

# UNIVERSITA DEGLI STUDI DI PISA

Relazione Finale Tirocinio  
Scuola di Ingegneria  
Dipartimento di Ingegneria dell'Informazione

Influenza del numero di lunghezze d'onda sulla  
qualità dell'immagine spettrale.

Tirocinante: Cocomini Davide  
Tutor universitario: Pistolesi Francesco, Lazzerini Beatrice  
Tutor aziendale: Maccari Antonio, Maccari Francesca

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Lo spettro visibile . . . . .	3
1.2	Percezione del colore . . . . .	4
1.3	Metamerismo . . . . .	5
1.4	Scanner spettrofotometrico . . . . .	6
<b>2</b>	<b>Obiettivo del tirocinio</b>	<b>8</b>
2.1	Confronto tra immagini . . . . .	8
2.1.1	Decimazione e media . . . . .	9
2.1.2	SSIM . . . . .	10
2.1.3	Algoritmo . . . . .	10
2.1.4	Componenti della formula . . . . .	11
2.1.5	Lab-SSIM . . . . .	11
<b>3</b>	<b>Svolgimento</b>	<b>12</b>
3.1	Strumenti . . . . .	12
3.1.1	Tavola Savoia-Navona-Matte . . . . .	12
3.2	Rappresentazione delle immagini . . . . .	14
3.3	Esempio di esecuzione . . . . .	16
<b>4</b>	<b>Conclusioni</b>	<b>17</b>

# 1 Introduzione

## 1.1 Lo spettro visibile

Lo “spettro del visibile” si trova nella parte centrale dello spettro ottico, il quale comprende anche i raggi infrarossi e quelli ultravioletti. E’ composto da quella parte di spettro elettromagnetico che include tutte le colorazioni percepite dall’occhio umano, partendo dal rosso fino ad arrivare al viola. La colorimetria utilizza la percentuale della luce incidente che è stata riflessa (nell’intervallo del visibile (400-700 nm) per descrivere il colore dell’oggetto. Applicazioni particolari quali la misurazione della fluorescenza, del bianco ed i colori mimetici prendono in considerazione anche le radiazioni UV (350-400nm) e NIR (700-1300nm). Ciascun oggetto colorato viene pertanto definito da una curva di riflettanza, similmente alle impronte digitali nell’uomo.

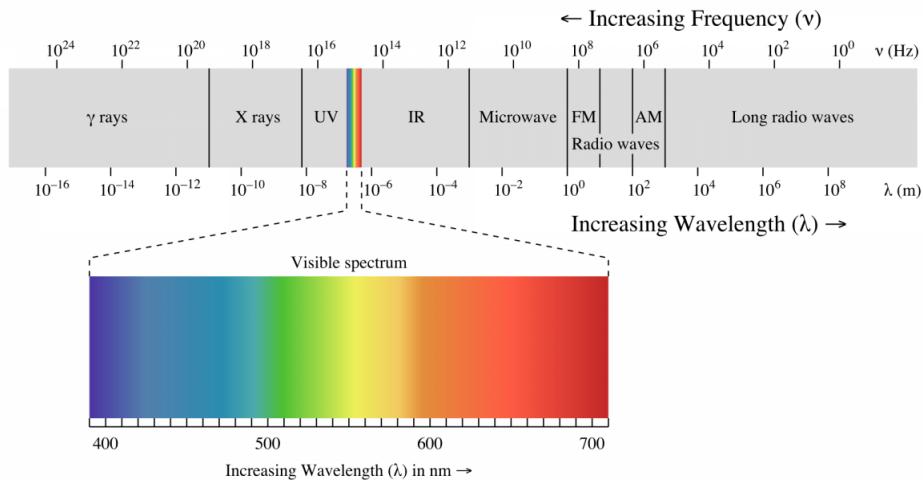


Figura 1: ”Lo spettro visibile”

## 1.2 Percezione del colore

Il colore nasce dalla luce. La luce che colpisce un oggetto viene parzialmente assorbita a seconda del colore. La parte non assorbita viene riflessa e trasmessa ai recettori cromatici all'interno dell'occhio umano. Questi ultimi trasformano la luce assorbita in impulsi che percorrono le vie nervose fino a raggiungere il cervello, dove vengono interpretati: nasce così un'impressione cromatica. Dal punto di vista prettamente biologico il colore si genera pertanto nell'occhio dell'osservatore e costituisce un'impressione sensoriale.

A proposito di impressione sensoriale: ciascun individuo "percepisce" il colore in modo differente. Tale fenomeno non è riconducibile solamente al fatto che non esistono mai due occhi uguali tra loro. Anche l'interpretazione del colore varia infatti da individuo ad individuo. Perfino la stessa persona può percepire differentemente il colore in momenti diversi ed in base allo stato d'animo. Il colore stesso può pertanto generare sensazioni differenti.

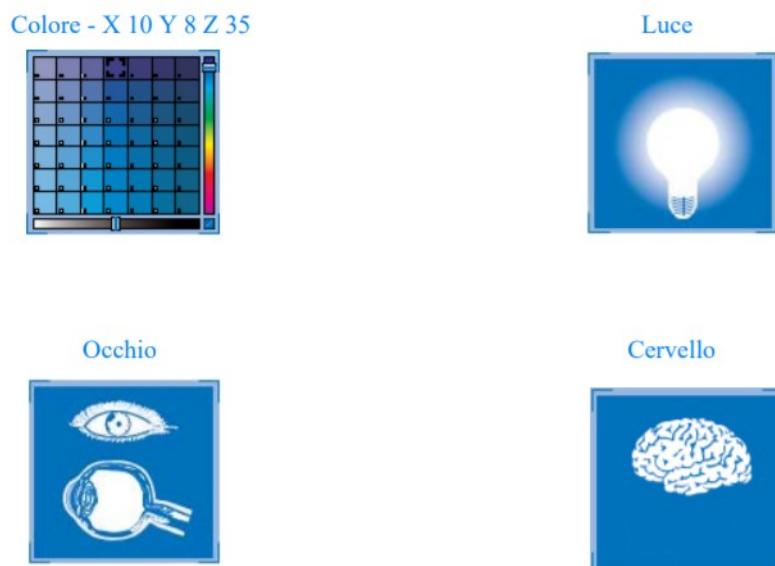


Figura 2: "Percezione del colore"

### 1.3 Metamerismo

Ciascuno di noi conosce il fenomeno per cui un oggetto colorato osservato sotto una determinata sorgente di luce come la luce diurna, presenta un colore diverso da quello visto sotto un'altra sorgente come ad esempio la luce di una lampadina ad incandescenza. Tale mutamento cromatico, caratteristico di quasi tutti gli oggetti colorati, viene spesso erroneamente definito metamerismo. Ma che cos'è in realtà il metamerismo? Il metamerismo può essere spiegato in termini di variazioni dei valori di tristimolo (X, Y e Z), che quantificano la percezione del colore. Due campioni sono uguali sotto una determinata sorgente di luce quando i relativi valori XYZ sono identici per quel particolare illuminante. Tale condizione è sempre soddisfatta sotto tutti gli illuminanti nel caso di campioni identici caratterizzati da curve di riflettanza uguali. I campioni metamerici presentano invece curve di riflettanza diverse. Ne consegue che tali campioni possono avere valori di tristimolo XYZ uguali sotto un certo illuminante ed apparire uguali, ma al mutare dell' illuminante, valori XYZ diversi ed apparire dunque diversi fra loro.

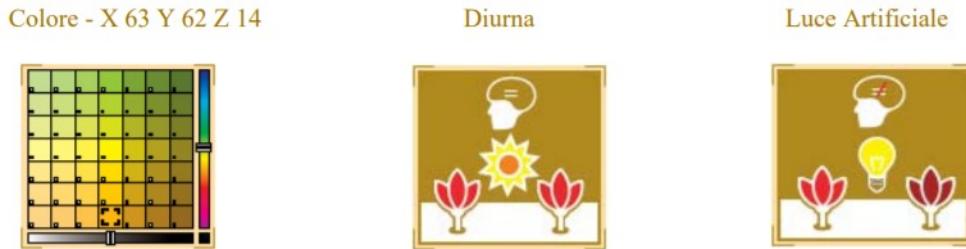


Figura 3: "Metamerismo"

## 1.4 Scanner spettrofotometrico

Lo scanner spettrofotometrico è nato per riprendere immagini adatte a costituire archivi di dipinti e di altre opere d'arte di superficie piana, utilizzabili per scopi di diagnostica, di restauro e di multimedialità. Le immagini d'archivio devono possedere almeno questi requisiti: riprodurre fedelmente il colore, essere confrontabili con immagini dello stesso oggetto riprese a distanza di tempo, essere stabili nel tempo per fornire una documentazione storica delle opere d'arte. Immagini di questo tipo si ottengono se sono basate sulla misurazione del fattore di riflessione spettrale della loro superficie. Esso è determinato dal rapporto tra il segnale dovuto alla luce riflessa dalla superficie e quella riflessa da un riflettore ideale in identiche condizioni di misurazione ed è quindi indipendente dalle specifiche tecniche della strumentazione usata per la sua misurazione. Inoltre, consente di esprimere il colore nello spazio colorimetrico dell'Osservatore Standard della CIE che è equivalente allo spazio colorimetrico del sistema di visione dell'uomo. Le immagini basate sul fattore di riflessione spettrale costituiscono una nuova categoria di immagini digitali: le immagini spettrali. Le immagini spettrali hanno parecchi vantaggi rispetto alle immagini digitali tricromatiche perché consentono:

1. di sfruttare al meglio il sensore elettronico (numero pixel-immagine uguale numero pixel-sensore);
2. il controllo del metamerismo;
3. la codifica dei colori in ogni spazio colorimetrico;
4. il loro uso anche con i futuri sviluppi della colorimetria;
5. la riproduzione su base spettrale;
6. di scegliere l'illuminante e il tipo di Osservatore Standard CIE per il calcolo del colore.

Lo scanner iperspettrale ottiene informazioni relative alla luce riflessa dall'immagine ottenendo così una serie di immagini su più frequenze che unite compongono l'immagine vera e propria.

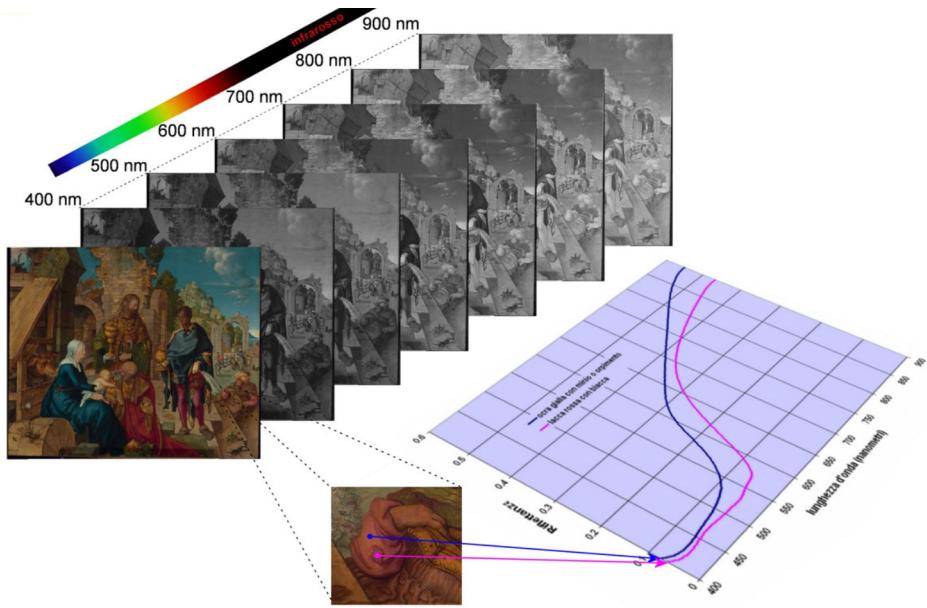


Figura 4: "Lunghezze d'onda"

## **2 Obiettivo del tirocinio**

### **2.1 Confronto tra immagini**

Il tirocinio ha l'obiettivo di ricercare una metodologia che misuri la differenza esistente tra due immagini aventi il medesimo soggetto, ma realizzate in tempi o con mezzi differenti affinché si possa capire qual è il minimo numero di lunghezze d'onda che uno scanner deve ottenere continuando ad ottenere un risultato soddisfacente. Riducendo le lunghezze d'onda è possibile ottimizzare i processi produttivi, abbattendo i tempi di acquisizione e produzione. Nel dettaglio, lo scanner estrae immagini su 31 lunghezze d'onda differenti ottenendo come risultato un'immagine estremamente fedele all'originale. Si cercherà di confrontare questa immagine con una ottenuta sullo stesso soggetto ma con un numero di lunghezze d'onda inferiore.

### 2.1.1 Decimazione e media

Le immagini da confrontare sono ottenute applicando separatamente due formule alle 31 immagini rappresentanti le lunghezze d'onda acquisite dallo scanner che insieme compongono il soggetto preso in considerazione.

Entrambe le formule dovranno effettuare delle elaborazioni per passare da 31 immagini a 15 e a 7.

La prima formula è quella di Decimazione e consiste nel campionare con una certa cadenza le 31 immagini iniziali:

Dalle 31 lunghezze d'onda si passa a 15 prendendo esclusivamente le immagini a 410,430,450,470,490,510,530,550,570,590,610,630,650,670,690.

Dalle 15 lunghezze d'onda si passa a 7 prendendo esclusivamente le immagini a 430,470,510,550,590,630,670.

La seconda formula, quella della Media, consiste invece nel creare delle immagini tramite la fusione di sottogruppi di quelle iniziali. In particolare si applicano le formule:

$$1. \text{ 15 lunghezze d'onda: } Im_Y = \frac{(Im_X + Im_Y + Im_Z)}{2}$$

$$2. \text{ 7 lunghezze d'onda: } Im_Z = \frac{Im_X + Im_Y + Im_Z + Im_K + Im_J}{4}$$

con  $X, Y, Z, K$  e  $J$  i valori delle lunghezze d'onda associati all'immagine (400, 410 ecc).

Alle immagini ottenute con questi due metodi viene applicato un illuminante e vengono generate delle immagini  $L^*a^*b^*$  e quindi a colori. Queste immagini sono quelle che vogliamo riuscire a confrontare per capire se, riducendo il numero di lunghezze d'onda prese in considerazione, il risultato rimane accettabile.

### 2.1.2 SSIM

Risulta quindi necessario individuare un metodo oggettivo per il confronto tra l'immagine originale e quella compressa. Si è deciso di utilizzare lo structural similarity index method (SSIM) che è un metodo ampiamente utilizzato per la predizione la qualità percettiva delle immagini televisive e cinematografiche. L'SSIM considera il degrado dell'immagine come cambiamento percepito nelle informazioni strutturali, mentre incorpora anche importanti fenomeni percettivi, inclusi sia il mascheramento della luminanza che i termini di mascheramento del contrasto.

### 2.1.3 Algoritmo

L'indice SSIM viene calcolato su varie finestre di un'immagine. La misura tra due finestre  $x$  e  $y$  di dimensione NxN è:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

1.  $\mu_x$  la media di  $x$ ;
2.  $\mu_y$  la media di  $y$ ;
3.  $\sigma_x^2$  la varianza di  $x$ ;
4.  $\sigma_y^2$  la varianza di  $y$ ;
5.  $\sigma_{xy}$  la covarianza di  $x$  e  $y$ ;
6.  $c_1 = (k_1L)^2$ ,  $c_2 = (k_2L)^2$  due variabili per stabilizzare la divisione con il denominatore inadatto;
7.  $L$  la gamma dinamica dei valori dei pixel;
8.  $k_1 = 0.01$  e  $k_2 = 0.03$  predefiniti.

#### 2.1.4 Componenti della formula

La formula SSIM si basa su tre misurazioni di confronto tra i campioni di  $x$  e  $y$ : luminanza  $l$ , contrasto  $c$  e struttura  $S$ . Le singole funzioni di confronto sono:

$$1. \ l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

$$2. \ c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

$$3. \ s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

con, oltre alle definizioni di cui sopra:

$$c_3 = c_2/2$$

L'SSIM è quindi una combinazione ponderata di tali misure comparative:

$$SSIM(x, y) = [l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma]$$

#### 2.1.5 Lab-SSIM

La formula SSIM ha tuttavia un limite, può essere applicata solamente ad immagini in grayscale. Quello che invece vogliamo fare noi è confrontare due immagini nello spazio L\*a\*b\*.

Viene quindi usato un indice sperimentale denominato Lab-SSIM. Questo consiste nel calcolare, sfruttando le formule sopracitate, i valori di luminanza dei canali a e b, di contrasto e di struttura del canale L e fare la media tra tutti questi valori.

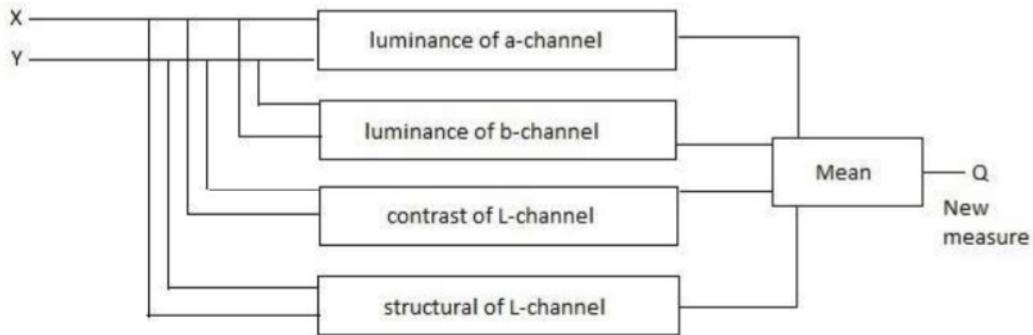


Figura 5: "Lab-SSIM"

## 3 Svolgimento

### 3.1 Strumenti

Per affrontare i problemi relativi alla manipolazione delle immagini si è deciso di utilizzare il linguaggio C++ con il supporto della libreria OpenCV. OpenCV è una nota libreria open-source disponibile in C++, Python e Java e permette di effettuare le operazioni più disparate sulle immagini. Inoltre, essa include a sua volta altre librerie come la libtiff che utilizzeremo per la lettura delle immagini.

#### 3.1.1 Tavola Savoia-Navona-Matte

Come soggetto dell'immagine è stata scelta la tavola Savoia-Navona-Matte, rappresentativa di numerose sfumature di colore ed utilizzate per effettuare test sugli spettrofotometri.

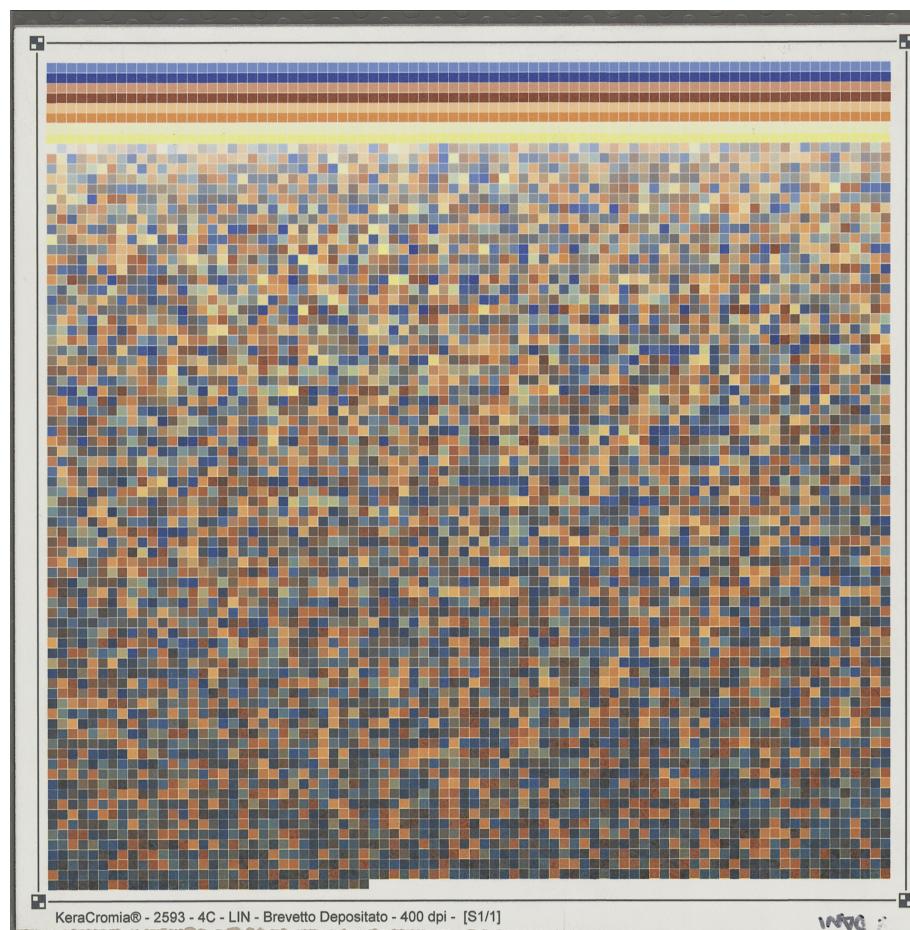


Figura 6: "Tavola Savoia-Navona-Matte"

La tavola è ottenuta come il risultato dell'unione di 31 bande nello spettro visibile da 400nm a 700nm, una ogni 10nm. In Figura 6 sono rappresentati alcuni esempi significativi delle bande sopracitate.

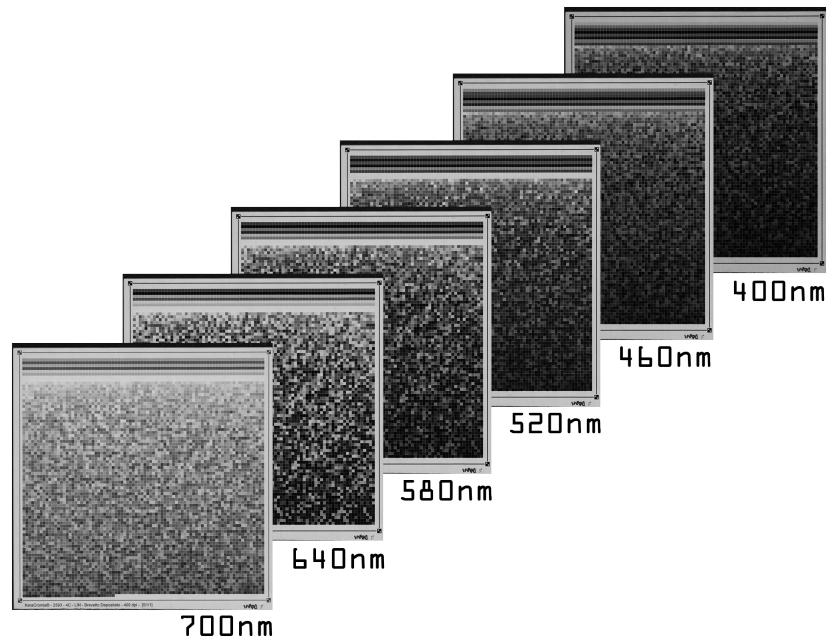


Figura 7: "Suddivisione in bande della tavola Savoia-Navona-Matte"

### 3.2 Rappresentazione delle immagini

Avendo a disposizione due immagini in formato TIFF, il primo problema da affrontare è riuscire a rappresentare queste immagini in delle strutture dati che possano essere poi manipolate per ottenerne informazioni. Per fare ciò sfruttiamo la libreria OpenCV e, in particolare, la funzione imread. Tramite questa funzione, si leggono le immagini e le si trasforma in matrici sfruttando la struttura dati "Mat" messa a disposizione dalla OpenCV:

---

```
1     Mat originalImage, compressedImage;
2     string defaultSettings ;
3     string originalImagePath("images/Tav-2593_Savoia_Navona_H0
4         .tif");
5     string compressedImagePath("images/Tav-2593
6         _Savoia_Navona_H0_a2.tif");
7     cout << "Vuoi usare le impostazioni di default? S/N" <<
8         endl;
9     cin >> defaultSettings;
10    if (defaultSettings.compare("N") == 0) {
11        string folderName, originalImageName,
12            compressedImageName;
13        cout << "Inserisci il nome della cartella dove
14            metterai le immagini" << endl;
15        cin >> folderName;
16        cout << "Inserisci il nome della prima immagine da
17            confrontare (senza formato)" << endl;
18        cin >> originalImageName;
19        cout << "Inserisci il nome della seconda immagine da
20            confrontare (senza formato)" << endl;
21        cin >> compressedImageName;
22        originalImagePath = folderName + "/" + originalImageName
23            + ".tif";
24        compressedImagePath = folderName + "/" +
25            compressedImageName + ".tif";
26        cout << "Impostazioni settate con successo" << endl;
27    }
28    cout << "Le immagini selezionate sono: " << endl <<
29        originalImagePath << endl << compressedImagePath <<
30        endl;
31    cout << "Inizio computazione ..." << endl;
32    originalImage = imread(originalImagePath,
33        CV_LOAD_IMAGE_UNCHANGED);
34    compressedImage = imread(compressedImagePath,
35        CV_LOAD_IMAGE_UNCHANGED);
```

---

La funzione imread tuttavia interpreta le immagini come se fossero nello spazio colore RGB, pertanto è necessario avvalersi della funzione cvtColor per usarle come immagini L\*a\*b\*. Inoltre ci avvaliamo della funzione split() per dividere l'immagine in 3 canali (L,a,b):

---

```

1   Mat originalImageLab, compressedImageLab;
2   cvtColor(originalImage, originalImageLab, COLOR_RGB2Lab);
3   cvtColor(compressedImage, compressedImageLab,
4           COLOR_RGB2Lab);
5   Mat originalImageSplitted[3], compressedImageSplitted[3];
6   split (originalImageLab, originalImageSplitted);
7   split (compressedImageLab, compressedImageSplitted);

```

---

C'è tuttavia un ulteriore problema. Le immagini da confrontare sono ad 8-bit e la conversione tramite cvtColor non è sufficiente per ottenere dei valori corretti a causa delle specifiche dettate da OpenCV.

Dalla documentazione di OpenCV si apprende il modo in cui la funzione converte i valori L,a,b:

$$L \leftarrow L * 255/100, a \leftarrow a + 128, b \leftarrow b + 128$$

Si procede quindi normalizzando le strutture Mat ottenute tramite un'apposita funzione:

---

```

1   Mat* normalizeLabValues(Mat image[]) {
2       int rows = image[0].rows;
3       int cols = image[0].cols;
4
5       for (int i = 0; i < rows; i++) {
6           for (int j = 0; j < cols; j++) {
7               // L
8               Scalar intensity = image[0].at<uchar>(i, j);
9               Scalar normalized_intensity = intensity.val[0] * 100
10                  / 256;
11               image[0].at<uchar>(i, j) = normalized_intensity.val
12                  [0];
13
14               // a
15               intensity = image[1].at<uchar>(i, j);
16               normalized_intensity = intensity.val[0] - 126;
17               image[1].at<uchar>(i, j) = normalized_intensity.val
18                  [0];

```

```

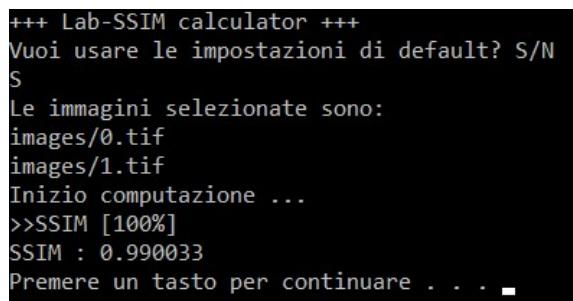
19         normalized_intensity = intensity.val[0] - 126;
20         image[2].at<uchar>(i, j) = normalized_intensity.val
21             [0];
22     }
23 }
24
25     image[0].convertTo(image[0], CV_64FC3);
26     image[1].convertTo(image[1], CV_64FC3);
27     image[2].convertTo(image[2], CV_64FC3);
28
29     return image;
30 }
```

---

Conclusi questi passaggi intermedi si può procedere col calcolare l'SSIM. La funzione scorre come una finestra mobile (di dimensione block size) contemporaneamente l'immagine originale e quella compressa e ne calcola le varianze, le medie e la covarianza. Ognuno di questi confronti contribuirà al risultato della funzione che sarà sempre un numero compreso tra -1 ed 1. Se le due immagini sono identiche, allora il risultato sarà 1 altrimenti sarà un numero sempre minore fino al -1 che rappresenta due immagini completamente diverse.

### 3.3 Esempio di esecuzione

All'esecuzione del programma si può scegliere se usare i nomi di default oppure selezionare una cartella e dei nomi personalizzati per le immagini da confrontare.



```

+++ Lab-SSIM calculator ***
Vuoi usare le impostazioni di default? S/N
S
Le immagini selezionate sono:
images/0.tif
images/1.tif
Inizio computazione ...
>>SSIM [100%]
SSIM : 0.990033
Premere un tasto per continuare . . . ■
```

Figura 8: "Esempio di esecuzione"

## 4 Conclusioni

Analizziamo di seguito i risultati ottenuti confrontando l'immagine originale con quella ottenuta attraverso il metodo della decimazione.

Si può concludere che la Decimazione produce un risultato praticamente identico alla reale scansione con lo stesso numero di lunghezze d'onda.

Tabella 1: Risultati

Lunghezze d'onda	Errore Medio	Deviazione Standard	Errore Massimo
Scansione 15	0.86	0.19	1.53
Scansione 7	1.45	0.35	2.64
Decimazione 15	0.87	0.21	2.04
Decimazione 7	1.46	0.37	2.64