l'immagine nella classificazione.

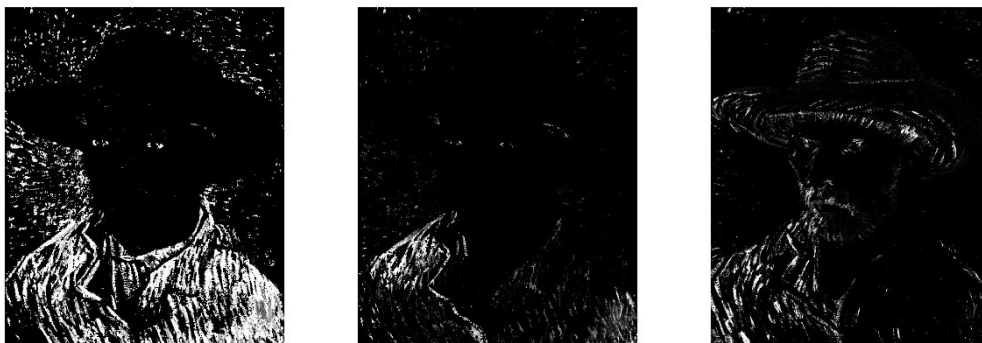


Figura 3.2: Output in scala di grigi delle tre matrici dei rapporti (HS, HV, SV) per il descrittore HSV color ratio.

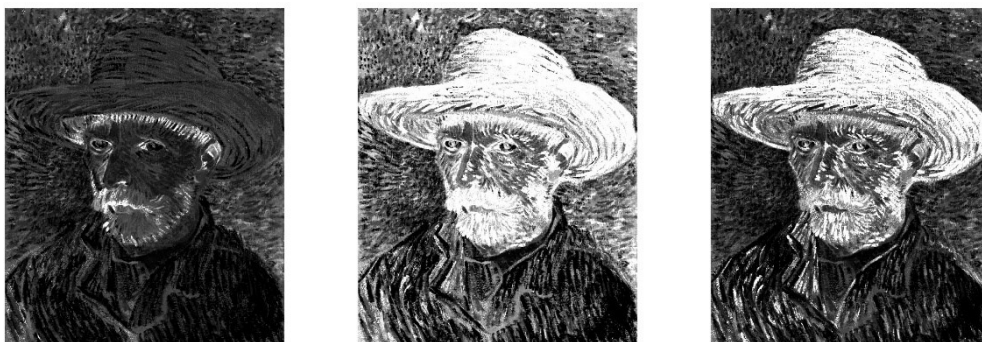


Figura 3.3: Output in scala di grigi delle tre matrici dei rapporti (RG, RB, GB) per il descrittore RGB color ratio.

Una possibile versione alternativa consiste nel considerare solamente il pixel esaminato nel calcolo del rapporto, ovvero ridurre le dimensioni della finestra a  $1 \times 1$ , con il vantaggio di eseguire meno operazioni.

### 3.3. Minimum Color Distance

MCD è un metodo di estrazione, come suggerisce il nome, basato sul colore. Differisce dai precedenti perché in questo caso il feature vector nella fase finale non viene costruito attraverso l'applicazione di LBP. Il concetto su cui si basa MCD è il fatto che ogni genere è caratterizzato dall'utilizzo di un certo tipo di tonalità. Dunque, un dipinto che presenti soprattutto colori “vicini” a tale genere, è probabile vi appartenga. Per calcolare questa

“vicinanza” si utilizzano i valori di rosso, verde e blu di ogni pixel come se fossero delle coordinate di un vettore.

Il primo passo consiste nel generare una palette colori, con  $N$  colori per ognuno dei generi. Per fare ciò va applicato un clustering di tipo k-means alle immagini di ogni genere, i cui centroidi saranno le coordinate RGB dei colori più presenti in ognuna.

Tali centroidi vengono quindi utilizzati per un ulteriore k-means interno ad ogni classe con i colori più usati nelle immagini che ne fanno parte. Il risultato è una palette  $N \times G$ , in cui  $N$  sono i colori estratti per ogni genere e  $G$  sono i generi esaminati.

Dopodiché tale palette viene utilizzata per calcolare il feature vector di ogni immagine, procedendo nel seguente modo:

1. Per ogni colore  $P_c$  della palette vengono esaminati tutti i pixel;
2. Per ogni pixel viene calcolata la distanza euclidea rispetto,  $P_c$ ;
3. Tali distanze vengono sommate nel seguente modo, a formare il valore corrispondente a  $P_c$  nella coordinata del feature vector corrispondente all'immagine:

$$Val_j = \sum_{m=1}^x \sum_{n=1}^y \frac{1}{dist(P_j, Image_{m,n})}$$

In cui  $x$  e  $y$  sono le dimensioni di altezza e larghezza dell'immagine,  $j$  è l'indice che scorre attraverso la palette (da 1 a  $N \times G$ ),  $P_j$  il colore della palette nella posizione  $j$ ,  $Image_{m,n}$  il valore dell'immagine nel pixel di coordinate  $m$  e  $n$ .

Ogni elemento del vettore  $N \times G$  è calcolato nel modo descritto sopra e descrive la vicinanza dei colori dell'immagine rispetto a quelli della palette.

Tale vettore viene poi utilizzato per ottenere una classificazione attraverso SVM.

### 3.4. Gist

Un altro metodo per l'estrazione dei descrittori è Gist (*Gradient-domain Image STitching*) [5]. Esso si basa sullo studio dell'immagine dal punto di vista della sua struttura complessiva [6].



Il codice utilizzato è stato preso direttamente dal repository di Matlab R2021a [7] ed è stato inserito senza modificare alcun parametro.

L'algoritmo di manipolazione delle immagini si articola nei seguenti passaggi:

1. Viene applicato il filtro di Gabor a  $L$  livelli, con  $O$  orientazioni diverse, che combinati permettono di ottenere  $N \times O$  mappe della dimensione di ogni immagine;
2. Ogni feature map viene suddivisa attraverso una griglia  $4 \times 4$  (16 regioni) e per ogni sezione viene calcolato il valore medio;
3. Concatenando i 16 valori di ogni mappa si ottiene un vettore di dimensione  $16 \times L \times O$ .

Nel caso specifico i valori di  $L$  e  $O$  sono rispettivamente 4 e 8 (quelli di default), la dimensione finale del descrittore risultante è 512.

### **3.5. EME-Gist**

In questo documento viene proposto il seguente descrittore, dal cui nome si evince che è la combinazione tra EME-LBP e Gist.

Il procedimento di estrazione è il medesimo di EME-LBP, ma nella fase finale invece di applicare LBP per ottenere il feature vector, viene utilizzato l'algoritmo Gist. Come spiegato nella parte relativa a quest'ultimo, l'output di ogni immagine ha dimensione 512. Il vantaggio principale di questa metodologia è che condensa in un unico passaggio l'analisi della texture dell'immagine con quella della sua composizione globale.

### **3.6. ORGLBP**

OrgLBP è l'algoritmo per l'estrazione del Local Binary Pattern. Tuttavia, si tratta della versione implementata da Marko Heikkilä e Timo Ahonen [8].

In breve, l'algoritmo esegue per ogni pixel le seguenti operazioni:

1. Per gli  $N$  pixel vicini in un cerchio di raggio  $R$  esegue un confronto con il pixel centrale, quello esaminato;
2. Se i vicini sono maggiori, nella loro posizione viene considerato il valore 1, altrimenti 0, generando una matrice binaria, la quale va convoluta con una matrice di potenze di 2, da 1 a  $2^{N-1}$ ;

3. Il risultato della convoluzione può essere visto come la composizione di un numero con  $N$  bit; infatti, il nuovo valore del pixel centrale è compreso tra 0 e  $2^N-1$ ; Una volta generata la nuova matrice LBP, si contano le occorrenze di ciascun valore, il cui risultato è detto “istogramma LBP”, un vettore di dimensione  $2^N$ . Nel caso specifico dei test in oggetto,  $N$  e  $R$  sono stati rispettivamente 8 e 3 e la dimensione dell’output è 256. OrgLBP è la versione di LBP utilizzata nei precedenti casi, ma è stato utilizzato anche come descrittore a sé stante dopo essere stato applicato alle immagini originali. L’unica differenza è l’utilizzo di una soglia aggiuntiva, cioè per avere il valore 1 nella posizione vicina occorre che tale pixel sia maggiore del valore di quello centrale sommato alla soglia. Tale soglia è 0.01.

# Capitolo 4

## Test dei metodi e risultati

Sono stati eseguiti diversi test per la valutazione dei descrittori, per analizzare diversi aspetti di ognuno e per avere una visione d'insieme più completa possibile. In tutto sono stati sperimentati dieci metodi, comprensivi di quelli citati nel capitolo 3 in diverse versioni elencate di seguito:

- LAB MCD: versione di MCD con la rappresentazione  $L^*A^*B^*$ ;
- RGB MCD: versione di MCD con la rappresentazione RGB;
- HSV-Color Ratio 1: versione principale di Color Ratio con la rappresentazione HSV;
- RGB-Color Ratio 1: versione principale di Color Ratio con la rappresentazione RGB;
- HSV-Color Ratio 2: versione alternativa di Color Ratio con la rappresentazione HSV, che effettua il rapporto pixel per pixel;
- RGB-Color Ratio 2: versione alternativa di Color Ratio con la rappresentazione RGB, che effettua il rapporto pixel per pixel.

### 4.1. Dataset Utilizzati

Per il test degli algoritmi di estrazione si utilizzano due dataset. Il primo [9] è un archivio di 927 immagini afferenti a 6 movimenti differenti: Impressionismo, Post-Impressionismo, Alto Rinascimento, Rinascimento centro-europeo, Ukiyo e Rococò. La cardinalità dei 6 stili non è equamente distribuita, con rispettivamente 204, 170, 163, 147, 116 e 127 immagini. Inoltre, tale dataset è composto solo da ritratti. Il secondo dataset utilizzato [10] invece non è altro che un'estensione del primo, con l'aggiunta di 407 immagini distribuite tra altri 3 generi: Cubismo, Espressionismo e Romanticismo, con rispettivamente 118, 140 e 149 immagini.

Tale composizione è stata raggiunta lavorando su un pool di 768 file, selezionando solo quelli che rappresentano ritratti.

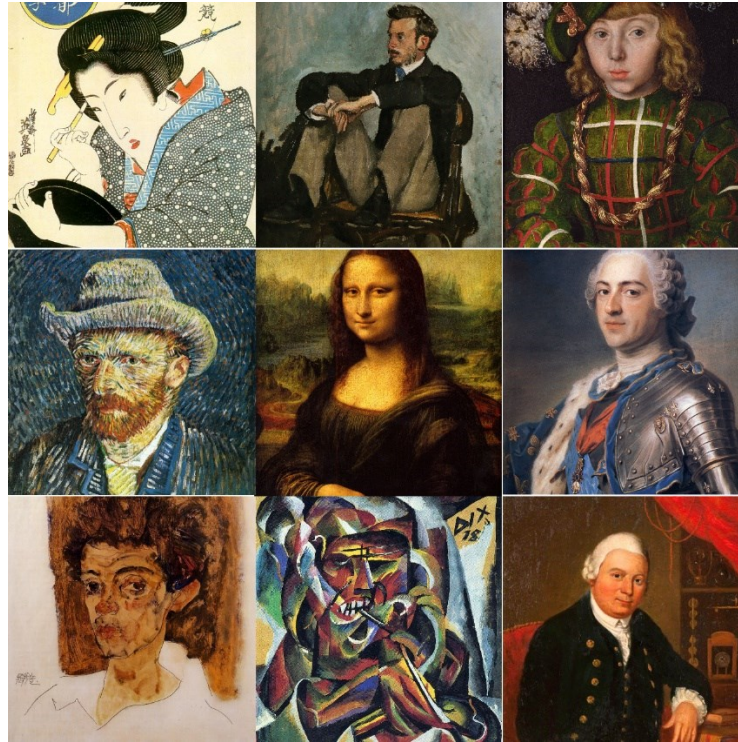


Figura 4.1: Esempi di immagini appartenenti ai generi del primo dataset (prime due righe) ed immagini aggiunte a formare la seconda versione del dataset (ultima riga).

## 4.2. Classificazione “One vs One” binaria

Innanzitutto, è stata eseguita la valutazione dell’accuracy valutando coppie di classi in tutte le possibili combinazioni. A tale scopo viene utilizzata la versione base della classificazione SVM di Matlab (la coppia di funzioni “`fitsvm`” e “`predict`”), applicata al primo dataset e sono stati eseguiti semplici calcoli statistici su ogni classificazione.

Le tabelle 4.1 e 4.2 riportano le percentuali ottenute per ogni descrittore. La prima cosa che si nota è la difficoltà di ogni descrittore, ovvero le inferiori percentuali ottenute, nel distinguere diverse correnti dello stesso movimento artistico, come ad esempio Impressionismo e post-Impressionismo. Al contrario quasi tutti sono stati in grado di riconoscere Ukiyo rispetto a tutti gli altri, essendo questo un movimento contraddistinto da tecniche e cultura completamente diverse rispetto agli altri.

Altro aspetto rilevante di questa tabella è che permette di intuire quali sono i descrittori più efficaci, tra tutti GIST ed EME-GIST, che quindi molto probabilmente lo risulterebbero anche in una classificazione completa, ovvero la seconda tipologia di test effettuati.

		EME-LBP	GIST	EME-GIST	LAB-MCD	RGB-MCD
Alto Rinascimento	Impressionismo	0.758	0.794	0.791	0.510	0.676
	Rinascimento Europeo	0.670	0.703	0.703	0.575	0.650
	Post Impressionismo	0.793	0.824	0.824	0.585	0.769
	Rococò	0.729	0.814	0.814	0.613	0.796
	Ukiyo	0.889	0.900	0.900	0.848	0.959
Impressionismo	Rinascimento Europeo	0.826	0.852	0.852	0.633	0.738
	Post Impressionismo	0.691	0.694	0.691	0.571	0.767
	Rococò	0.841	0.870	0.874	0.661	0.795
	Ukiyo	0.903	0.903	0.900	0.825	0.915
Rinascimento Europeo	Post Impressionismo	0.793	0.877	0.877	0.552	0.747
	Rococò	0.800	0.884	0.884	0.571	0.754
	Ukiyo	0.892	0.900	0.900	0.842	0.952
Post Impressionismo	Rococò	0.885	0.921	0.921	0.626	0.813
	Ukiyo	0.920	0.924	0.924	0.828	0.950
Rococò	Ukiyo	0.904	0.921	0.926	0.922	0.898

Tabella 4.1: Risultati delle comparazioni tra due classi con metà dei descrittori discussi.

		RGB Color Ratio 1	RGB Color Ratio 2	HSV Color Ratio 1	HSV Color Ratio 2	ORG-LBP
Alto Rinascimento	Impressionismo	0.738	0.722	0.700	0.711	0.741
	Rinascimento Europeo	0.643	0.656	0.673	0.640	0.670
	Post Impressionismo	0.751	0.766	0.772	0.787	0.769
	Rococò	0.659	0.729	0.711	0.711	0.744
	Ukiyo	0.850	0.875	0.839	0.842	0.892
Impressionismo	Rinascimento Europeo	0.791	0.779	0.791	0.773	0.826
	Post Impressionismo	0.686	0.694	0.716	0.700	0.678
	Rococò	0.790	0.835	0.790	0.841	0.867
	Ukiyo	0.865	0.896	0.871	0.853	0.903
Rinascimento Europeo	Post Impressionismo	0.816	0.819	0.816	0.835	0.809
	Rococò	0.748	0.812	0.828	0.784	0.808
	Ukiyo	0.869	0.884	0.869	0.880	0.892
Post Impressionismo	Rococò	0.864	0.871	0.892	0.903	0.889
	Ukiyo	0.893	0.903	0.851	0.889	0.906
Rococò	Ukiyo	0.882	0.908	0.878	0.900	0.900

Tabella 4.2: Risultati delle comparazioni tra due classi con gli altri descrittori discussi.

### **4.3. Classificazione “One vs All” globale**

Per avere una visione reale della precisione di ogni metodo, non è sufficiente osservare le percentuali ottenute comparando coppie di movimenti, bensì è necessaria una comparazione multiclasse.

Per eseguirla è stata utilizzata una versione modificata della classificazione binaria utilizzata in precedenza, basata sempre sull'utilizzo delle due funzioni sopra citate. La valutazione inoltre è stata basata su una suddivisione del dataset in 10 fold, per garantire risultati più generici possibile.

I risultati ottenuti sono osservabili nella tabella 4.3, che mette in luce le reali differenze di prestazioni tra i vari descrittori. Tra questi spicca nuovamente EME-GIST, descrittore proposto in questo documento. Inoltre, è possibile notare che le versioni “light” dei metodi Color Ratio presentano prestazioni molto simili a quelle delle versioni originali, rappresentando quindi delle valide alternative anche per quanto concerne le prestazioni globali del sistema.

### **4.4. Modifica dei parametri di SVM**

La funzione `fitesvm` presente su Matlab ha la possibilità di inserire diversi parametri utili alla classificazione.

Nell'esecuzione dei test il focus è stato su due di questi: `KernelFunction` e `BoxConstraint`. Nel primo caso si tratta della funzione kernel da utilizzare, `BoxConstraint` invece è un parametro (intero positivo) che definisce le penalità subite durante l'addestramento a causa di support vector che si trovano fuori posizione. Quest'ultimo permette inoltre di evitare overfitting, ovvero più è alto, meno support vector considera per la ricerca dell'iperpiano di separazione. Tuttavia, partendo dal valore di default (1), è stato necessario ridurlo a valori frazionari, tranne per i due MCD. La ricerca dei valori ottimali è stata effettuata operando una suddivisione di tipo k-fold cross-validation: il dataset è stato diviso nel seguente modo: 1/10 dei dati è stato scelto come test set, i restanti 9/10 sono stati suddivisi in 10 fold, ognuno dei quali conteneva 1 fold per la validazione e 9 di training.

A rigor di logica, nel primo caso dovrebbe essere aumentata la probabilità di incorrere nell'overfitting; tuttavia, la ricerca dei valori migliori di tale variabile è stata eseguita

attraverso una valutazione di tipo k-fold. In tale scenario, quindi, è possibile concludere che tale valore non è buono solo per un caso isolato, ma lo è per il descrittore in generale. Nella tabella 4.3 è presente anche il miglior valore di BoxConstraint trovato per ogni metodo.

<b>Descrittore</b>	<b>Accuracy</b>	<b>Box Constraint</b>	<b>Dimensione Vettore</b>
<b>EME-LBP</b>	63,71%	0.0008	256
<b>GIST</b>	66,52%	0.00046	512
<b>EME-GIST</b>	66.20%	0.00046	512
<b>LAB-MCD</b>	61.09%	1.8	96
<b>RGB-MCD</b>	62.50%	3	96
<b>RGB-Color Ratio 1</b>	60.22%	0.0044	768
<b>RGB-Color Ratio 2</b>	59.13%	0.00038	768
<b>HSV-Color Ratio 1</b>	63.91%	0.000073	768
<b>HSV-Color Ratio 2</b>	64.78%	0.00066	768
<b>Org-LBP</b>	61.41%	0.00057	256

Tabella 4.3: Risultati dell'accuracy globale e valori di Box Constraint ottenuti dall'analisi delle prestazioni attraverso k-fold cross-validation.

## 4.5. Test con dataset esteso

Alcuni metodi [11], sono stati testati solo sul primo dataset. Sono quindi stati testati insieme agli altri su un dataset più corposo per verificarne l'efficacia in diverso contesto. In questo caso è stata effettuata solo l'analisi dell'accuracy globale, i cui risultati sono illustrati nella tabella 4.4. Dai test è possibile notare che la maggior parte dei metodi presenta un accuracy quasi dimezzata, ad eccezione di quelli collegati a GIST e le due versioni di MCD, che nonostante la diminuzione superano di poco il 50%.



<b>Descrittore</b>	<b>Accuracy</b>
<b>EME-LBP</b>	28,47%
<b>GIST</b>	64,74%
<b>EME-GIST</b>	64.22%
<b>LAB-MCD</b>	51.42%
<b>RGB-MCD</b>	50.75%
<b>RGB-Color Ratio 1</b>	33.05%
<b>RGB-Color Ratio 2</b>	20.77%
<b>HSV-Color Ratio 1</b>	15.62%
<b>HSV-Color Ratio 2</b>	31.63%
<b>Org-LBP</b>	27.66%

Tabella 4.4: Risultati dei descrittori applicati al secondo dataset.



# Conclusioni

In conclusione, le nuove versioni dei descrittori presentate nei capitoli precedenti sono considerabili come delle alternative credibili alle versioni principali, per differenti motivi. Per quanto riguarda Color Ratio 2, i risultati hanno dimostrato che riducendo le dimensioni della finestra su cui viene calcolato il rapporto non si ha una perdita rilevante di accuracy, bensì i valori rimangono simili, se non superiori. Nel caso di EME-GIST invece le prestazioni sono superiori a EME-LBP, mentre rispetto a GIST (metodo con l'accuracy più elevata) ha prestazioni leggermente inferiori, ma con maggiore contenuto informativo al suo interno.

Se le modifiche apportate a Color Ratio hanno prestazioni nella media rispetto agli altri analizzati, EME-GIST in quanto a prestazioni è preceduto solamente da GIST, rappresentando una valida opzione come descrittore nella classificazione dei dipinti.



# Bibliografia

- [1] Süsstrunk, S., R. Buckley and Steve Swen. “Standard RGB Color Spaces.” Color Imaging Conference (1999).
- [2] <https://www.lorenzogovoni.com/support-vector-machine/#:~:text=Gli%20algoritmi%20SVM%20utilizzano%20un,nella%20maggior%20parte%20dei%20casi.>
- [3] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). *Machine Learning*. 20 (3):273–297. doi:10.1007/BF00994018. S2CID 206787478.
- [4] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 240 (7):0 971–987, July 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1017623
- [5] Richard Szeliski, “Computer Vision: Algorithms and Applications”, 2010, Springer. Page 461
- [6] A. Oliva and A. Torralba. Modeling the shape of a scene: a holistic representation of the spatial envelope. *IJCV* 60, 2004.
- [7] Aditya Khosla (2021). Computer vision feature extraction toolbox (<https://github.com/adikhosla/feature-extraction>), GitHub. Retrieved September 8, 2021.
- [8] Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. “Face recognition with local binary patterns.” *Computer vision-eccv* 2004.
- [9] Immagini per il primo dataset: <https://data.mendeley.com/datasets/289kxpn57/1>
- [10] Immagini per il secondo dataset: <https://www.kaggle.com/torres07/art-movements>

- [11] S. Liu, J. Yang, S.S. Agaian, et al., “Novel features for art movement classification of portrait paintings”, *Image and Vision Computing* (2021), <https://doi.org/10.1016/j.imavis.2021.104121>