



UNIVERSITÀ DEGLI STUDI DI MILANO

DIPARTIMENTO DI INFORMATICA E COMUNICAZIONE

VIA COMELICO 39 – 20135 MILANO – ITALIA

Realizzazione di software implementante le misure alternative dell'indice di contrasto percepito

1. Introduzione

In questo report vengono presentati i software realizzati per l'implementazione delle misure alternative del contrasto percepito presentate nel report precedente. Il software è stato realizzato in Matlab (C).

2. Il test set

A partire da due immagini con risoluzioni differenti sono state prodotte due diverse serie di immagini aumentando (per sei volte) e diminuendo (per sei volte) il contrasto con passo 10:

Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
---------	---------	---------	---------	---------	---------	------	---------	---------	---------	---------	---------	---------

Di conseguenza ci sono 6 immagini con contrasto minore dell'originale, e 6 immagini con contrasto maggiore. La variazione del contrasto è stata realizzata attraverso il software GIMP.

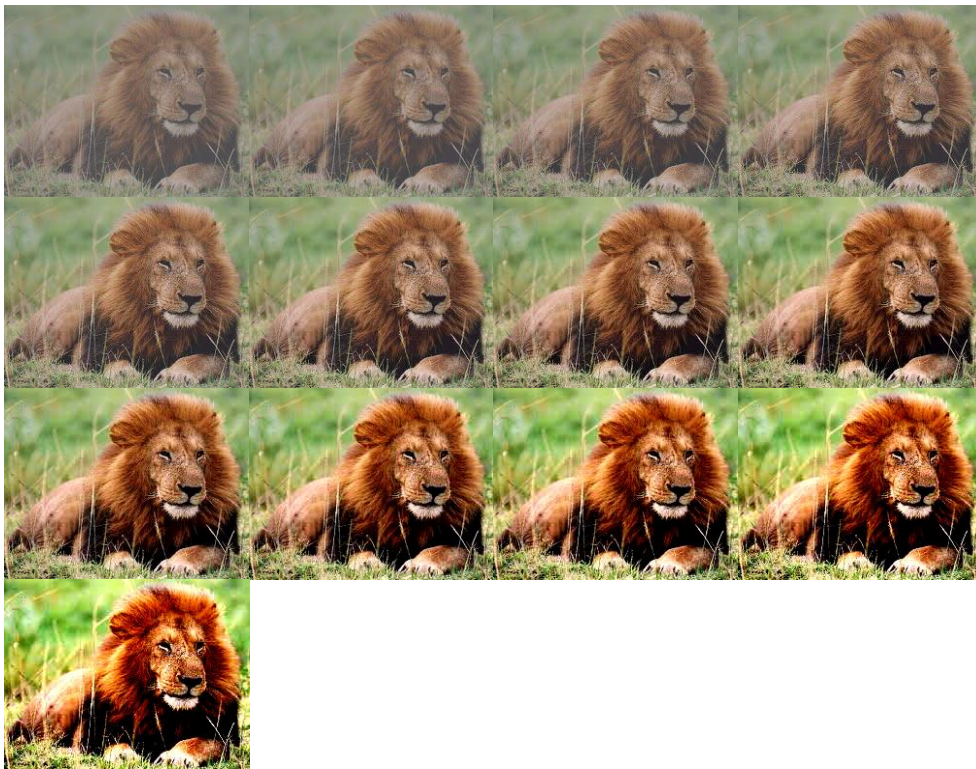


Figura 1. Prima immagine, *Leone*, di dimensione 253x199 pixel. La settima immagine è l'originale (seconda immagine a partire da sinistra, nella seconda colonna)

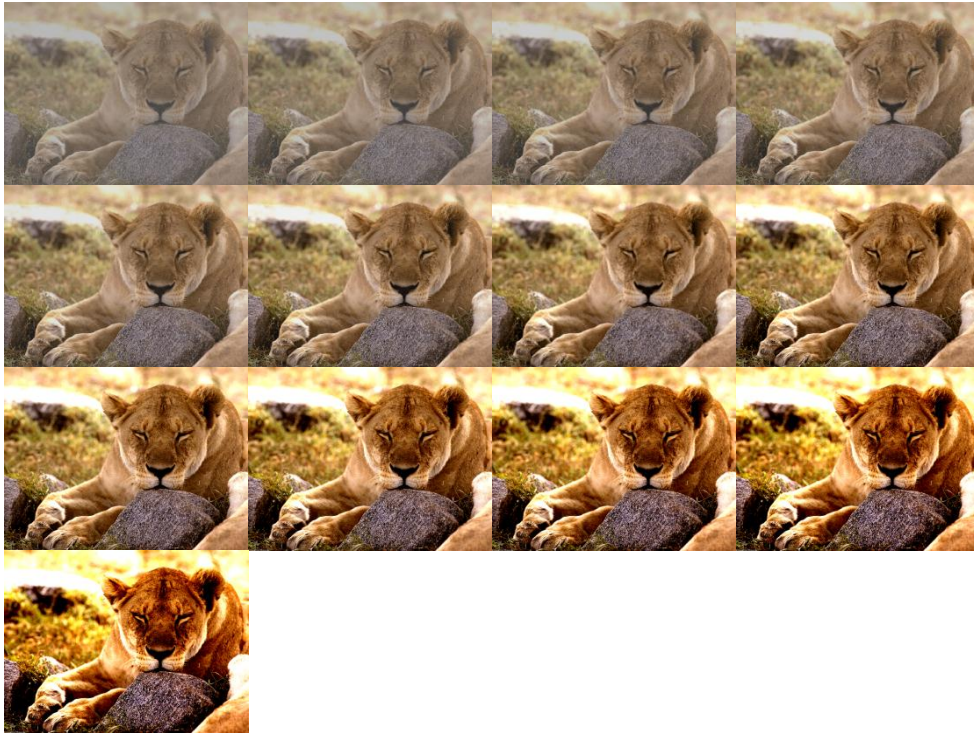


Figura 2. Seconda immagine, *Leonessa*, di dimensione 1600 x 1200 pixel. La settima immagine è l'originale.

Osservando le due immagini nella loro forma originale, è chiaro che l'immagine rappresentante il leone ha maggiore contrasto cromatico, a causa del suo sfondo.

Tale immagine infatti ha uno sfondo verde, differente dal colore castano della pelliccia, mentre lo sfondo della seconda immagine ha un colore simile al pelo della leonessa, rendendo soggetto e sfondo meno distinguibili.

Ci aspettiamo quindi che il contrasto misurato (CM) della seconda immagine sia minore del contrasto misurato nella prima.

3 Le misure di contrasto implementate

Il contrasto è stato calcolato usando l'algoritmo descritto nel report precedente, negli spazi colore CIELab e RGB, con le seguenti opzioni:

- 1) Usando il canale L^* del CIELab
- 2) Usando tutti e tre i canali dello spazio CIELab con pesi equivalenti
- 3) Usando tutti e tre i canali dello spazio CIELab con pesi differenti
- 4) Usando tutti e tre i canali RGB con pesi equivalenti

5) Usando tutti e tre i canali RGB con pesi differenti

L'algoritmo utilizzato è mostrato in figura 3. L'immagine originale è convertita nello spazio CIELab, viene costruita una struttura a piramide riducendo la dimensione dell'immagine originale. Ogni dimezzamento produce un nuovo livello e si continua fino a quando il numero di righe o colonne diventa minore di 25. Poi viene calcolato un contrasto medio per ogni livello, calcolando il valore di contrasto locale per tutti i pixel nell'immagine, considerando gli 8-vicini.

Alla fine, si calcola il valore medio del contrasto mediando i valori dei contrasti locali, risultando così in una misura globale di contrasto.

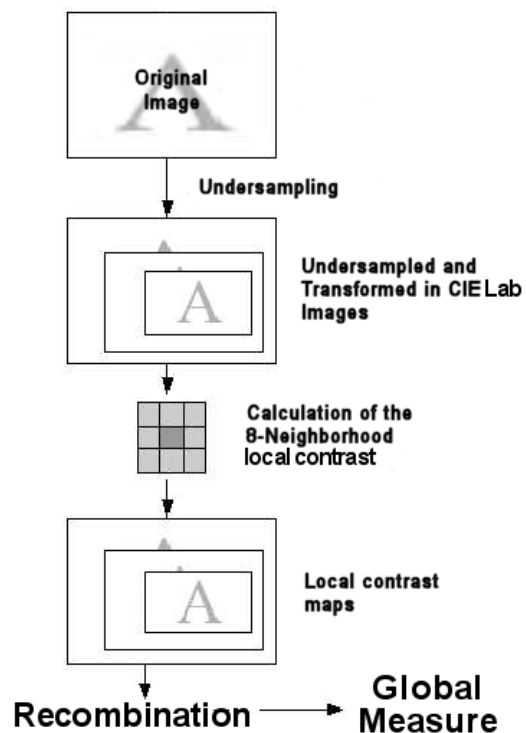


Figura 3: Schema generale della misura di contrasto implementata

3.1 Misura di contrasto sul canale L* dello spazio CIELab

In questa prima implementazione la misura di contrasto è applicata solo sull'informazione di luminanza del canale L dello spazio CIELab. In figura 4 viene mostrato il contrasto misurato (CM) per il *leone* e per la *leonessa*.

L'andamento della variazione di contrasto è quasi lineare in entrambe le immagini.

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
Lion	2.0565	2.4156	2.7998	3.2193	3.6752	4.1822	4.7316	5.2745	5.7613	6.2228	6.6762	7.1298	7.5597
Lioness	1.6724	1.9597	2.2713	2.6054	2.9691	3.3693	3.806	4.1953	4.5715	4.932	5.2606	5.5514	5.8147

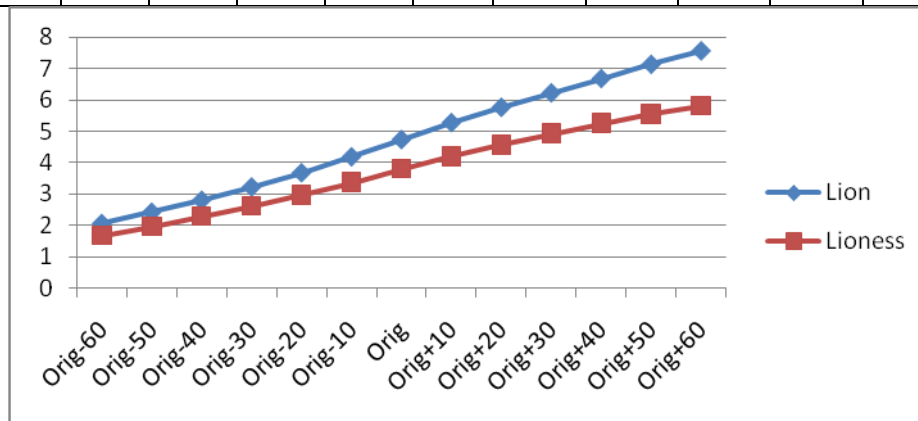


Figura 4: Leone vs Leonessa con 13 differenti variazioni di contrasto, misure su L*.

Il codice per il calcolo di questa implementazione è il seguente:

```
clear
Im=imread('lioness60.bmp');
ImD = im2double(Im);
cform = makecform('srgb2lab'); %creates the color transformation
structure C that defines the color space conversion specified by
'srgb2lab'.
labIm = applycform(ImD ,cform); % Conversion to CIELAB
[rows, cols, dim]=size(labIm);
levels=1; %original image
Temp = 0;TempL=0;
contrastTemp = 0;
contrastOfLevel = 0;
contrast = 0;
while min(size(labIm,1), size(labIm,2)) > 25
    for r=2:rows-1
        for c=2:cols-1
```

```

        TempL = ((abs(labIm(r,c,1)-labIm(r-1,c-
1,1))+abs(labIm(r,c,1)-labIm(r-1,c,1))+ abs(labIm(r,c,1)-labIm(r-
1,c+1,1))+abs(labIm(r,c,1)-labIm(r,c+1,1))+ abs(labIm(r,c,1)-
labIm(r+1,c+1,1))+abs(labIm(r,c,1)-labIm(r+1,c,1))+
abs(labIm(r,c,1)-labIm(r+1,c-1,1))+abs(labIm(r,c,1)-labIm(r,c-
1,1)))))/8;

        Temp = TempL;
        contrastTemp = contrastTemp + Temp;
    end
end
contrastOfLevel = contrastTemp / (rows*cols);
contrast = contrast + contrastOfLevel;
labIm = impyramid(labIm, 'reduce'); % Undersampling for next
level
levels=levels+1;
[rows, cols, dim]=size(labIm);
Temp=0;TempL=0; % Clearing all values of earlier level to
compute values for new level.
contrastTemp=0;
contrastOfLevel=0;
end
contrast=contrast/(levels-1)

```

3.2 Misura di contrasto sui tre canali dello spazio CIELab

In questa implementazione la misura di contrasto è applicata su tutti i canali dello spazio CIELab. In figura 5 viene mostrato il contrasto misurato (CM) per il *leone* e per la *leonessa*.

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
Lion	1.1654	1.3663	1.5833	1.8202	2.0788	2.3686	2.6804	3.0412	3.4335	3.8763	4.3777	4.9454	5.5957
Lioness	0.9212	1.0785	1.2478	1.4308	1.6303	1.8504	2.0912	2.3884	2.726	3.106	3.5201	3.9691	4.4658

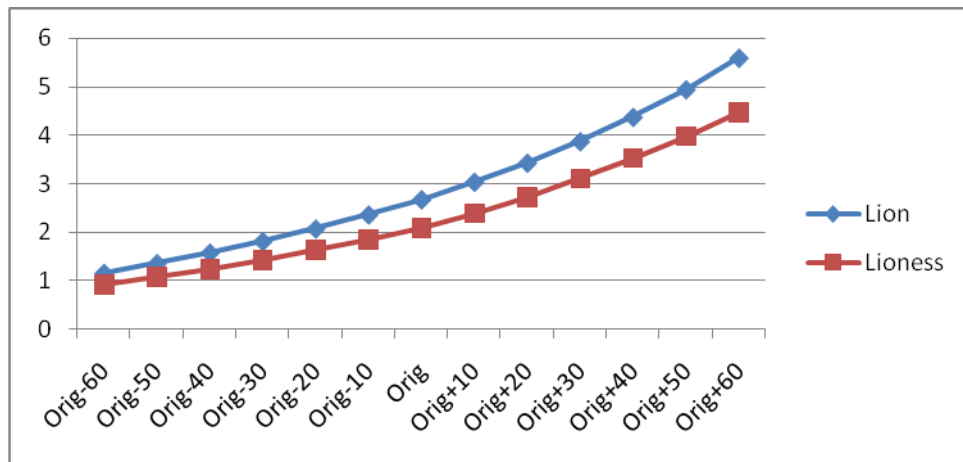


Figura 5: Leone vs Leonessa con 13 differenti variazioni di contrasto, misure su L^* , a^* , b^* , pesi equivalenti.

La figura 5 mostra i valori di CM del leone e della leonessa, pesando tutti i canali in maniera uguale: $0.33 \times L^*$, $0.33 \times a^*$ e $0.33 \times b^*$ dove L^* è la lightness e a^* e b^* sono le componenti cromatiche.

L'utilizzo dei canali cromatici ha ridotto i valori di contrasto misurato in entrambe le immagini. Ad es. il valore massimo di CM nel leone passa da 7.5597 a 5.5957 e nella leonessa da 5.8147 a 4.4658.

Il codice per il calcolo di questa implementazione è il seguente:

```
clear
Im=imread('lioness60.bmp');
ImD = im2double(Im);
cform = makecform('srgb2lab'); %creates the color transformation
structure C that defines the color space conversion specified by
'srgb2lab'.
labIm = applycform(ImD ,cform); % Conversion to CIELAB
[rows, cols, dim]=size(labIm);
levels=1; %original image
Temp = 0;TempL=0;TempA=0;TempB=0;
contrastTemp = 0;
contrastOfLevel = 0;
contrast = 0;
while min(size(labIm,1), size(labIm,2)) > 25
```

```

    for r=2:rows-1
        for c=2:cols-1
            TempL = 0.33*((abs(labIm(r,c,1)-labIm(r-1,c-
1,1))+abs(labIm(r,c,1)-labIm(r-1,c,1))+ abs(labIm(r,c,1)-labIm(r-
1,c+1,1))+abs(labIm(r,c,1)-labIm(r,c+1,1))+ abs(labIm(r,c,1)-
labIm(r+1,c+1,1))+abs(labIm(r,c,1)-labIm(r+1,c,1))+
abs(labIm(r,c,1)-labIm(r+1,c-1,1))+abs(labIm(r,c,1)-labIm(r,c-
1,1))))/8;

            TempA = 0.33*((abs(labIm(r,c,2)-labIm(r-1,c-
1,2))+abs(labIm(r,c,2)-labIm(r-1,c,2))+ abs(labIm(r,c,2)-labIm(r-
1,c+1,2))+abs(labIm(r,c,2)-labIm(r,c+1,2))+ abs(labIm(r,c,2)-
labIm(r+1,c+1,2))+abs(labIm(r,c,2)-labIm(r+1,c,2))+
abs(labIm(r,c,2)-labIm(r+1,c-1,2))+abs(labIm(r,c,2)-labIm(r,c-
1,2))))/8;

            TempB = 0.33*((abs(labIm(r,c,3)-labIm(r-1,c-
1,3))+abs(labIm(r,c,3)-labIm(r-1,c,3))+ abs(labIm(r,c,3)-labIm(r-
1,c+1,3))+abs(labIm(r,c,3)-labIm(r,c+1,3))+ abs(labIm(r,c,3)-
labIm(r+1,c+1,3))+abs(labIm(r,c,3)-labIm(r+1,c,3))+
abs(labIm(r,c,3)-labIm(r+1,c-1,3))+abs(labIm(r,c,3)-labIm(r,c-
1,3))))/8;

            Temp = TempL + TempA + TempB;
            contrastTemp = contrastTemp + Temp;
        end
    end
    contrastOfLevel = contrastTemp / (rows*cols);
    contrast = contrast + contrastOfLevel;
    labIm = impyramid(labIm, 'reduce'); % Undersampling for next
level
    levels=levels+1;
    [rows, cols, dim]=size(labIm);
    Temp=0;TempL=0;TempA=0;TempB=0; % Clearing all values of
earlier level to compute values for new level.
    contrastTemp=0;
    contrastOfLevel=0;
end
contrast=contrast/(levels-1)

```


3.3 Misura di contrasto sui tre canali dello spazio CIELab con pesi differenti

In questa implementazione la misura di contrasto è applicata su tutti i canali dello spazio CIELab ma i tre canali sono pesati in modo differente: $0.50 \times L^*$, $0.25 \times a^*$ e $0.25 \times b^*$.

In figura 6 viene mostrato il contrasto misurato (CM) per il leone e per la leonessa.

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
Lion	1.397	1.639	1.8994	2.1837	2.4937	2.84	3.2135	3.6225	4.0415	4.4923	4.9855	5.529	6.1291
Lioness	1.116	1.3069	1.5131	1.7353	1.9774	2.2442	2.5358	2.8583	3.208	3.586	3.9819	4.3948	4.8368

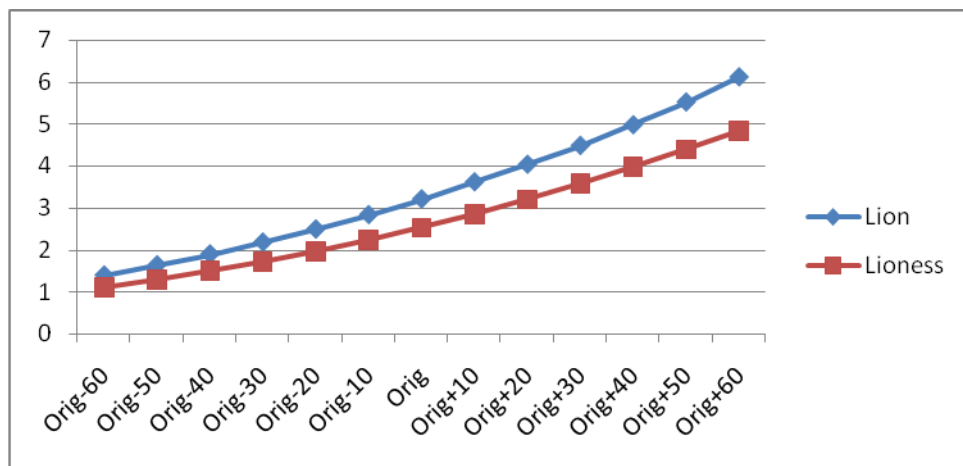


Figura 6: Leone vs Leonessa con 13 differenti variazioni di contrasto, misure su L^* , a^* , b^* , pesi diversi.

Il codice per il calcolo di questa implementazione è il seguente:

```
clear
Im=imread('lioness60.bmp');
ImD = im2double(Im);
cform = makecform('srgb2lab'); %creates the color transformation
structure C that defines the color space conversion specified by
'srgb2lab'.
labIm = applycform(ImD ,cform); % Conversion to CIELAB
[rows, cols, dim]=size(labIm);
levels=1; %original image
```

```

Temp = 0;TempL=0;TempA=0;TempB=0;
contrastTemp = 0;
contrastOfLevel = 0;
contrast = 0;
while min(size(labIm,1), size(labIm,2)) > 25
    for r=2:rows-1
        for c=2:cols-1
            TempL = 0.50*((abs(labIm(r,c,1)-labIm(r-1,c-
1,1))+abs(labIm(r,c,1)-labIm(r-1,c,1))+ abs(labIm(r,c,1)-labIm(r-
1,c+1,1))+abs(labIm(r,c,1)-labIm(r,c+1,1))+ abs(labIm(r,c,1)-
labIm(r+1,c+1,1))+abs(labIm(r,c,1)-labIm(r+1,c,1))+
abs(labIm(r,c,1)-labIm(r+1,c-1,1))+abs(labIm(r,c,1)-labIm(r,c-
1,1)))))/8;
            TempA = 0.25*((abs(labIm(r,c,2)-labIm(r-1,c-
1,2))+abs(labIm(r,c,2)-labIm(r-1,c,2))+ abs(labIm(r,c,2)-labIm(r-
1,c+1,2))+abs(labIm(r,c,2)-labIm(r,c+1,2))+ abs(labIm(r,c,2)-
labIm(r+1,c+1,2))+abs(labIm(r,c,2)-labIm(r+1,c,2))+
abs(labIm(r,c,2)-labIm(r+1,c-1,2))+abs(labIm(r,c,2)-labIm(r,c-
1,2)))))/8;
            TempB = 0.25*((abs(labIm(r,c,3)-labIm(r-1,c-
1,3))+abs(labIm(r,c,3)-labIm(r-1,c,3))+ abs(labIm(r,c,3)-labIm(r-
1,c+1,3))+abs(labIm(r,c,3)-labIm(r,c+1,3))+ abs(labIm(r,c,3)-
labIm(r+1,c+1,3))+abs(labIm(r,c,3)-labIm(r+1,c,3))+
abs(labIm(r,c,3)-labIm(r+1,c-1,3))+abs(labIm(r,c,3)-labIm(r,c-
1,3)))))/8;
            Temp = TempL + TempA + TempB;
            contrastTemp = contrastTemp + Temp;
        end
    end
    contrastOfLevel = contrastTemp / (rows*cols);
    contrast = contrast + contrastOfLevel;
    labIm = impyramid(labIm, 'reduce'); % Undersampling for next
level
    levels=levels+1;
    [rows, cols, dim]=size(labIm);
    Temp=0;TempL=0;TempA=0;TempB=0; % Clearing all values of
earlier level to compute values for new level.
    contrastTemp=0;

```

```

contrastOfLevel=0;
end
contrast=contrast/(levels-1)

```

3.4 Misura di contrasto sui tre canali dello spazio RGB

In questa implementazione la misura di contrasto è applicata su tutti i canali dello spazio RGB dell'immagine. In figura 7 viene mostrato il contrasto misurato (CM) per il leone e per la leonessa.

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
Lion	0.0205	0.024	0.0278	0.0318	0.0362	0.0411	0.0465	0.0522	0.0572	0.0618	0.0661	0.0704	0.0746
Lioness	0.0167	0.0195	0.0226	0.026	0.0295	0.0335	0.0379	0.0417	0.0451	0.0483	0.0513	0.0538	0.0561

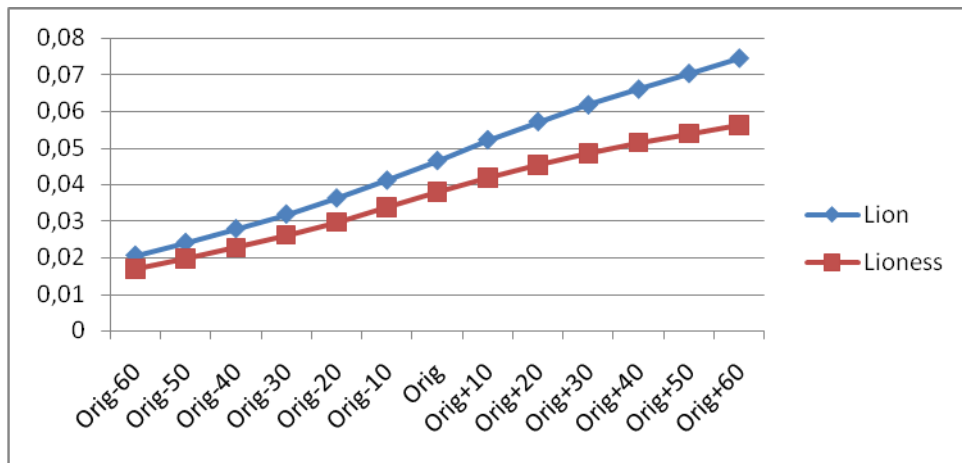


Figura 7: Leone vs Leonessa con 13 differenti variazioni di contrasto, nello spazio RGB, pesi equivalenti.

In questa misura i canali RGB, sono stati pesati in modo uniforme: 0.33R, 0.33G e 0.33B. Di nuovo, il leone ha CM più alti che la leonessa, tuttavia vediamo come, in entrambe le immagini, i valori sono notevolmente più bassi rispetto ai casi precedenti (il valore di CM più alto per il leone è 0.0746 mentre per la leonessa è 0.0561).

Il codice per il calcolo di questa implementazione è il seguente:

```
clear
Im=imread('lioness60.bmp');
ImD = im2double(Im);
rgbIm=ImD; % Image is already in RGB space
[rows, cols, dim]=size(rgbIm);
levels=1; %original image
Temp = 0;TempR=0;TempG=0;TempB=0;
contrastTemp = 0;
contrastOfLevel = 0;
contrast = 0;
while min(size(rgbIm,1), size(rgbIm,2)) > 25
    for r=2:rows-1
        for c=2:cols-1
            TempR = 0.33*((abs(rgbIm(r,c,1)-rgbIm(r-1,c-1,1))+abs(rgbIm(r,c,1)-rgbIm(r-1,c,1))+ abs(rgbIm(r,c,1)-rgbIm(r-1,c+1,1))+abs(rgbIm(r,c,1)-rgbIm(r,c+1,1))+ abs(rgbIm(r,c,1)-rgbIm(r+1,c+1,1))+abs(rgbIm(r,c,1)-rgbIm(r+1,c,1))+abs(rgbIm(r,c,1)-rgbIm(r+1,c-1,1))+abs(rgbIm(r,c,1)-rgbIm(r,c-1,1)))))/8;
            TempG = 0.33*((abs(rgbIm(r,c,2)-rgbIm(r-1,c-1,2))+abs(rgbIm(r,c,2)-rgbIm(r-1,c,2))+ abs(rgbIm(r,c,2)-rgbIm(r-1,c+1,2))+abs(rgbIm(r,c,2)-rgbIm(r,c+1,2))+ abs(rgbIm(r,c,2)-rgbIm(r+1,c+1,2))+abs(rgbIm(r,c,2)-rgbIm(r+1,c,2))+abs(rgbIm(r,c,2)-rgbIm(r+1,c-1,2))+abs(rgbIm(r,c,2)-rgbIm(r,c-1,2)))))/8;
            TempB = 0.33*((abs(rgbIm(r,c,3)-rgbIm(r-1,c-1,3))+abs(rgbIm(r,c,3)-rgbIm(r-1,c,3))+ abs(rgbIm(r,c,3)-rgbIm(r-1,c+1,3))+abs(rgbIm(r,c,3)-rgbIm(r,c+1,3))+ abs(rgbIm(r,c,3)-rgbIm(r+1,c+1,3))+abs(rgbIm(r,c,3)-rgbIm(r+1,c,3))+abs(rgbIm(r,c,3)-rgbIm(r+1,c-1,3))+abs(rgbIm(r,c,3)-rgbIm(r,c-1,3)))))/8;
            Temp = TempR + TempG + TempB;
            contrastTemp = contrastTemp + Temp;
        end
    end
contrastOfLevel = contrastTemp / (rows*cols);
```

```

contrast = contrast + contrastOfLevel;
rgbIm = impyramid(rgbIm, 'reduce'); % Undersampling for next
level
levels=levels+1;
[rows, cols, dim]=size(rgbIm);
Temp=0;TempR=0;TempG=0;TempB=0; % Clearing all values of
earlier level to compute values for new level.
contrastTemp=0;
contrastOfLevel=0;
end
contrast=contrast/(levels-1)

```

3.5 Misura di contrasto sui tre canali dello spazio RGB con pesi differenti

In questa implementazione la misura di contrasto è applicata su tutti i canali dello spazio RGB dell'immagine. Sono stati assegnati pesi differenti ai tre canali, in particolare: 0.2126R, 0.7152G e 0.0722B, dando così maggior peso al canale verde (per il quale l'occhio umano è più sensibile).

In figura 8 viene mostrato il contrasto misurato (CM) per il leone e per la leonessa.

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
Lion	0.0206	0.0241	0.0279	0.032	0.0364	0.0413	0.0468	0.0526	0.0579	0.0629	0.0679	0.0729	0.0777
Lioness	0.0168	0.0197	0.0228	0.0261	0.0297	0.0337	0.0381	0.0421	0.046	0.0498	0.0532	0.0563	0.0591

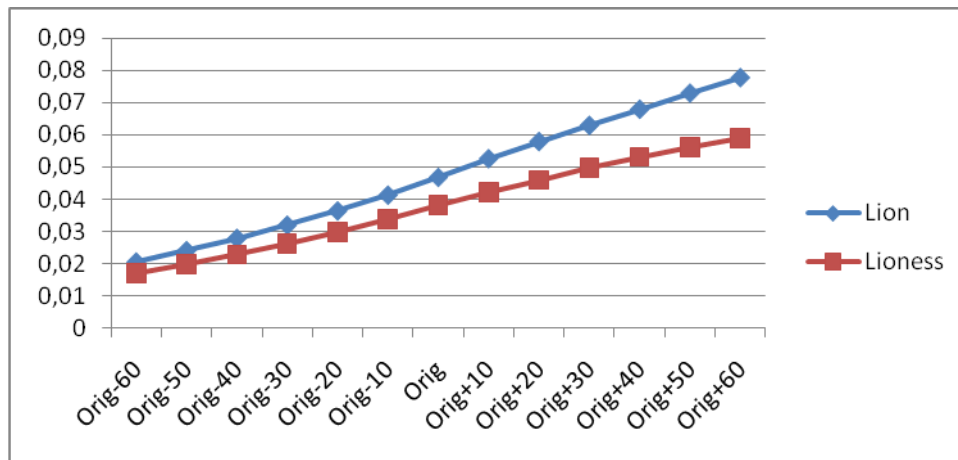


Figura 8: Leone vs Leonessa con 13 differenti variazioni di contrasto, nello spazio RGB, pesi differenti.

Il codice per il calcolo di questa implementazione è il seguente:

```
clear
Im=imread('lioness60.bmp');
ImD = im2double(Im);
rgbIm=ImD; % Image is already in RGB space
[rows, cols, dim]=size(rgbIm);
levels=1; %original image
Temp = 0;TempR=0;TempG=0;TempB=0;
contrastTemp = 0;
contrastOfLevel = 0;
contrast = 0;
while min(size(rgbIm,1), size(rgbIm,2)) > 25
    for r=2:rows-1
        for c=2:cols-1
            TempR = 0.2126*((abs(rgbIm(r,c,1)-rgbIm(r-1,c-1,1))+abs(rgbIm(r,c,1)-rgbIm(r-1,c,1))+ abs(rgbIm(r,c,1)-rgbIm(r-1,c+1,1))+abs(rgbIm(r,c,1)-rgbIm(r,c+1,1))+ abs(rgbIm(r,c,1)-rgbIm(r+1,c+1,1))+abs(rgbIm(r,c,1)-rgbIm(r+1,c,1))+ abs(rgbIm(r,c,1)-rgbIm(r+1,c-1,1))+abs(rgbIm(r,c,1)-rgbIm(r,c-1,1))))/8;
            TempG = 0.7152*((abs(rgbIm(r,c,2)-rgbIm(r-1,c-1,2))+abs(rgbIm(r,c,2)-rgbIm(r-1,c,2))+ abs(rgbIm(r,c,2)-rgbIm(r-1,c+1,2))+abs(rgbIm(r,c,2)-rgbIm(r,c+1,2))+ abs(rgbIm(r,c,2)-rgbIm(r+1,c+1,2))+abs(rgbIm(r,c,2)-rgbIm(r+1,c,2))+
```

```

abs(rgbIm(r,c,2)-rgbIm(r+1,c-1,2))+abs(rgbIm(r,c,2)-rgbIm(r,c-
1,2)))/8;

    TempB = 0.0722*((abs(rgbIm(r,c,3)-rgbIm(r-1,c-
1,3))+abs(rgbIm(r,c,3)-rgbIm(r-1,c,3))+ abs(rgbIm(r,c,3)-rgbIm(r-
1,c+1,3))+abs(rgbIm(r,c,3)-rgbIm(r,c+1,3))+ abs(rgbIm(r,c,3)-
rgbIm(r+1,c+1,3))+abs(rgbIm(r,c,3)-rgbIm(r+1,c,3)))+
abs(rgbIm(r,c,3)-rgbIm(r+1,c-1,3))+abs(rgbIm(r,c,3)-rgbIm(r,c-
1,3)))/8;

    Temp = TempR + TempG + TempB;
    contrastTemp = contrastTemp + Temp;
end
end
contrastOfLevel = contrastTemp / (rows*cols);
contrast = contrast + contrastOfLevel;
rgbIm = impyramid(rgbIm, 'reduce'); % Undersampling for next
level
levels=levels+1;
[rows, cols, dim]=size(rgbIm);
Temp=0;TempR=0;TempG=0;TempB=0; % Clearing all values of
earlier level to compute values for new level.
contrastTemp=0;
contrastOfLevel=0;
end
contrast=contrast/(levels-1)

```

4 Un confronto tra le misure di contrasto implementate

Presentiamo un sintetico confronto tra le misure di contrasto implementate. Tale confronto è solo una comparazione preliminare col solo scopo di descrivere qualitativamente le relazioni tra le varianti descritte. E' importante ricordare che tutte le misure di contrasto descritte non hanno valore assoluto, ma ciò che conta è la loro variazione rispetto al contesto considerato.

4.2 L* Vs LAB Vs LAB-Weighted

Dati riguardanti l'immagine rappresentante il leone:

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
L*	2.0565	2.4156	2.7998	3.2193	3.6752	4.1822	4.7316	5.2745	5.7613	6.2228	6.6762	7.1298	7.5597
LAB	1.1654	1.3663	1.5833	1.8202	2.0788	2.3686	2.6804	3.0412	3.4335	3.8763	4.3777	4.9454	5.5957
LABW	1.397	1.639	1.8994	2.1837	2.4937	2.84	3.2135	3.6225	4.0415	4.4923	4.9855	5.529	6.1291

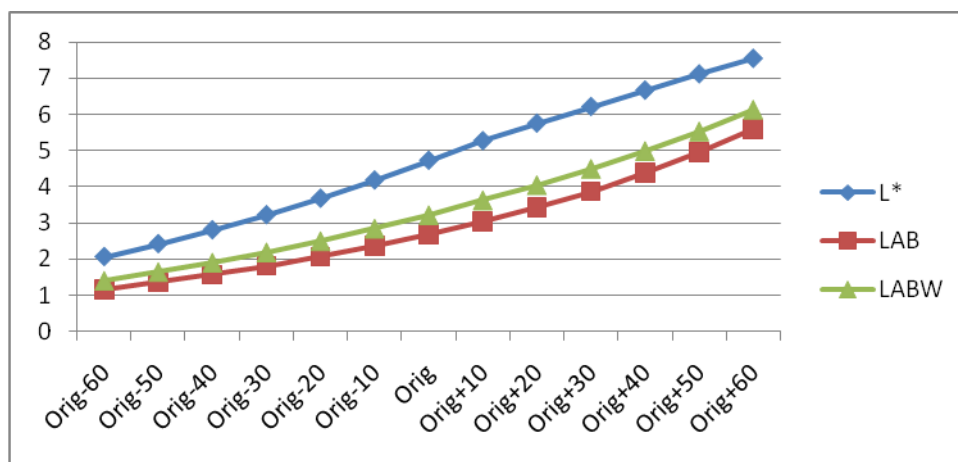


Figura 9: Leone con 13 differenti variazioni di contrasto per L*, LAB e LAB pesato (LABW)

Dati riguardanti l'immagine rappresentante la leonessa:

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
L*	1.6724	1.9597	2.2713	2.6054	2.9691	3.3693	3.806	4.1953	4.5715	4.932	5.2606	5.5514	5.8147
LAB	0.9212	1.0785	1.2478	1.4308	1.6303	1.8504	2.0912	2.3884	2.726	3.106	3.5201	3.9691	4.4658
LABW	1.116	1.3069	1.5131	1.7353	1.9774	2.2442	2.5358	2.8583	3.208	3.586	3.9819	4.3948	4.8368

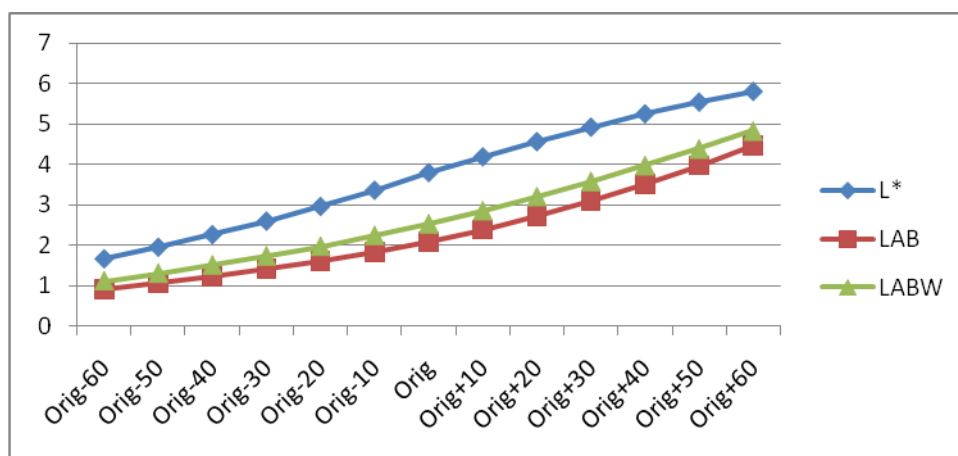


Figura 10: Leonessa con 13 differenti variazioni di contrasto per L*, LAB e LAB pesato (LABW)

Sono stati visualizzati i dati dei tre metodi utilizzati nello spazio CIELab in un unico grafico, sia per quanto riguarda l'immagine rappresentante il leone, sia per l'immagine rappresentante la leonessa. Usando solo il canale L* il contrasto misurato è più alto che negli altri casi, in entrambe le immagini, ed inoltre si ha una maggiore linearità nel grafico. Il sistema che utilizza gli stessi pesi nello spazio LAB invece ha il contrasto più basso.

4.2 RGB Vs RGB-Weighted

Dati riguardanti l'immagine rappresentante il leone:

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
RGB	0.0205	0.024	0.0278	0.0318	0.0362	0.0411	0.0465	0.0522	0.0572	0.0618	0.0661	0.0704	0.0746
RGBW	0.0206	0.0241	0.0279	0.032	0.0364	0.0413	0.0468	0.0526	0.0579	0.0629	0.0679	0.0729	0.0777

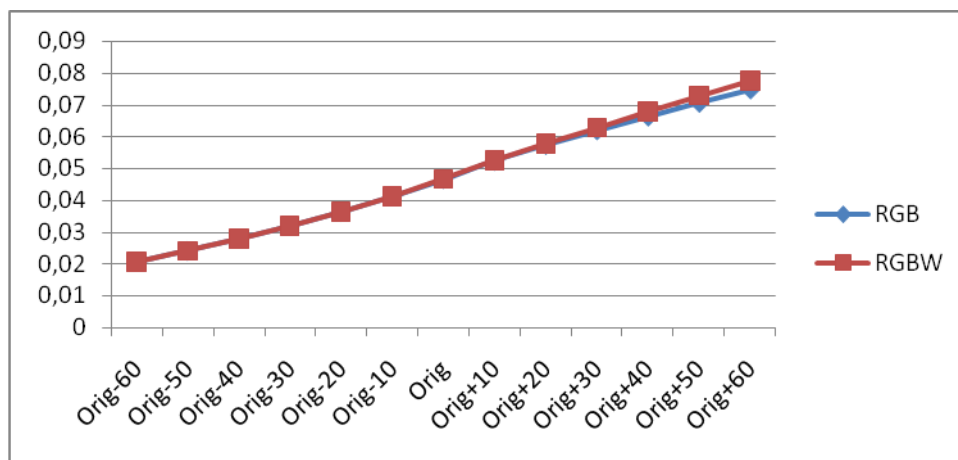


Figura 11: Leone con 13 differenti variazioni di contrasto per RGB and RGB pesato (RGBW).

Dati riguardanti l'immagine rappresentante la leonessa:

	Orig-60	Orig-50	Orig-40	Orig-30	Orig-20	Orig-10	Orig	Orig+10	Orig+20	Orig+30	Orig+40	Orig+50	Orig+60
RGB	0.0167	0.0195	0.0226	0.026	0.0295	0.0335	0.0379	0.0417	0.0451	0.0483	0.0513	0.0538	0.0561
RGBW	0.0168	0.0197	0.0228	0.0261	0.0297	0.0337	0.0381	0.0421	0.046	0.0498	0.0532	0.0563	0.0591

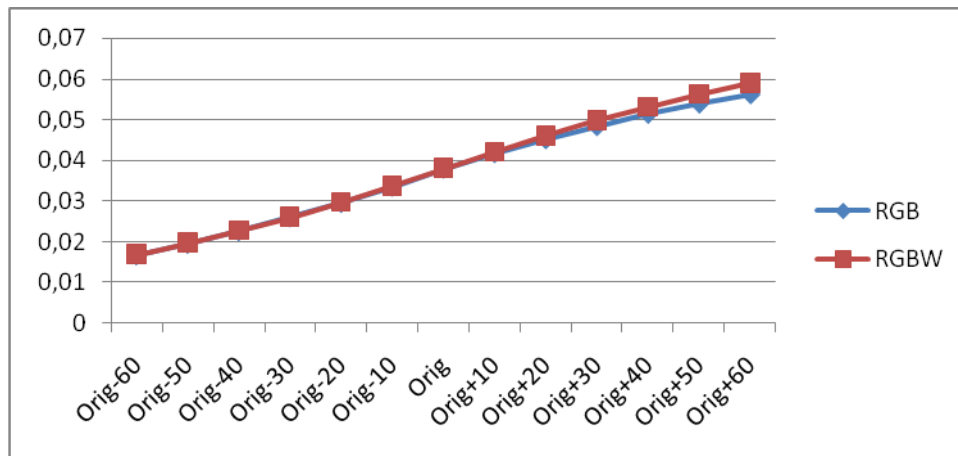


Figura 12: Leonessa con 13 differenti variazioni di contrasto per RGB and RGB pesato (RGBW).

Sono stati visualizzati sullo stesso grafico i dati riguardanti il contrasto misurato nello spazio RGB, con pesi equivalenti o differenti, per le due immagini che si stanno considerando. Si può notare che il CM è quasi identico in entrambi i casi. Soltanto alla fine della serie il CM calcolato sul RGB pesato (RGBW) è leggermente maggiore.

4. Conclusioni

Abbiamo presentato le implementazioni di alcune funzioni per il calcolo del contrasto percepito per immagini digitali.

Le misure presentate lavorano sia sulla sola luminanza che sull'informazione cromatica. In quest'ultimo caso abbiamo proposto diversi pesi per variare l'importanza relativa di ciascun canale cromatico. Altre modifiche potranno essere fatte in futuro, tenendo in considerazione ulteriori test.