Intelligent systems

Color Comparing Neural System

Project discussion and implementation

Gioele Carignani
Alessia Guida

2017/2018

Contents

1 Part I

1.1 Introduction

The goal of this part of the project is to develop a neural network able to estimate the difference (ΔE[1]) between two colors, the master and a disturbed copy.

1.2 Initialization

Starting from a dataset, *spectra* 421*1207, a good amount of copy and master samples had to be generated. For the master colors the dataset has simply been repeated *ncopies*=10 times, hence obtaining a new dataset of 421*12070. For the copy colors some noise has been added by generating a random quantity (*noise*) for each of the ten copies:

```
noise = random('unif',1.00,1.25);
copy(1:n_wl,i) = spectraRescaled(1:n_wl,i)*noise;
```

In order to obtain always the same sequence of random numbers the RNG has been seeded with a predefined value. The random quantity has been used for translating the original SPDs as shown in Figure 1

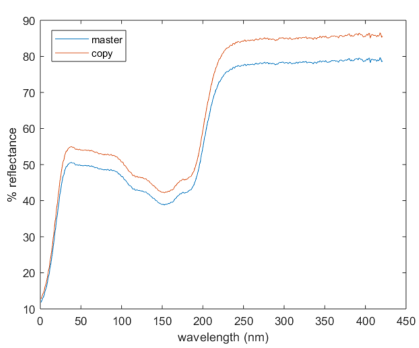

Figure 1: Master color and its translated copy

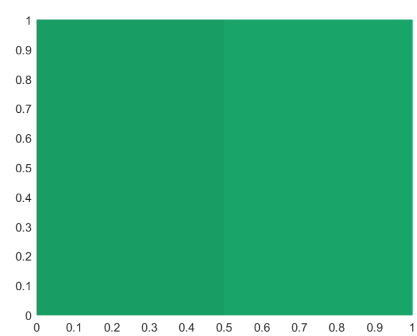

Figure 2: Visual difference between master and copy

[1]Project specifications

1.3 Targets computation

The target for this part were the differences between master colors and their copies, that can be computed by means of the following formula:

$$\Delta E^* = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2} \tag{1}$$

This could be done easily through the function *de(lab1,lab2)* provided by the *optprop* [2] package. With the selected noise parameters a mean value for the differences of 1.95 has been obtained.

1.4 Feature extraction

At this stage 2 datasets of 421*12070 each were available. These datasets would require too much time to be processed completely, hence the features that best represent the data had to be extracted. First of all, the visible spectrum has been divided into k wavelength intervals, and then each range has been associated to a certain number of features (*nfeatures*). The choice of k has been done empirically trying to find the best trade-off between information compression and information loss. A value of k=5 has been chosen since it has a high compression level but still represents the main characteristics of the sample as can be seen in Figure 3.

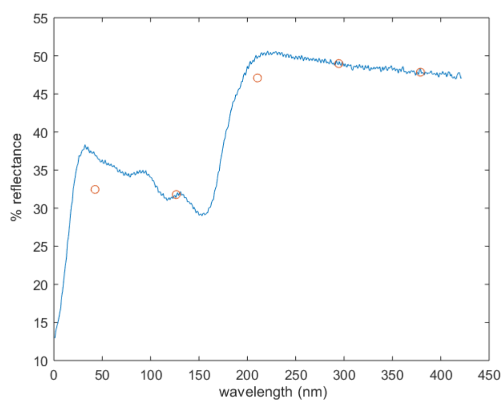

Figure 3: The line represents the SPD while the dots are the means extracted for k=5

[2] https://it.mathworks.com/matlabcentral/fileexchange/13788-optprop-a-color-properties-toolbox

The number of features extracted was *nfeatures*=7, and more specifically they were:

- mean
- standard deviation
- mode
- median
- skewness
- maximum
- minimum

At this point two datasets of (7*5)*12070 were available, that merged together formed a single dataset of 70*12070.

1.5 Feature selection

Only 6 out of the 70 initial rows were selected through the *sequentialfs* function fed with the feature extraction result as input and the differences computed at subsection 1.3 as target.
From this step the following output was obtained: *selectedFeatures*(6*12070).
This step has been repeated for different hidden layer sizes, preparing the selected features for the following step.

1.6 Fitting network

From subsection 1.5 the input for the network has been extracted (i.e. *selectedFeatures*), while from subsection 1.3 the targets have been computed. In MATLAB the following network has been created:

```
    hiddenLayerSize = n;
    net = fitnet(hiddenLayerSize);
    net.divideParam.trainRatio = 70/100;
    net.divideParam.valRatio   = 15/100;
    net.divideParam.testRatio  = 15/100;
    [net,tr] = train(net,x,t);
```

At this point the shallow neural network could be fed up with this two matrices and its performances could be evaluated through metrics such as *Mean Squared Error* and *Regression Coefficient*. In order to evaluate in a rigorous manner the performance metrics, the training has been repeated various times (10), knowing that the randomness between each repetition was a consequence of using MATLAB *dividerand* function. In this step it was of main importance the choice of the *Hidden Layer Size* parameter. From Figure 4 can be noticed how the network performances were highly influenced by this number.

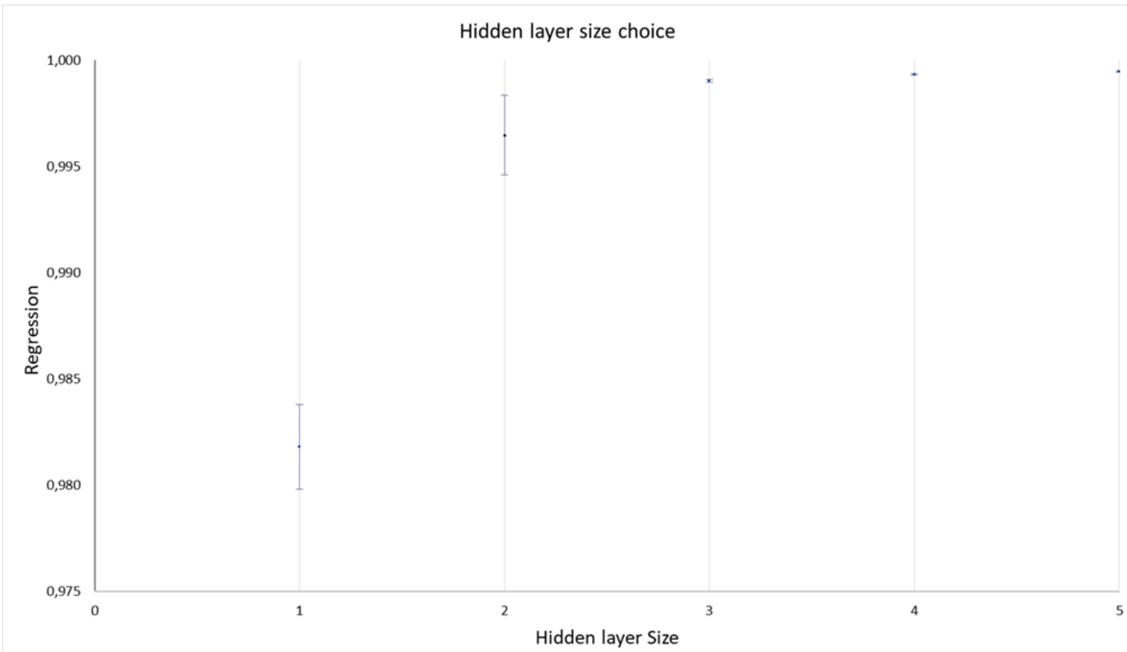

Figure 4: Regression coefficient of the network depending on the hidden layer size, 95% CI

The decision of using a network with *Hidden Layer Size* equal to 3 has been made since it had very good performances but nevertheless had a small size, hence needed a small training time.
The network produced the following results:

$$meansquarederror = 0,00380 \pm 0,00036$$
$$regression = 0,99903 \pm 0,00001$$

The good fitting done by the network is shown also in Figure 5.

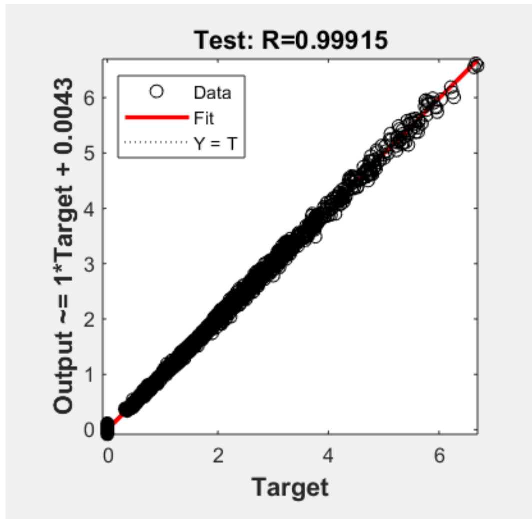

Figure 5: Regression plot

2 Part II

2.1 Problem

The ΔE used in the first part computed through the L*a*b* coordinates is an objective quantity but does not always match the human eye perception (i.e. it has a precision $\approx$ 75% [3]). This is why a correction had to be introduced. The inconsistencies occur only in some parts of the color space that are easier to discover in the L*C*h color space .

2.2 Particular areas

In particular the following areas have been identified.

2.2.1 Dark colors

The ΔE computed over dark colors might bring up high difference values (e.g. 5) but the colors could seem equal to an human observer, hence the adjusted difference should be reduced.

2.2.2 Blue and violet colors

Blue and violet colors present enhanced visual differences w.r.t. to the computed ΔE, for this reason the adjusted difference should be enlarged.

2.2.3 Yellow and saturated colors

Yellow and saturated colors instead tend to present higher ΔE w.r.t. to the perceived one, this is why for this set the adjusted difference should be reduced.

[3] `https://opentextbc.ca/graphicdesign/chapter/4-4-lab-colour-space-and-delta-e-measurements/`

2.3 Fuzzy Inference System

In order to correct the computed ΔE it has been developed a Fuzzy Inference System as shown in Figure 14

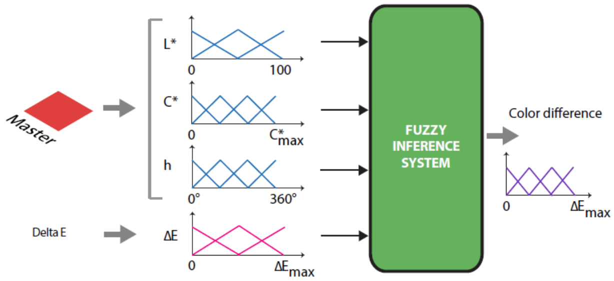

Figure 6: Input/Output scheme of the FIS

2.3.1 Chroma conversion

The input for our system includes the L*C*h coordinates, but the Chroma value range is not constant and depends on L and h values, therefore it has been converted to a percentage value. This has been done through the *maxchroma()* [4] function provided by the *lchconversions* package, that is able to compute the maximum Chroma with a fixed L and h in order to be within the sRGB gamut. Since all the samples in the dataset are within the sRGB gamut this approximation was effective for the project purposes.

```
function [lchConverted] = convertChroma(lch)
lchConverted=lch;
for i=1:length(lch)
  Cmax = maxchroma('lab','lightness',lch(1,i),'hue',lch(3,i));
  lchConverted(2,i) = lch(2,i)*100/Cmax;
end
```

[4]http://myplace.frontier.com/~chooks9592/manual/html/maxchroma.html

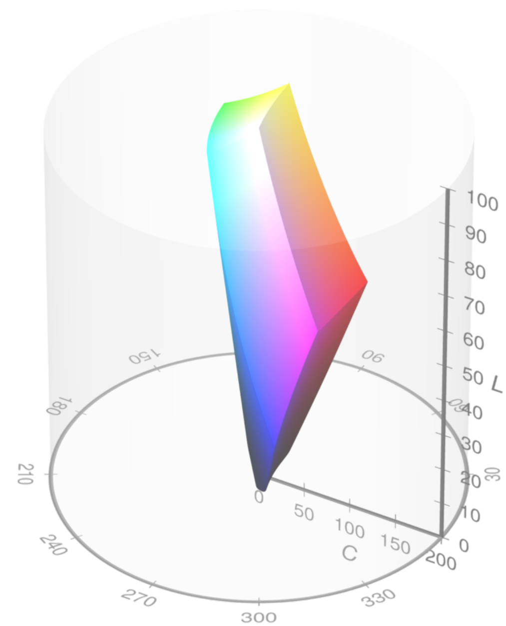

Figure 7: sRGB gamut within the LCH color space

2.3.2 Membership functions

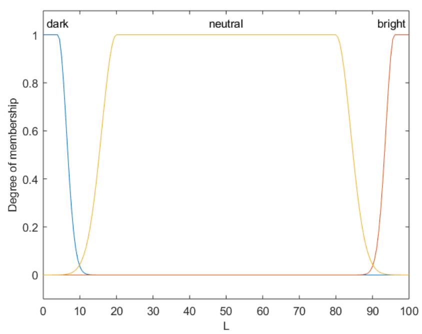

Figure 8: L membership function

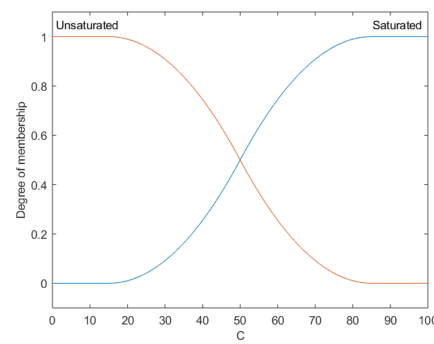

Figure 9: C membership function

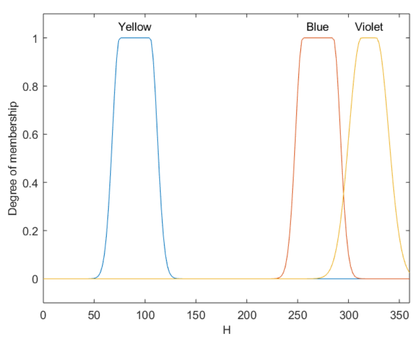

Figure 10: Hue membership function

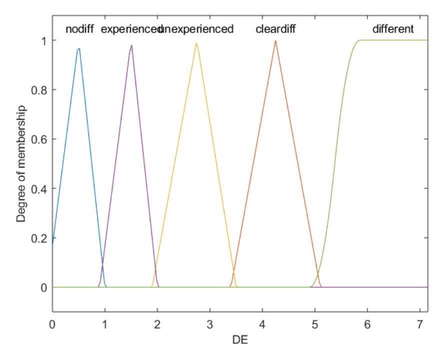

Figure 11: ΔE membership function for input and output

2.3.3 Fuzzy rules

Based on ΔE value, the following classes of differences have been defined:

- $0 < \Delta E < 1$: nodiff
- $1 \leq \Delta E < 2$: experienced
- $2 \leq \Delta E < 3,5$: unexperienced
- $3,5 \leq \Delta E < 5$: cleardiff
- $5 \leq \Delta E$: different

ΔE belonging to the extreme categories such as different and nodiff have been left unchanged since their values are so high or low that a correction would wrongly change their belonging class.

L	**C**	**h**	**ΔE**	**adjustedΔE**
dark	-	-	-	nodiff
not dark	-	blue	experienced	unexperienced
not dark	-	violet	experienced	unexperienced
not dark	not saturated	yellow	unexperienced	experienced
not dark	saturated	-	unexperienced	experienced

Table 1: Main rules applied in the fuzzy inference system

2.3.4 FIS results

The Fuzzy Inference System with the rules and the membership functions specified in subsubsection 2.3.3 and subsubsection 2.3.2 produced as output *adjusted_de* with the aim of matching more precisely the human perception in the areas described in subsection 2.2.

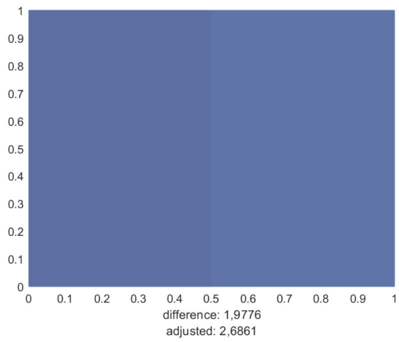

Figure 12: Blue area positive correction

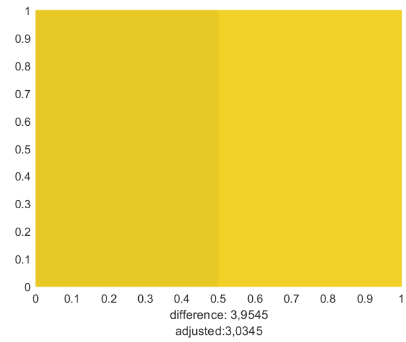

Figure 13: Yellow area negative correction

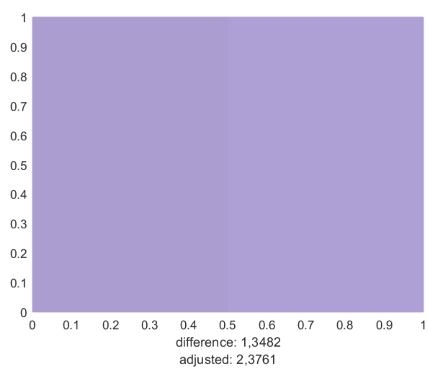

Figure 14: Violet area positive correction

2.4 Fitting network

The same network defined in section 1 has been fed with new targets (i.e. *adjusted_de*). With 10 repetitions the following results have been obtained(95% C.I.):

$$meansquarederror = 0,28949 \pm 0,01389$$

$$regression = 0,93496 \pm 0,00534$$

2.5 Conclusions

The results obtained in subsection 2.4 are quite good, but the highest differences between predicted output and target have been analyzed. It has been noticed that this higher differences are the result of choosing to leave unchanged, while applying fuzzy rules, ΔE values belonging to the extreme classes (i.e. nodiff and different defined in subsubsection 2.3.3). Hence these errors could be caused by the attempt of the network to predict lower values for colors such as yellow or higher for blues, ignoring the fact that extreme categories should not be corrected.