

UNIVERSITÀ DI PISA

Scuola di Ingegneria
Corso di laurea in Ingegneria Informatica
A.A. 2018/2019

Sviluppo di una rete neurale per il riconoscimento
delle differenze percettive su un set di immagini per
trovare la minima risoluzione necessaria alla loro
individuazione

Candidato
Davide Cocomini

Relatori
Beatrice Lazzerini
Francesco Pistolesi

Indice

1	Introduzione	3
2	Colorimetria	4
2.1	Lo spettro visibile	4
2.1.1	Percezione del colore	5
2.2	Spazio colore	6
2.2.1	Modello CIE XYZ	6
2.2.2	Modello CIE L*a*b*	7
2.3	Risoluzione e compressione	8
2.3.1	Esempi di risoluzione	9
3	Reti Neurali	10
3.1	Reti neurali e cervello umano	10
3.1.1	Dal neurone biologico a quello artificiale	10
3.2	Perceptron	11
3.2.1	Addestramento del perceptron	12
4	Svolgimento	13
4.1	Creazione del dataset	13
4.1.1	Codice C++	14
4.1.2	Esempi di applicazione dei filtri	15
4.2	Features extraction	16
4.2.1	Preparazione degli input e degli output	16
4.2.2	Estrazione dei dati dalle immagini	16
4.2.3	Codice C++	18
4.2.4	Preparazione degli output	19
4.2.5	Esempio di risposta della rete	20
4.2.6	Esempi di valutazioni da parte dell'osservatore	21
4.3	Considerazioni preliminari	22
4.3.1	Analisi dei giudizi dell'osservatore	22
4.3.2	Calcolo del Lab-SSIM	23
4.3.3	Analisi dei risultati ottenuti con il Lab-SSIM	24
4.4	Addestramento della rete	25
4.4.1	Valutazione delle prestazioni	26
4.4.2	Features selection	27
5	Test	28
6	Conclusioni	29
7	Ringraziamenti	30

1 Introduzione

L'individuazione delle differenze percettive, tra due o più immagini, è uno dei grandi problemi dell'industria della colorimetria moderna che ha la necessità di emulare il comportamento umano nella percezione dei colori. È infatti spesso necessario riprodurre un materiale o un tessuto partendo da un originale, nel tentativo di emularlo il più fedelmente possibile. Per fare ciò, sono stati realizzati macchinari in grado di acquisire immagini ad alta definizione dei soggetti originali e di quelli riprodotti così da poterli confrontare attraverso indici matematici più o meno precisi, ad esempio SSIM, nel tentativo di identificare eventuali differenze percettive. Ovviamente, sia per l'acquisizione che per il confronto tra immagini ad alta definizione, sono necessari tempi e costi non indifferenti che sarebbero abbattuti se questi processi potessero avvenire a risoluzioni più basse. Lo scopo di questa ricerca è quindi quello di realizzare una rete neurale che sia in grado di identificare le differenze tra le immagini e sfruttarla per capire qual è la minima risoluzione che le immagini devono possedere affinché queste differenze siano individuabili.

2 Colorimetria

La colorimetria è la disciplina che si occupa di normalizzare la misurazione del colore attraverso lo studio dei modelli di colore.

2.1 Lo spettro visibile

Tutte le colorazioni percepite dall'occhio umano compongono lo "spettro del visibile" che si trova nella parte centrale dello spettro ottico, il quale comprende anche i raggi infrarossi e quelli ultravioletti. La colorimetria utilizza la percentuale della luce incidente che è stata riflessa compresa nell'intervallo del visibile (400-700 nm) per descrivere il colore dell'oggetto. Ciascun oggetto colorato viene pertanto definito da una curva di riflettanza, similmente alle impronte digitali nell'uomo.

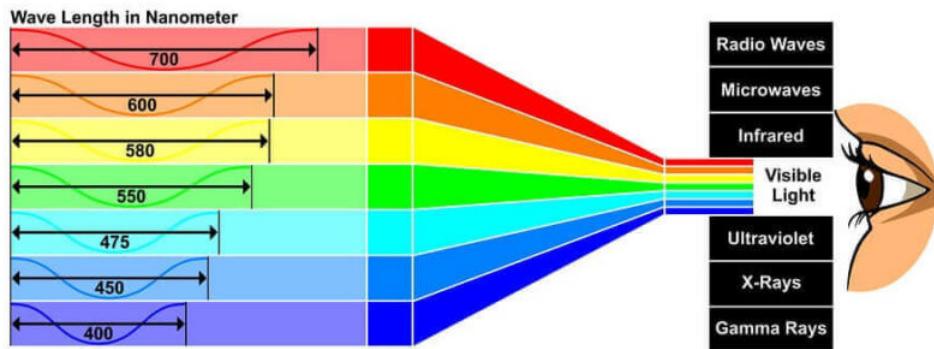


Figura 1: "Lo spettro visibile"

2.1.1 Percezione del colore

Il colore nasce dalla luce. La luce che colpisce un oggetto viene parzialmente assorbita a seconda del materiale che lo compone. 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. Proprio perché nella percezione del colore vengono coinvolte componenti biologiche, ciascun individuo "percepisce" il colore in modo differente. Tale fenomeno può essere ricondotto sia al fatto che non esistono mai due occhi uguali tra loro sia al fatto che l'interpretazione del colore varia da individuo ad individuo. Perfino la stessa persona può percepire differentemente il colore in momenti diversi ed in base allo stato d'animo. È quindi molto complesso definire in maniera oggettiva un colore.

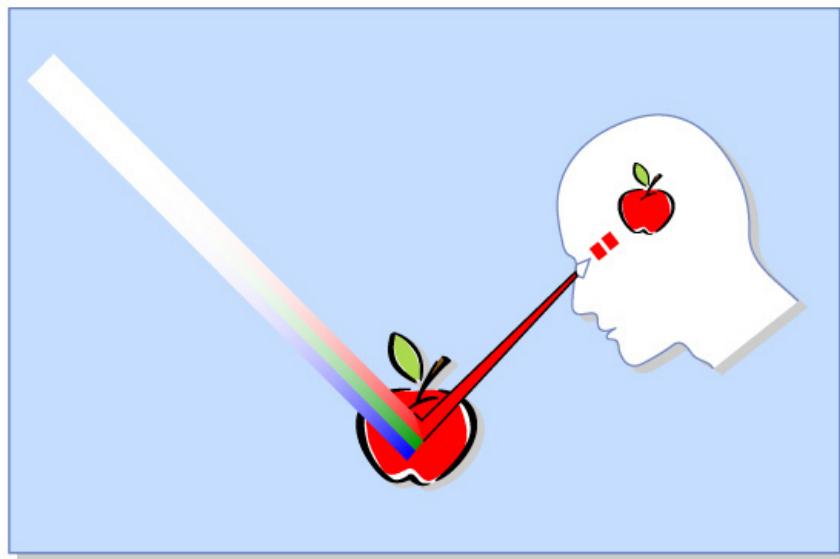


Figura 2: "Percezione del colore"

2.2 Spazio colore

Gli spazi colore sono schemi di rappresentazione dei colori. Ognuno di essi si basa su alcuni parametri e facendoli variare riesce a rappresentare un numero più o meno grande di colori. Esistono numerosi spazi colore che, in base alle loro peculiarità, si prestano meglio ad uno specifico utilizzo piuttosto che ad un altro.

2.2.1 Modello CIE XYZ

Il modello CIE XYZ rappresenta tutti i colori caratterizzati da tre parametri: luminosità, tinta e purezza, rappresentati attraverso un solido. Del solido viene solitamente rappresentata soltanto la sezione secondo il piano XY, in cui X ed Y indicano la cromaticità (tinta e purezza) mentre la luminosità non viene rappresentata.

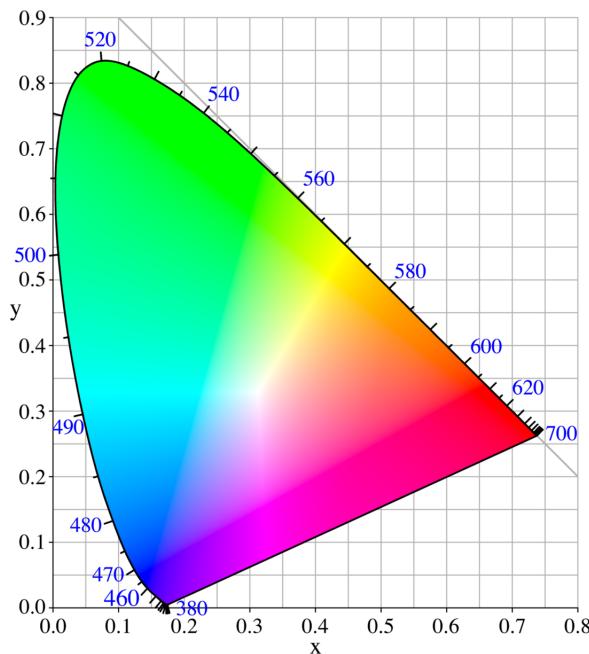


Figura 3: "Spazio colore CIE XYZ"

Questo spazio colore si basa sui valori di tristimolo, cioè dei parametri che definiscono il modo in cui l'essere umano percepisce i colori. I valori tristimolo di un colore con una distribuzione di potenza spettrale $I(\lambda)$ sono date in termini di un osservatore standard da:

$$\mathbf{X} = \int_0^{\infty} I(\lambda) \bar{x}(\lambda) d\lambda; \mathbf{Y} = \int_0^{\infty} I(\lambda) \bar{y}(\lambda) d\lambda; \mathbf{Z} = \int_0^{\infty} I(\lambda) \bar{z}(\lambda) d\lambda;$$

2.2.2 Modello CIE L*a*b*

Il modello CIE L*a*b*, è un'evoluzione del modello CIE XYZ. È un modello tridimensionale che si sviluppa lungo tre assi ortogonali. Due assi sul piano orizzontale riguardano la cromaticità: l'asse **a** si estende nel verde (-a) al rosso (+a) e l'asse **b** dal blu (-b) al giallo (+b); un asse verticale riguarda la luminosità o luminanza **L** che diminuisce dal basso verso l'alto. Questo particolare spazio colore include tutti i colori percepibili e quindi anche tutto il gamut degli spazi RGB e CMYK ed è indipendente dal dispositivo che li rappresenta. Essendo quindi molto più simile al modo in cui l'essere umano riesce a percepire i colori, questo spazio colore si presta particolarmente bene quando si devono individuare le differenze percettive all'interno di più immagini.

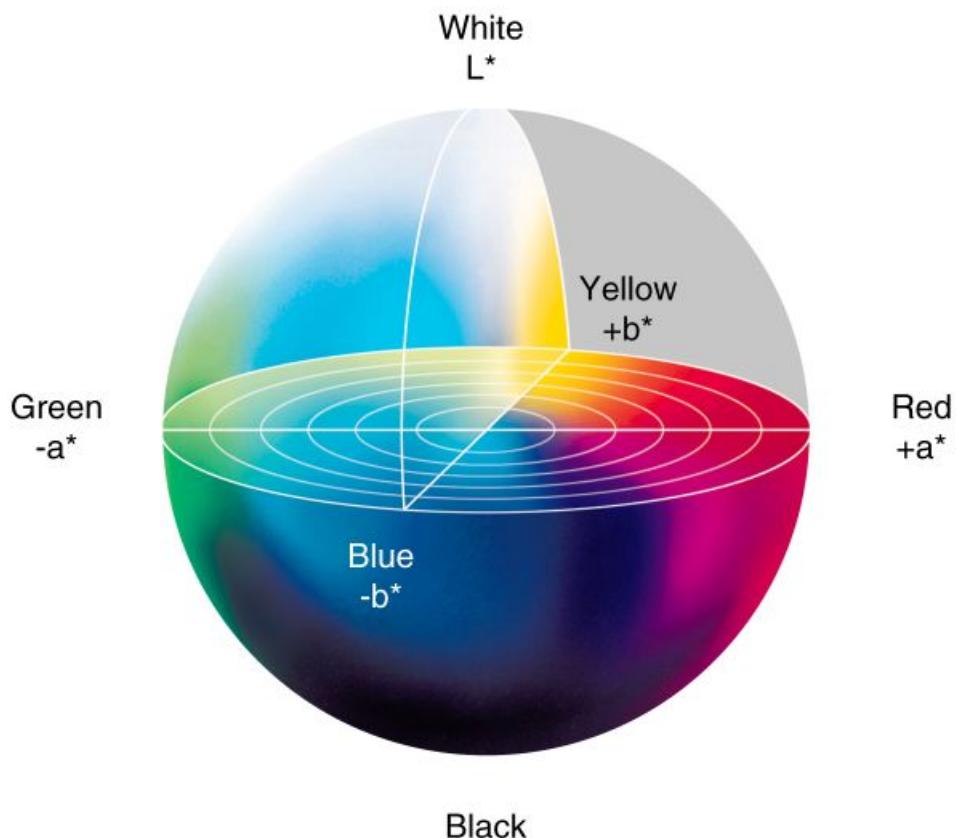


Figura 4: "Spazio colore CIE L*a*b*"

2.3 Risoluzione e compressione

La risoluzione di un’immagine corrisponde al numero di pixel per pollice che contiene, indicata con il termine in DPI (punti per pollice). Un maggior numero di dpi si traduce in una maggiore quantità di informazioni e quindi in una migliore qualità dell’immagine. Ovviamente, maggiore è il numero di informazioni che un’immagine contiene, maggiore sarà il peso del file. Per questa ragione sono state sviluppate alcune tecniche di compressione tra cui quella JPEG. Questa tecnica stabilisce due metodi di compressione di base, uno basato sull’uso della trasformata discreta del coseno, con compressione di tipo ”lossy” e cioè con perdita di informazione, e l’altro sull’uso di un metodo predittivo con compressione di tipo lossless, cioè senza perdita di informazione. La perdita di informazioni durante la compressione può compromettere drasticamente la qualità dell’immagine.

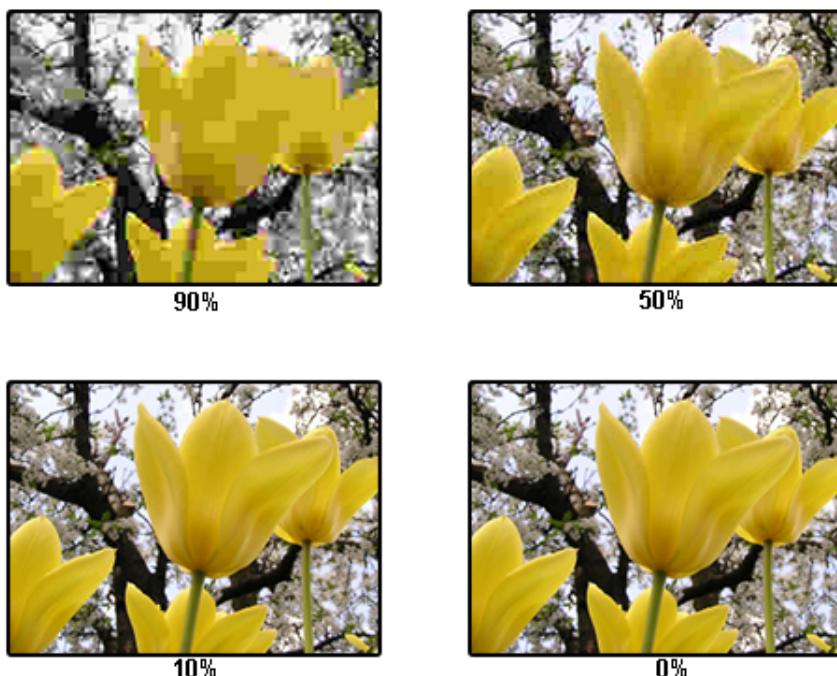


Figura 5: ”Esempio di compressione JPEG”

2.3.1 Esempi di risoluzione

Ai fini della nostra ricerca, sono state prese in considerazione 5 risoluzioni diverse. Ad ogni risoluzione è associata una coppia di valori numerici rappresentativi della larghezza e dell'altezza dell'immagine:

- 4K → 3840x2160;
- 3K → 3000x2000;
- 1080p → 2048x1080;
- 720p → 1280x720;
- 480p → 544x480;

Il prodotto di questi valori rappresenta il numero di pixel presenti in un'immagine ad una data risoluzione. Ad esempio, a 4K avremo $3840 * 2160 = 8.294.400$, mentre a 480p il numero di pixel sarà $544 * 480 = 261.120$. Quindi un'immagine a 4K userà un numero di pixel decisamente superiore rispetto ad una a 480p e questo le consentirà di rappresentare l'immagine con un numero di dettagli superiore e quindi ad una qualità migliore.

3 Reti Neurali

3.1 Reti neurali e cervello umano

Le reti neurali artificiali sono nate per emulare attività tipiche del cervello umano. Risultano quindi molto utili per risolvere problemi di riconoscimento delle immagini o del linguaggio umano. Per riuscire ad emulare questi comportamenti ci si è quindi ispirati al funzionamento del cervello umano. Nel sistema nervoso esistono miliardi di neuroni. Un neurone è formato da un corpo cellulare e da molti prolungamenti ramificati, detti dendriti, attraverso i quali il neurone riceve segnali elettrici da altri neuroni. Ogni neurone ha anche un prolungamento filamentoso chiamato assone. All'estremità l'assone si ramifica formando terminali attraverso i quali i segnali elettrici vengono trasmessi ad altre cellule. Tra un terminale di un assone e la cellula ricevente esiste uno spazio. I segnali superano questo spazio per mezzo di sostanze chimiche dette neurotrasmettitori. Il punto di connessione tra terminale e dendrite è detto sinapsi. Un neurone si “attiva”, cioè trasmette un impulso elettrico lungo il suo assone quando si verifica una differenza di potenziale elettrico tra l'interno e l'esterno della cellula. L'impulso elettrico provoca la liberazione di un neurotrasmettore dai terminali dell'assone, che a loro volta possono influenzare altri neuroni.

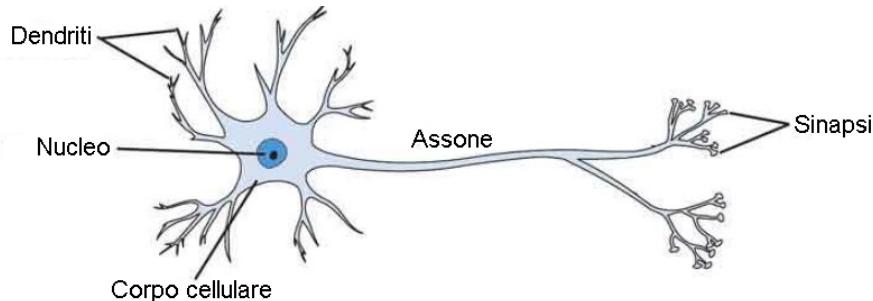


Figura 6: "Neurone biologico"

3.1.1 Dal neurone biologico a quello artificiale

Per riprodurre artificialmente il cervello umano occorre realizzare una rete di elementi molto semplici in grado di imparare e generalizzare. Tipicamente, il neurone artificiale ha molti ingressi ed una sola uscita. Per determinare la conducibilità e quindi l'importanza del canale di ingresso, ogni neurone ha associato un peso. L'attivazione del neurone è una funzione della somma pesata degli ingressi.

3.2 Perceptron

Il metodo più usato per addestrare una rete neurale consiste nel presentare in ingresso alla rete un insieme di esempi (training set). La risposta fornita dalla rete per ogni esempio viene confrontata con la risposta desiderata, si valuta la differenza (errore) fra le due e, in base a tale differenza, si aggiustano i pesi cercando di ottenere il risultato desiderato.

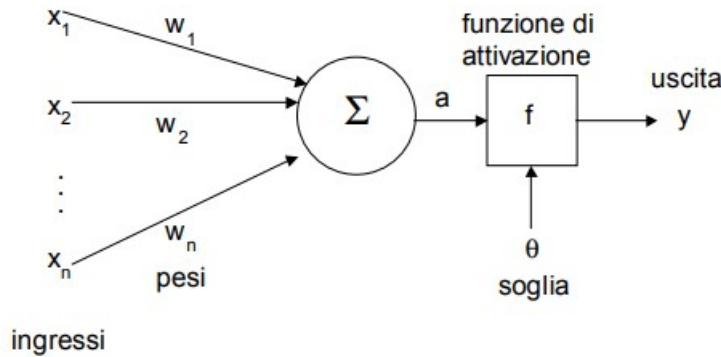


Figura 7: "Modello di neurone"

Abbiamo n canali di ingresso x_1, \dots, x_n , a ciascuno dei quali è associato un peso. I pesi w_i sono numeri reali che riproducono le sinapsi. Se $w_i > 0$, il canale è detto eccitatorio, se $w_i < 0$, il canale è inibitorio. Il valore assoluto di un peso rappresenta la forza della connessione. L'uscita, cioè il segnale con cui il neurone trasmette la sua attività all'esterno, è calcolata applicando la funzione di attivazione alla somma pesata degli ingressi:

$$y = f\left(\sum_{i=1}^n w_i x_i\right)$$

3.2 Perceptron

Un perceptron è una rete neurale ad un singolo livello che si dimostra molto utile nei problemi di classificazione. In generale è composto da una serie x_1, \dots, x_n di input a cui sono associati dei pesi w_1, \dots, w_n . Vi è poi una funzione che si occupa di sommare i prodotti dei valori di input e dei rispettivi pesi. Il risultato passa infine per una funzione di attivazione che genera l'output.

3.2.1 Addestramento del perceptron

Per addestrare un perceptron bisogna realizzare un training set adeguato per il problema che si vuole risolvere. Dopo aver predisposto il dataset è necessario seguire i seguenti passi:

1. si inizializzano i pesi w_i con valori casuali;
2. si presenta alla rete un ingresso x_k insieme al valore t_k desiderato in uscita;
3. si calcola la risposta y_k della rete e si aggiornano i pesi mediante la delta rule;
4. si ripete il ciclo dal passo 2, finché la risposta della rete non risulti soddisfacente.

In particolare la delta rule è una regola usata per aggiustare i valori dei pesi di un neurone andando alla ricerca di quelli più adeguati. Dati t ed y , rispettivamente, l'uscita desiderata e l'uscita neurale, l'errore δ è dato da:

$$\delta = t - y$$

Fissato un numero reale η compreso tra 0 e 1 detto learning rate, la delta rule stabilisce che la variazione del generico peso Δw_i è:

$$\Delta w_i = \eta \delta x_i$$

Dopo l'addestramento la rete viene testata controllandone il comportamento su un insieme di dati, detto test set, non utilizzati durante la fase di training. La fase di test ha quindi lo scopo di valutare la capacità di generalizzazione della rete neurale.

4 Svolgimento

4.1 Creazione del dataset

Per poter realizzare la rete è necessario creare un dataset significativo affinché questa possa essere addestrata e testata. Per fare ciò sono state prese in considerazione 40 immagini ad una risoluzione di 4K sulle quali sono state successivamente fatte delle elaborazioni.

In particolare le elaborazioni sono state:

1. Per ogni immagine nel dataset iniziale è stata generata un'immagine in una risoluzione minore o uguale (4K, 3K, 1080p, 720p, 480p).
2. Per ogni immagine generata al punto 1 sono stati applicati tre filtri cromatici ciascuno a tre intensità diverse.

Tra cui due filtri che coinvolgono indistintamente tutte le zone dell'immagine, ed un filtro che altera solo le zone di una specifica tonalità di colore, in questo caso, quelle del blu. Questo procedimento ha così generato 1.200 immagini, utilizzabili per addestrare e testare la rete.

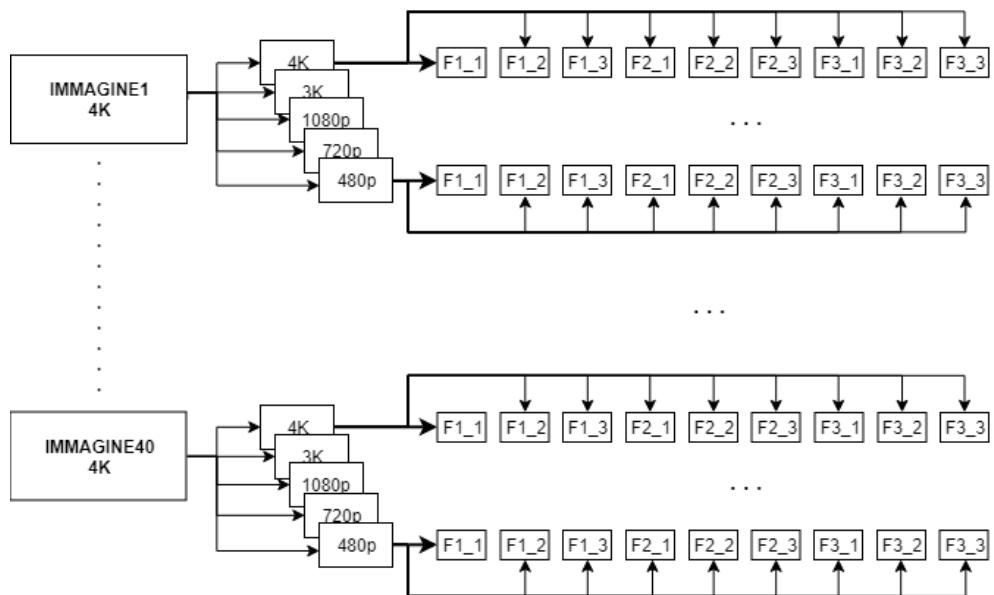


Figura 8: "Generazione del dataset"

4.1 Creazione del dataset

4.1.1 Codice C++

Per ottenere il risultato precedentemente descritto è stato sviluppato un programma in C++ sfruttando la libreria OpenCV. Per ogni file nella directory delle immagini vengono generate le immagini a risoluzioni più basse e, per ognuna di queste, vengono applicati i filtri:

```
1  for (dimension resolution : resolutions){
2      Size size(resolution.height, resolution.width);
3      Mat resizedImage;
4      resize(imageOriginal, resizedImage, size);
5      string newPath = basePath + "resized/" + resolution.name + "/" +
6          fileName + ".tif";
7
8      if (resolution.name.compare("4K") == 0)
9          copyFile(filePath, newPath);
10     else
11         imwrite(newPath, resizedImage);
12     for (int intensityF1 = 6, double intensityF2 = 1.1, int intensityF3 = 5;
13          intensityF1 <= 56, intensityF2 <= 1.5; intensityF3 <= 15;
14          intensityF1 += 25, intensityF2 += 0.2, intensityF3 += 5)
15     {
16         Mat imageFiltered = applyFilter(newPath, 1, intensityF1);
17         string newPathFiltered = basePath + "resized/" + resolution.name +
18             "/filters/" + explode(fileName, '_')[0] + "_B" + to_string(
19                 nameCounter) + ".tif";
20         imwrite(newPathFiltered, imageFiltered);
21
22         imageFiltered = applyFilter(newPath, 2, intensityF2);
23         newPathFiltered = basePath + "resized/" + resolution.name + "/
24             filters/" + explode(fileName, '_')[0] + "_B" + to_string(
25                 nameCounter+3) + ".tif";
26         imwrite(newPathFiltered, imageFiltered);
27
28         nameCounter++;
29     }
30 }
```

4.1 Creazione del dataset

4.1.2 Esempi di applicazione dei filtri

Di seguito vengono riportati alcuni esempi di immagini generate con il precedente procedimento le cui differenze sono facilmente percepibili ad occhio.

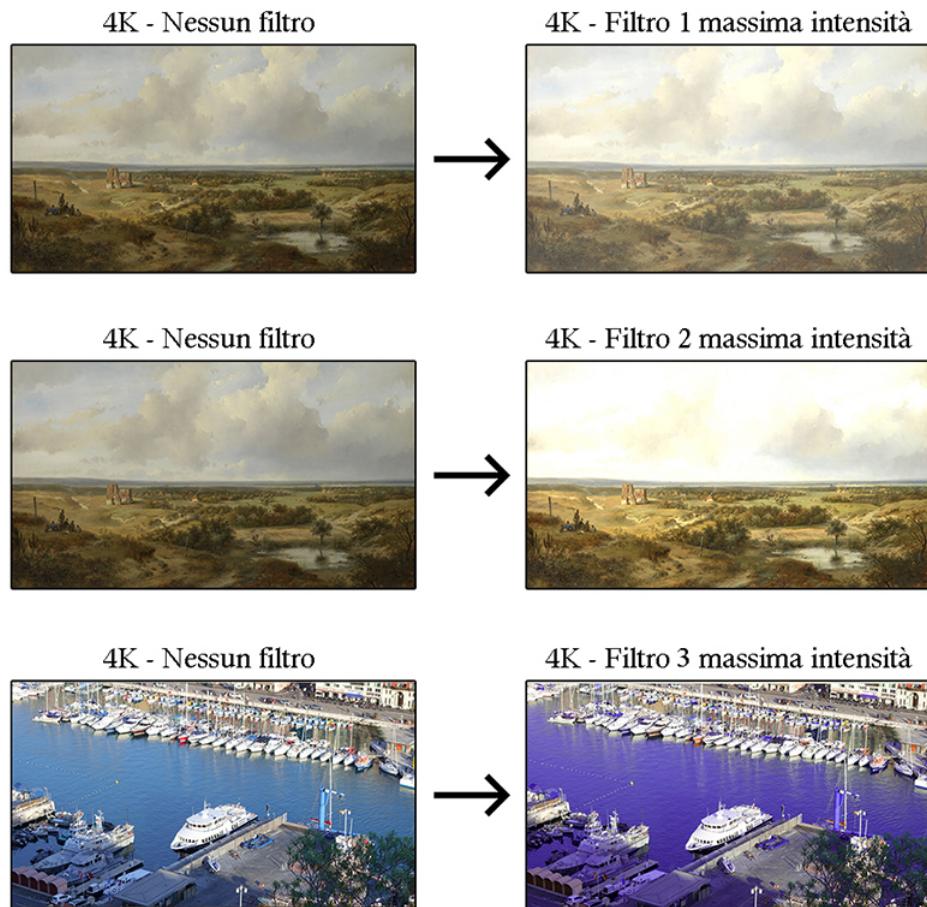


Figura 9: "Esempi di applicazione dei filtri"

4.2 Features extraction

Per addestrare la rete si è deciso di usare un tipo di addestramento detto "supervisionato". L'addestramento supervisionato è una tecnica di apprendimento automatico che mira a istruire una rete in modo da consentirgli di elaborare automaticamente previsioni sui valori di uscita di un sistema rispetto ad un input sulla base di una serie di esempi ideali, costituiti da coppie di input e di output, che gli vengono inizialmente forniti. L'estrazione delle informazioni dal dataset da fornire come input alla rete, viene detta Features Selection.

4.2.1 Preparazione degli input e degli output

Affinché si possa procedere con l'addestramento, è necessario fornire alla rete dei dati che possa comprendere, in grado di rappresentare tutte le immagini da confrontare (input). Dobbiamo inoltre fornire delle valutazioni effettuate da un osservatore umano (output), che indichino la somiglianza tra le varie coppie di immagini da confrontare. Studiando questi dati la rete sarà in grado di emulare il comportamento umano nell'identificazione delle differenze percettive tra due immagini.

4.2.2 Estrazione dei dati dalle immagini

I dati di input vengono estratti dalle immagini nel nostro dataset. In particolare, questi dati faranno riferimento a zone delle immagini sufficientemente grandi da poter permettere alla rete di comprenderne non solo il colore, ma anche la struttura, distinguendone forme e dettagli complessivi. Per fare ciò, tutte le immagini sono state suddivise in 9 zone di dimensione uguale e per ognuna di esse sono stati calcolati i valori di media e varianza. Questo procedimento viene effettuato per ognuno dei 3 canali (L, a, b). Per la media è stata usata la funzione `mean()` della libreria OpenCV, mentre per la varianza è stata implementata una funzione apposita:

```
1  double variance(Mat & m, int i, int j, int block_height, int block_width)
2  {
3      double variance = 0;
4      Mat m_tmp = m(Range(i, i + block_height), Range(j, j + block_width));
5      Mat m_squared(block_height, block_width, CV_64F);
6
7      multiply(m_tmp, m_tmp, m_squared);
8
9      double avg = mean(m_tmp)[0];
10     double avg_2 = mean(m_squared)[0];
11     var = sqrt(avg_2 - avg * avg);
12
13     return variance;
14 }
```

4.2 Features extraction

Media e varianza sono tipicamente utilizzati in colorimetria proprio per scopi simili, come ad esempio, nel calcolo dell'SSIM.

La rete dovrà avere in ingresso la lista dei valori di ogni immagine originale, associata ad ogni sua immagine filtrata. Dopo aver calcolato i valori, questi vengono inseriti in una matrice di dimensione 1800x108, dove 1800 sono il numero di coppie ($Im_i, F_j(Im_i, in)$) con Im_i immagine i -esima, F_j funzione filtro j -esima e in intensità del filtro, mentre 108 sono il numero di valori calcolati per ogni coppia.

In particolare, date X ed Y rispettivamente immagine originale e filtrata, n numero di zone su cui sono suddivise le immagini, una riga della matrice contiene:

- $\mu_{xi}(C)$: media di ogni zona dell'immagine X nei 3 canali;
- $\mu_{yi}(C)$: media di ogni zona dell'immagine Y nei 3 canali;
- $\sigma_{xi}^2(C)$: varianza di ogni zona dell'immagine X nei 3 canali;
- $\sigma_{yi}^2(C)$: varianza di ogni zona dell'immagine Y nei 3 canali;

con $i \in (1, n)$ e $C \in (L, a, b)$.

I valori sono quindi: 3 canali * 9 finestre * 4 valori = 108, per riga. La matrice di questi valori rappresenterà quindi l'input della nostra rete.

4.2 Features extraction

4.2.3 Codice C++

Per estrapolare questi valori è stata realizzata la seguente funzione C++:

```
1   for (int k = 0; k < nbBlockPerHeight; k++) {
2       for (int l = 0; l < nbBlockPerWidth; l++) {
3           {
4               int m = k * block.height;
5               int n = l * block.width;
6               // Avg values for b-channel
7               double avg_ob = mean(img_original[0](Range(k, k + block.height),
8                               Range(l, l + block_width))[0]);
9               double avg_cb = mean(img_filtered[0](Range(k, k + block.height),
10                               Range(l, l + block_width))[0]);
11              values.push_back(avg_ob);
12              values.push_back(avg_cb);
13              // Avg values for a-channel
14              double avg_oa = mean(img_original[1](Range(k, k + block.height),
15                               Range(l, l + block_width))[0]);
16              double avg_ca = mean(img_filtered[1](Range(k, k + block.height),
17                               Range(l, l + block_width))[0]);
18              values.push_back(avg_oa);
19              values.push_back(avg_ca);
20              // Avg values for L-channel
21              double avg_oL = mean(img_original[2](Range(k, k + block.height),
22                               Range(l, l + block_width))[0]);
23              double avg_cL = mean(img_filtered[2](Range(k, k + block.height),
24                               Range(l, l + block_width))[0]);
25              values.push_back(avg_oL);
26              values.push_back(avg_cL);
27              // Sigma values for b-channel
28              double sigma_oa = variance(img_original[2], m, n, block.height,
29                               block_width);
30              double sigma_ca = variance(img_filtered [2], m, n, block.height,
31                               block_width);
32              values.push_back(sigma_oa);
33              values.push_back(sigma_ca);
34              // Sigma values for a-channel
35              double sigma_ob = variance(img_original[1], m, n, block.height,
36                               block_width);
37              double sigma_cb = variance(img_filtered [1], m, n, block.height,
38                               block_width);
39              values.push_back(sigma_ob);
40              values.push_back(sigma_cb);
41              // Sigma values for L-channel
42              double sigma_oL = variance(img_original[0], m, n, block.height,
43                               block_width);
44              double sigma_cL = variance(img_filtered [0], m, n, block.height,
45                               block_width);
46              values.push_back(sigma_oL);
47              values.push_back(sigma_cL);
48          }
49      }
```

4.2.4 Preparazione degli output

Affinché la rete possa associare i valori rappresentativi di una coppia di immagini da confrontare, ad una valutazione della loro somiglianza, si sono considerati 5 livelli crescenti di affinità: L, ML, MM, MH, H. Sono state quindi analizzate 1800 coppie di immagini (40 immagini * 9 immagini filtrate * 5 risoluzioni), e ad ognuna di queste è stato assegnato un livello in questa scala. Così come un essere umano guardando una coppia di immagini, riesce ad individuarne le differenze, la rete si comporterà esattamente allo stesso modo ma guardando alle immagini come un insieme di caratteristiche espresse da valori numerici. La rete, guardando i valori rappresentativi di una coppia di immagini, cercherà di estrapolare un giudizio basandosi sulla sua esperienza pregressa. Questa esperienza gli viene fornita mostrandogli degli esempi, che sono proprio gli output prodotti dal soggetto umano. Quando la rete sarà giunta ad un responso, dovrà comunicare all'esterno il suo risultato. Per farlo utilizzerà i neuroni di output, attivando quello corrispondente al giudizio che ha ottenuto. I neuroni di uscita saranno quindi 5 e il valore di uscita della rete sarà:

- 0,0,0,0,1 → L
- 0,0,0,1,0 → ML
- 0,0,1,0,0 → MM
- 0,1,0,0,0 → MH
- 1,0,0,0,0 → H

Dove con 1 si indica il neurone attivo e con 0 il neurone non attivo.

4.2.5 Esempio di risposta della rete

Spieghiamo ciò che si vuole ottenere dalla rete neurale attraverso un esempio. Si considerano due immagini, una originale ed una con applicato il primo filtro alla massima intensità. Da esse sono stati estrapolati i dati che le rappresentano, come visto nel paragrafo 4.2.2, che sono poi stati passati in input alla nostra rete. Passando attraverso una serie di "livelli nascosti", che la rete crea per effettuare le elaborazioni necessarie all'ottenimento dell'output, la rete si accorge che le due immagini sono percettivamente molto diverse tra loro. Procede quindi attivando l'ultimo neurone di uscita.

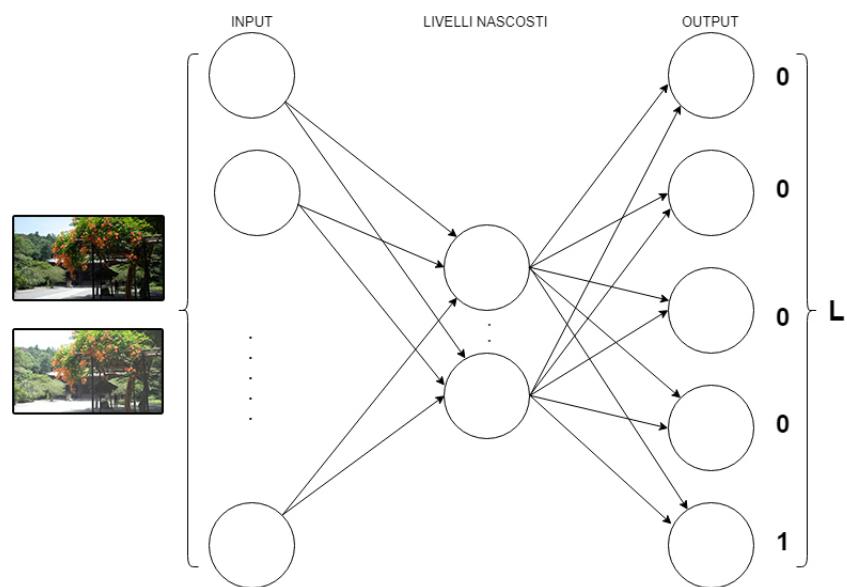


Figura 10: "Esempio di risposta della rete"

4.2.6 Esempi di valutazioni da parte dell'osservatore

L'osservatore umano, compirà lo stesso procedimento ma assegnando ad ogni coppia di immagini, un valore simbolico, come illustrato nel paragrafo 4.2.4. Di seguito vengono riportati alcuni esempi di valutazioni effettuate dall'osservatore umano:



Originale

Filtro 1 intensità minima

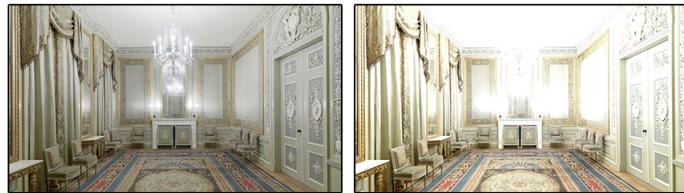
H



Originale

Filtro 3 intensità massima

MM



Originale

Filtro 2 intensità massima

L

Figura 11: "Esempi di risposta dell'osservatore umano"

4.3 Considerazioni preliminari

Prima di procedere con l'addestramento della rete, analizziamo i dati ottenuti nei precedenti paragrafi.

4.3.1 Analisi dei giudizi dell'osservatore

Prendendo in considerazione tutte le 1800 coppie di immagini originali e filtrate e associandole ai rispettivi giudizi dati dall'osservatore umano nel paragrafo 4.2.4, si nota una tendente accentuazione delle differenze cromatiche, al diminuire della risoluzione. In particolare, prendendo come esempio il giudizio dato ad una coppia di immagini a 4K e confrontandolo con quello della stessa coppia a risoluzioni inferiori, per tutte le possibili coppie nel dataset, si ottengono i seguenti risultati:

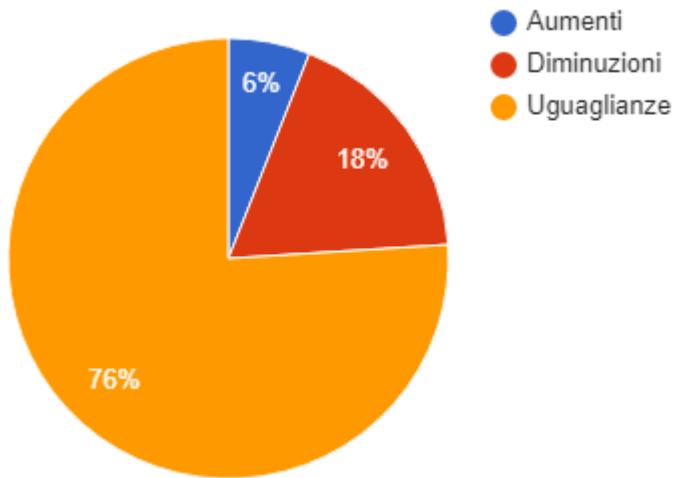


Figura 12: "Variazione della percezione umana"

Come si evince dal grafico, all'abbassarsi della risoluzione, le differenze percettive tra le immagini rimangono nella maggioranza dei casi invariate e tendono ad essere accentuate, causando così una valutazione più bassa dell'indice di similarità. Ciò accade nel 94% dei casi presi in considerazione.

4.3.2 Calcolo del Lab-SSIM

Per avvalorare questa tesi ed evitare che questo comportamento possa essere riconducibile solo alla percezione dell'osservatore umano preso in considerazione, si è deciso di effettuare lo stesso procedimento considerando però un indice matematico capace di esprimere le differenze percettive tra due immagini $L^*a^*b^*$, il Lab-SSIM. Questo indice è diverso dall'SSIM tradizionale e consiste nel calcolare, 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.

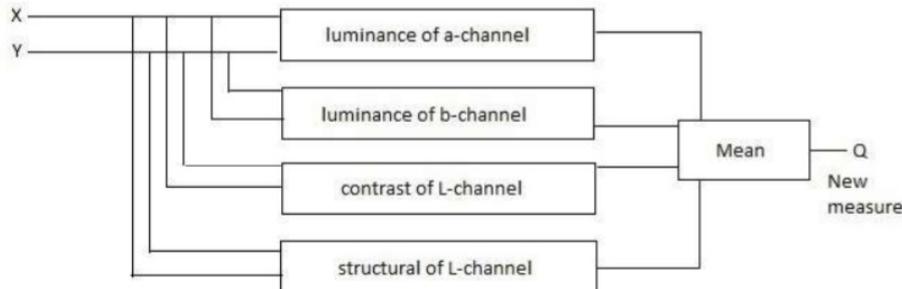


Figura 13: "Lab-SSIM"

Il Lab-SSIM si basa quindi 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:

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

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

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

in cui si hanno:

- μ_x e μ_y la media di x e y ;
- σ_x^2 e σ_y^2 la varianza di x e y ;
- σ_{xy} la covarianza di x e y ;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ due variabili per stabilizzare la divisione con il denominatore inadatto;
- c_3 costante pari a $c_2/2$.
- $k_1 = 0.01$ e $k_2 = 0.03$ predefiniti;

4.3.3 Analisi dei risultati ottenuti con il Lab-SSIM

Dopo aver calcolato il Lab-SSIM per tutte le coppie di immagini da confrontare, si passa ad analizzare i risultati, cercando di capire se, abbassando la risoluzione, le differenze continuano ad essere visibili. Si fanno quindi le stesse considerazioni del paragrafo 4.3, ma questa volta considereremo come giudizio l'indice e non quello dato dall'osservatore umano.

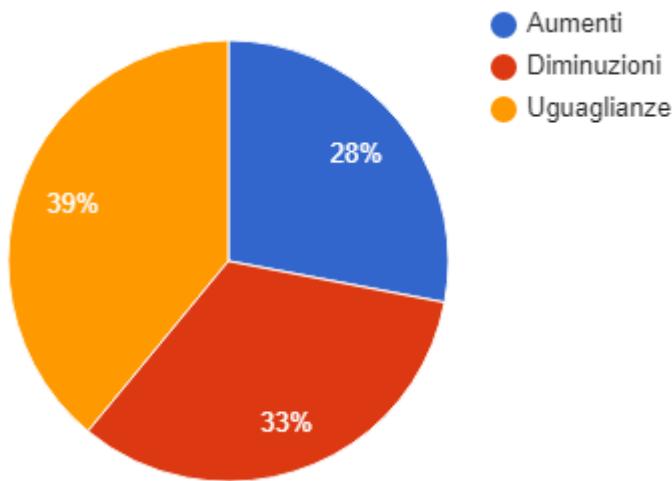


Figura 14: "Variazione del Lab-SSIM"

In questo caso, i giudizi rimangono invariati o si abbassano nel 72% dei casi. La tendenza sembra essere confermata, in quanto le uguaglianze continuano ad essere la maggioranza e le diminuzioni sono superiori agli aumenti, anche se in maniera meno marcata rispetto ai giudizi dell'osservatore umano. Da questi risultati si può già intuire che all'abbassarsi della risoluzione, le differenze percettive non solo rimangono percepibili ma tendono addirittura ad accentuarsi.

4.4 Addestramento della rete

Adesso che siamo in possesso degli input e degli output da fornire alla rete, possiamo passare ad effettuare un primo addestramento. Considerando x e t rispettivamente l'insieme delle feature estratte dalle immagini e l'insieme dei giudizi dati dall'osservatore umano, si addestra la rete utilizzando il seguente codice Matlab:

```
1 % Funzione di training
2 trainFcn = 'trainlm';
3
4 % Creazione della rete
5 hiddenLayerSize = 10;
6 net = patternnet(hiddenLayerSize, trainFcn);
7
8 % Suddivisione del dataset in training, test e validation set
9 net.divideFcn = 'dividerand';
10 net.divideMode = 'sample';
11 net.divideParam.trainRatio = 70/100;
12 net.divideParam.valRatio = 10/100;
13 net.divideParam.testRatio = 20/100;
14
15 % Funzione per il calcolo delle performance e lista di grafici da generare
16 net.performFcn = 'mse'; % mean squared error
17
18 net.plotFcns = {'plotperform', 'plottrainstate', 'ploterrhist', ...
19 'plotconfusion', 'plotroc'};
20
21 % Addestramento della rete
22 [net, tr] = train(net, x, t);
23
24 % Test della rete e performance
25 y = net(x);
26 e = gsubtract(t, y);
27 performance = perform(net, t, y)
28 tind = vec2ind(t);
29 yind = vec2ind(y);
30 percentErrors = sum(tind ~= yind) / numel(tind);
31
32 trainTargets = t .* tr.trainMask{1};
33 valTargets = t .* tr.valMask{1};
34 testTargets = t .* tr.testMask{1};
35 trainPerformance = perform(net, trainTargets, y)
36 valPerformance = perform(net, valTargets, y)
37 testPerformance = perform(net, testTargets, y)
```

Il dataset iniziale è stato diviso in 3 parti: Training Set, Validation Set e Test Set. Essi servono rispettivamente per addestrare, validare e testare la rete. In particolare, il Training Set serve per far capire alla rete quali risposte deve dare in base agli input che gli vengono forniti, il Validation Set serve per evitare il fenomeno dell'overfitting mentre il Test Set serve per valutarne le performance alla fine dell'addestramento.

4.4 Addestramento della rete

4.4.1 Valutazione delle prestazioni

Una volta completato l'addestramento, per valutarne le prestazioni, analizziamo la matrice di confusione:

Training Confusion Matrix						Validation Confusion Matrix								
Output Class	1	340 27.0%	22 1.7%	7 0.6%	2 0.2%	2 0.2%	91.2% 8.8%	1	44 24.4%	10 5.6%	2 1.1%	2 1.1%	1 0.6%	74.6% 25.4%
	2	31 2.5%	263 20.9%	16 1.3%	5 0.4%	3 0.2%	82.7% 17.3%	2	6 3.3%	25 13.9%	3 1.7%	2 1.1%	0 0.0%	69.4% 30.6%
	3	0 0.0%	9 0.7%	84 6.7%	4 0.3%	0 0.0%	86.6% 13.4%	3	0 0.0%	4 2.2%	3 1.7%	6 3.3%	2 1.1%	20.0% 80.0%
	4	0 0.0%	2 0.2%	17 1.3%	223 17.7%	6 0.5%	89.9% 10.1%	4	1 0.6%	2 1.1%	5 2.8%	25 13.9%	7 3.9%	62.5% 37.5%
	5	0 0.0%	0 0.0%	2 0.2%	10 0.8%	212 16.8%	94.6% 5.4%	5	0 0.0%	0 0.0%	2 1.1%	5 2.8%	23 12.8%	76.7% 23.3%
		91.6% 8.4%	88.9% 11.1%	66.7% 33.3%	91.4% 8.6%	95.1% 4.9%	89.0% 11.0%		86.3% 13.7%	61.0% 39.0%	20.0% 80.0%	62.5% 37.5%	69.7% 30.3%	66.7% 33.3%
		8.4%	11.1%	33.3%	8.6%	4.9%	11.0%		13.7%	39.0%	80.0%	37.5%	30.3%	33.3%
Target Class						Target Class								
Test Confusion Matrix						All Confusion Matrix								
Output Class	1	83 23.1%	16 4.4%	4 1.1%	1 0.3%	0 0.0%	79.8% 20.2%	1	467 25.9%	48 2.7%	13 0.7%	5 0.3%	3 0.2%	87.1% 12.9%
	2	22 6.1%	50 13.9%	14 3.9%	4 1.1%	0 0.0%	55.6% 44.4%	2	59 3.3%	338 18.8%	33 1.8%	11 0.6%	3 0.2%	76.1% 23.9%
	3	1 0.3%	3 0.8%	15 4.2%	4 1.1%	2 0.6%	60.0% 40.0%	3	1 0.1%	16 0.9%	102 5.7%	14 0.8%	4 0.2%	74.5% 25.5%
	4	0 0.0%	9 2.5%	8 2.2%	42 11.7%	10 2.8%	60.9% 39.1%	4	1 0.1%	13 0.7%	30 1.7%	290 16.1%	23 1.3%	81.2% 18.8%
	5	0 0.0%	1 0.3%	1 0.3%	14 3.9%	56 15.6%	77.8% 22.2%	5	0 0.0%	1 0.1%	5 0.3%	29 1.6%	291 16.2%	89.3% 10.7%
		78.3% 21.7%	63.3% 36.7%	35.7% 64.3%	64.6% 35.4%	82.4% 17.6%	68.3% 31.7%		88.4% 11.6%	81.3% 18.8%	55.7% 44.3%	83.1% 16.9%	89.8% 10.2%	82.7% 17.3%
		21.7%	36.7%	64.3%	35.4%	17.6%	31.7%		11.6%	18.8%	44.3%	16.9%	10.2%	17.3%
Target Class						Target Class								

Figura 15: "Matrice di confusione"

Ciò che si evince è che l'accuratezza della rete si aggira attorno all'80%. Questo è già un risultato discreto ma può sicuramente essere migliorato. Idealmente, si vorrebbe ottenere che il 100% dei giudizi fosse corretto e quindi la matrice avrebbe valori soltanto sulla diagonale. In questo caso tuttavia, notiamo una buona precisione sulle valutazioni estreme (H ed L) ma una più scarsa capacità di giudizio delle coppie di immagini con diffe-

renze percettive meno marcate. Ciò può essere dovuto al fatto che, le reti neurali, necessitano di un numero preciso di dati nel training set affinché possano essere addestrate in maniera corretta. Dato x numero di features in input alla rete, e h numero di neuroni presenti nei livelli nascosti, il numero n adeguato di esempi che devono essere presenti nel dataset deve essere:

$$n \geq ((x * 5) + (h * 5)) * 5$$

Nel nostro caso, le features sono l'insieme di varianze e medie calcolate per tutte le coppie (108 features), come descritto dettagliatamente nel paragrafo 4.2.2, mentre i neuroni nascosti sono fissati a 10 per ipotesi. Effettuando il calcolo di cui sopra, si ottiene:

$$n \geq ((108 * 5) + (10 * 5)) * 5 = 2.950$$

Quindi, per addestrare adeguatamente la nostra rete sarebbero necessari almeno 2.950 esempi di confronti tra coppie di immagini. Non disponendo di questi dati, ciò che possiamo fare è effettuare la features selection, cioè l'identificazione delle features maggiormente discriminanti tra le 108 ottenute durante la features extraction. Riducendo il numero di features in ingresso alla rete, anche il numero di esempi necessari sarà ridotto, potendo addestrare adeguatamente la rete anche con soli 1.800 confronti.

4.4.2 Features selection

Durante la features selection, si applica un algoritmo in grado di identificare, tra le varie features estratte durante la features extraction, quelle che contribuiscono maggiormente all'ottenimento del risultato. Conoscendo questo dato, è poi facile diminuire il numero di features in ingresso alla rete, andando a selezionare solo quelle che servono veramente alla rete. Per effettuare la features selection si comincia selezionando una feature dal dataset iniziale, essa sarà data in ingresso ad una rete che cercherà di calcolare il risultato di cui noi necessitiamo. Si osserva il risultato e si passa alla feature successiva. Dopo aver testato la rete con tutte le features prese singolarmente, si procede testandola con tutte le features prese a coppia. Si va avanti così aggiungendo di volta in volta una nuova feature ai test. Questa tipologia di features selection è detta forward. Esiste anche quella backword in cui si parte dall'insieme totale delle features e si procede eliminandone una ad ogni step. Alla fine di questo processo, si osservano i risultati della rete nei vari casi e si prendono le features che hanno permesso di ottenere i risultati migliori.

5 Test

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

6 Conclusioni

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

7 Ringraziamenti

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.