Info Vis

Article

Information Visualization
2018, Vol. 17(2) 161–177

DOI: 10.1177/1473871617700682
journals.sagepub.com/home/ivi
SAGE

Unsupervised color coding for visualizing image classification results

Simone Bianco and Raimondo Schettini

Abstract
In this article, we describe a general purpose system that, given as input a segmented/classified image, automatically provides different visual outputs exploiting solid colors, color boundaries, and transparent colors. Moreover, if the names of the classes are given, the system automatically places a textual label in the less salient sub-region of the corresponding class. For color-class association and class label placement, we take into account the underlying image color and structure exploiting both saliency and superpixel representation. The color selection and the color-class association are formulated both as optimization problems and heuristically solved using a Local Search procedure. Results show the effectiveness of the proposed system on images having different content and different number of annotated regions.

Keywords
Image segmentation, high-contrast colors, color-class association, optimization

Introduction

In image segmentation and classification applications, the user may want to visually code and associate ancillary information, such as names, to the different classes depicted in the image. The decision of how to visually code the different classes and eventually where to place the textual annotations is a problem often referred as view management.[1] Such a problem commonly arises in different domains, such as visual retrieval,[2] video surveillance,[3] medical imaging,[4,5] cartography,[6] remote sensing,[7] and augmented reality.[8] Among visual features, color is widely employed to visualize the output of a segmentation-classification process. While in some applications, the color-class association is strongly driven by conventions (e.g. in cartography), in many other domains this process cannot be easily automated since there are no consolidated conventions or common guidelines. While several heuristic procedures have been proposed to select high-contrast color palettes (see the following section), the problem of color-class association is not completely solved as a serious problem arises because the chosen colors must be displayed together and assigned to classes that maybe spatially adjacent. In practice, this is mainly based on a trial-and-error approach and demands a concerted effort on the part of the user. In many dynamic applications (e.g. augmented reality, video surveillance) or when very large datasets have to be processed,[9–11] this activity should be automated as much as possible.

In this work, we propose a general purpose system that, given as input a segmented/classified image, provides different visual outputs exploiting solid colors, color boundaries and transparent colors. In the first case, each class is filled with the associated color, thus resulting in a pseudo-color representation of the image; in the second case, colored region boundaries are displayed on the original image; in the third case,

Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Milan, Italy

Corresponding author:
Simone Bianco, Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Viale Sarca 336, 20126 Milan, Italy.
Email: simone.bianco@disco.unimib.it

transparent colors are overlaid on the original image. For the second and third outputs, class labels (if available) are automatically placed. In the overlay solutions, to select the colors and to place class names, we take into account the underlying color distribution and structure of the image using both a superpixel representation and saliency map. Both the label placement and the high-contrast color selection problems are NP-hard.[12,13] In this work, we model them as optimization problems and describe a method for heuristically solving them using a Local Search (LS) procedure.

The main contributions of this work are as follows:

- The introduction of a new general purpose system able to provide different visual outputs exploiting solid colors, color boundaries, and transparent colors. The system is completely unsupervised and its core is constituted by algorithms able to work in any color space, with any color distance, and with any color palette.
- Introduction of the use of transparent colors by taking into account underlying image colors, in order to maximize visibility in all the use cases in which the system can be used.
- The formalization of the high-contrast color selection and color-class association as a single problem, and the design of an efficient optimization algorithm to solve it.
- A systematic way to automatically find the best locations where to place class labels.
- The possibility to download the system (http://www.ivl.disco.unimib.it/activities/cc4visual-im-class-results/) and easily expand its functionalities.

Related work

The great majority of the automatic approaches in the state of the art consist of heuristic procedures proposed to define high-contrast sets of colors without considering their association to the image classes (leaving color-class association to the user). For example, Kelly[14] conceived a list of 22 maximally contrasting surface colors, such that each color of the list is maximally different from the one immediately preceding it. Later Carter and Carter[15] formulated the first algorithm to compute easily discriminable sets of colors. Several authors[16–20] have devised algorithms that can also fulfill a number of ergonomical requirements. In 1995, Campadelli et al.[21] presented an abstract formulation of the problem of selecting high-contrast color sets, defining it as a combinatorial optimization problem on graphs. More recently, Carter and Huertas[22] have studied the ability of different metrics and several color spaces to enhance the discriminability of small visual targets with ultra-large color differences. Glasbey et al.[23] proposed a greedy method for the selection of set of colors for categorical image and showed that its performance is comparable with that of a simulated annealing algorithm. Breslow et al.[24] proposed an algorithm for generating color scales for both categorical and ordinal coding; their method uses a positional space partition strategy to generate a lightness scale and then applies the method developed from Campadelli et al.[13] to select discriminable colors according to the lightness constraint. Rodriguez-Pardo and Sharma[25] proposed a dynamical solution to the problem using a hierarchical clustering followed by a simple truncation to select the desired number of high-contrast colors at run time. Radlak and Smolka[26] proposed a methodology for deriving optimized visualization based on maximizing local distance between colors. They present visualization results using a new color contrast measure optimized with a genetic algorithm (GA). Their method can be used to obtain pseudo-color encoded images of segmentation results.

Concerning color-class association, as an alternative to automatic approaches, some interactive systems for color coding have been also proposed.[27–31] They share the idea of using a fixed color palette of high-contrast colors and rely on the user intervention to associate them to the image classes. In this article instead, both the search of a high-contrast set of colors and the color-class association are performed automatically together with the placement of the class label.

The proposed system

The processing pipeline of the proposed system is reported in Figure 1. Given an image to be coded, its classification/segmentation, the set of available colors (in the following, color palettes), and (optionally) the labels of the classes, the system generates different proposals ranging from solid color solutions (see Figure 1(a) and (b)) to color boundary and transparent color overlay solutions (see Figure 1(c) and (d)). For the latter solutions, the optimal locations for the region labels are also produced in output.

Each proposal depends on the input palette, the number, and the spatial relationship of the classes in the image to be coded. To generate such proposals, the problem of selecting high-contrast color sets is jointly addressed with the color-class association problem, in order to maximize the color discriminability among adjacent classes. For color boundary and transparent color overlay solutions, a further problem is faced: the chosen colors have also to be at maximum contrast with respect to the underlying image colors. Finally, to place the class labels, the less salient sub-region able to

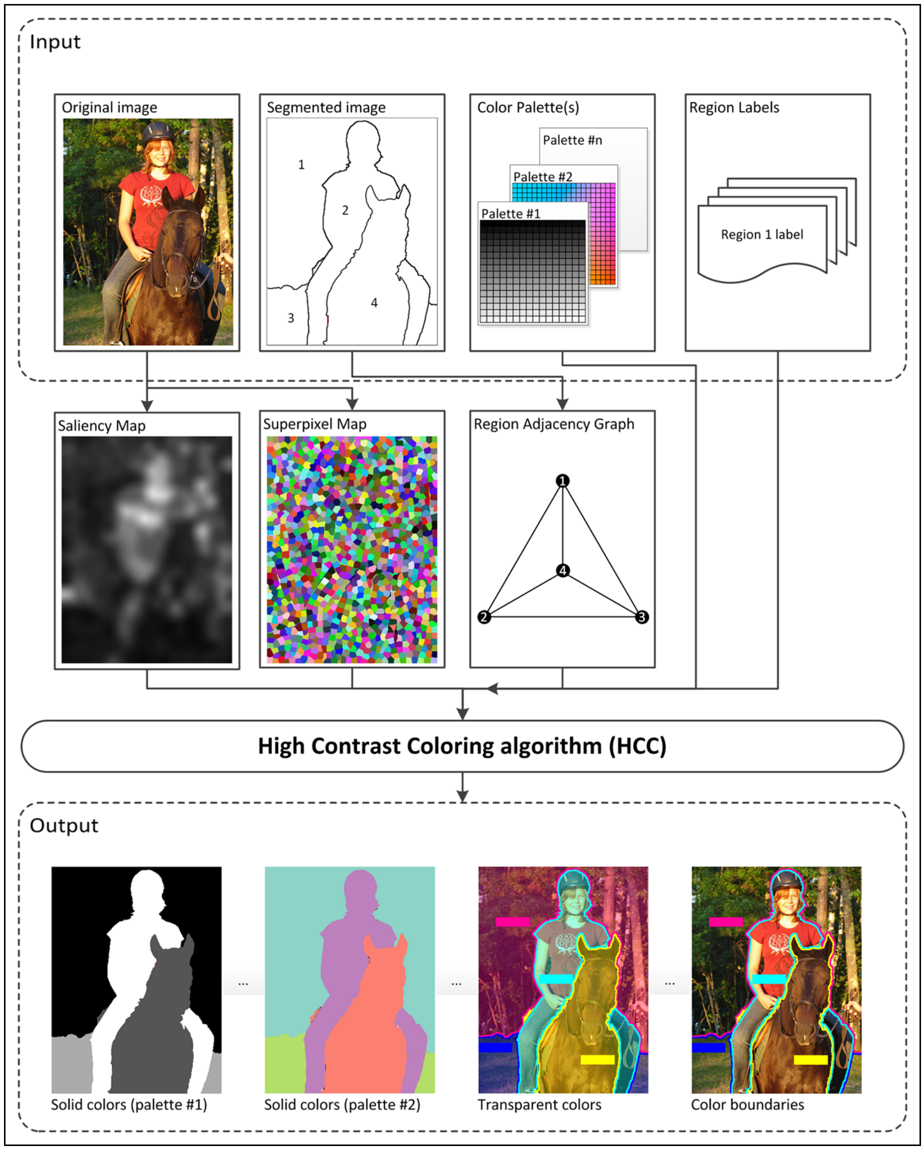

Figure 1. Workflow of the proposed system.

contain the entire label must be found for each class. To take into account the content of the image to be coded and the spatial relationship of the classes, suitable image descriptions in terms of superpixel representation, saliency, and class adjacency representation are exploited.

The proposed system takes as input an RGB image I, its segmented/classified version R with class regions $r_1, \ldots, r_n$, the color palette C, and the class labels l_i, $i = 1, \ldots, n$. The first stage consists in the computation of the superpixel map S,[32,33] which is composed by superpixels $s_1, \ldots, s_m$, and the computation of the saliency map M. The superpixel representation is used since the number of pixels is high even at moderate resolutions, making the optimization on the level of pixels intractable.[33] Superpixels instead are local, coherent, and preserve most of the structure. In this work, we followed Mori et al.[33] which applied Normalized Cuts algorithm[34] to produce the superpixel map. An example of superpixel map is shown in Figure 2(d). Saliency intuitively characterizes some parts of the image that appear to an observer to stand out relative to their neighboring parts.[35] In this work, we use saliency to detect which are, for each region, the locations of the least prominent parts. These are the candidate locations to place the class label. To compute the saliency map in this work we use the method proposed in Zhang and Sclaroff.[36] It is a Boolean map–based saliency model which leverages global topological cues that are known to help in

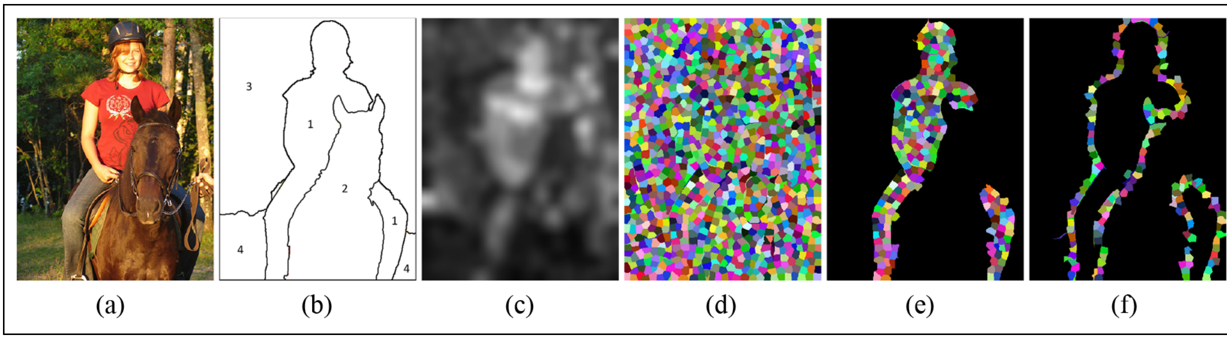

Figure 2. (a) Original image, (b) image segments, (c) saliency map, (d) superpixel map, (e) superpixels belonging to I_1, that is, the inside of region r_1, and (f) superpixels belonging to O_1, that is, the outside of the same region.

perceptual figure-ground segregation. The model is based on the surroundedness cue taken from Gestalt psychological studies. An example of saliency map is shown in Figure 2(c).

Both S and M are computed from I. The segmented image R is used to compute the region adjacency graph (RAG). The adjacency matrix A is obtained from the RAG: the element $A_{(i,k)}$ is set to 1 if nodes i and j are connected in the RAG (i.e. the regions r_i and r_j are adjacent in R) and set to ∞ otherwise.

The looping on the regions then begins. For each region $i = 1, \ldots, n$, all the superpixels $s_j \cap r_i$ are found, and the sets I_i, O_i are initialized to the empty set, that is, $I_i = O_i = \emptyset$. For each superpixel s_j, a k-means clustering on the RGB values of the pixels identified by positions $\mathbf{x} \in I \cap s_j$ is run with $k = 2$ to find the two color centroids c_1 and c_2. If the distance between colors c_1 and c_2 is less than a fixed threshold t, then their average is added to the set of colors I_i, otherwise both c_1 and c_2 are added

$$I_i = \begin{cases} I_i = I_i \cup \text{mean}(c_1, c_2) & \text{if } d(c_1, c_2) < t \\ I_i = I_i \cup c_1 \cup c_2 & \text{otherwise} \end{cases} \tag{1}$$

The pixels of the saliency map M corresponding to the current region r_i are then filtered using a box filter b_i having dimensions $[w_i, h_i]$, which are defined by the size of the label l_i assigned to the corresponding class. In order to avoid as much as possible to the label to stick out from the current region, a modified version of M is computed

$$M'(\mathbf{x}) = \begin{cases} M'(\mathbf{x}) & \text{if } \mathbf{x} \in r_i \\ P & \text{otherwise} \end{cases} \tag{2}$$

If $i > 1$, M' is updated by setting to P the label locations for the regions $r_j, j < i$ in order to avoid overlapping labels. The new map M' is used to produce the output $B_i = b_i * M'$. The location L_i where to place the label for region r_i is then found as

$$L_i = \arg\min_{\mathbf{x}} B_i(\mathbf{x}) \tag{3}$$

By design, only one label for each class is placed. In case of classes composed of multiple regions, the label is placed only in the largest region in the less salient part.

The next step is the identification of the part of the image adjacent to r_i, that is, the set of pixels neighboring with r_i. To this end, the dilation mathematical morphology operator is applied to r_i to obtain dr_i. The region nr_i, representing the adjacent space external to region r_i is then obtained as $nr_i = dr_i - r_i$. The superpixels $s_j \in nr_i$ are then found, and for each superpixel, the two centroids c_1 and c_2 are found. As for I_i, O_i is updated as follows

$$O_i = \begin{cases} O_i = O_i \cup \text{mean}(c_1, c_2) & \text{if } d(c_1, c_2) < t \\ O_i = O_i \cup c_1 \cup c_2 & \text{otherwise} \end{cases} \tag{4}$$

The final stage of the proposed system is to find a set of colors c_i, $i = 1, \ldots, n$ (one for each image class), $\{c_i\}_{i=1}^{N} \in C$ having maximum contrast between them, and with the colors I_i, O_i, $i = 1, \ldots, n$. This is done using the high-contrast coloring (HCC) algorithm described in the next subsection.

The pseudo-code of the algorithm implemented in the proposed system is given in algorithm 1.

HCC algorithm

Given a palette C of possible colors and the sets of colors, respectively, inside and outside each region r_i, that is, I_i and O_i, $i = 1, \ldots, n$, the problem is to find a set of n colors having maximum contrast among them and

Algorithm 1 Pseudo-code of the algorithm implemented in the proposed system.

```
Data: RGB image I, segmented image
      R = ⋃_{i=1}^N r_i, color palette(s) P, region
      labels {l_i}_{i=1}^N
Result: colors C_i, label locations L_i,
      i = 1, ..., N
Compute saliency map M
Compute superpixel segmentation S
Initialize adjacency matrix A = Inf(N, N)
for each region r_i ∈ R, i = 1 to N do
    I_i = O_i = ∅
    M' = M
    Find superpixels s_k ∈ r_i
    for each superpixel s do
        Set M'(x) = P for each x ∈ s
        Run clustering on RGB colors to find 2
         centroids c_1 and c_2
        if d(c_1, c_2) < t then
            I_i = I_i ∪ mean(c_1, c_2)
        else
            I_i = I_i ∪ c_1 ∪ c_2
    if i > 1 then
        for j = 1, ..., i − 1 do
            Set M'(x) = P for each
             x ∈ [L_j, L_j + w] × [L_j, L_j + h]
    dr_i = dilate(r_i)
    Find superpixels s_k ∈ nr_i = dr_i − r_i
    for each superpixel s do
        Run clustering on RGB colors to find 2
         centroids c_1 and c_2
        if d(c_1, c_2) < t then
            O_i = O_i ∪ mean(c_1, c_2)
        else
            O_i = O_i ∪ c_1 ∪ c_2
    Find regions r_j, j = {1, ..., N} − i s.t.
    r ∩ dr_i ≠ ∅
    if ¬ isempty(j) then
        A_(i,j) = 1
    B = boxfilter(M', h, w)
    Find L_i s.t. L_i = arg min_{x∈r_i} B
Find N high contrast colors wrt
A,I_i,O_i,i = 1, ..., N that maximize Eq. 5
```

with respect to I_i and O_i and to place them so that the local contrast is maximized. Formally

$$c_1,\ldots,c_n = \arg\max_{c_1^*,\ldots,c_n^*} \min\left(\left[\min_{\substack{j=1,\ldots,n\\k=1,\ldots,n\\j\neq k}} d(c_j^*, c_k^*),\ \min_{\substack{j=1,\ldots,n\\k=1,\ldots,n\\j\neq k}} A_{(j,k)} d(c_j^*, c_k^*),\ \min_{\substack{i=1,\ldots,n\\j=1,\ldots,|I_i|}} d(c_i^*, I_i\{j\}),\ \min_{\substack{i=1,\ldots,n\\j=1,\ldots,|O_i|}} d(c_i^*, O_i\{j\})\right] \circ [w_d, w_A, w_I, w_O]^{-1}\right) \tag{5}$$

where the operator $\{\cdot\}$ is used to index elements inside a set, $|\cdot|$ is the set cardinality, and $^\circ$ is used to denote the Hadamard product. The first term in the right-hand side of equation (5) computes the minimum distance among the selected colors, the second one the minimum distance between the colors corresponding to adjacent regions, the third one the minimum distance between the color associated with the region r_i and the colors inside it (i.e. I_i), and the fourth one the minimum distance between the color associated with the region r_i and the colors outside it (i.e. O_i).

Some of the possible outputs are reported in Figure 1. They are obtained by acting on the weights $[w_d, w_A, w_I, w_O]$, as reported in Table 1.

The optimization problem is solved with the LS procedure introduced in Bianco and Citrolo.[37] The LS procedure is based on the concept of neighborhood, which is defined as follows

$$N_h(K) = \{\tilde{K} \subset C : |K\backslash\tilde{K}| = h, h < n, |\tilde{K}| = n\} \tag{6}$$

In this work the 1-neighborhood $N_1(\cdot)$ is used, which means that two solutions are 1-neighbors if and only if they differ for just one element. Starting from a random solution with n colors, in each iteration, an exhaustive search over the 1-neighborhood of the current solution is performed: the solution that leads to the highest improvement according to equation (5) is selected as the new current solution. The optimization ends when no improved solution is found. In order to speed-up convergence, the solution is initialized using grid search. Starting from the image region r_1, c_1 is chosen to solve equation (5) by setting $[w_d, w_A, w_I, w_O] = [0, 0, w_I, w_O]$ and $n = 1$. For image regions from r_2 to r_n, new elements are added to the solution by solving equation (5) and keeping fixed the previous elements.

Experimental results

The proposed method can work with any color palette having any cardinality, described in any color space given that a color distance is defined. The default color palette is the Munsell Atlas which contains over 1300 colors. This choice is due to the fact that the classified images may not only be displayed on the screen but also printed. Since the gamuts of feasible colors of common displays and color printers differ considerably, a high-contrast color coding on the screen may be drastically modified when printed. Since many colors cannot be reproduced in print, restricting the selection of color to the Munsell Atlas greatly limits this drawback.[30]

Three other different color palettes C among which to select the n colors are used in the following. They have been chosen as can be easily visualized and appreciated as a whole. The three color palettes are reported

Table 1. Output obtained with different parameter settings in equation (5).

Parameters	Output obtained
$[w_d, w_A, w_I, w_O]$	Solution taking into account the original image colors inside each region. Suggested when region boundaries have to be displayed or colors have to be overlaid on the original image.
$[w_d, w_A, 0, 0]$	Solution ignoring the original image colors. Suggested when a pseudo-color representation of the image is needed.
$[w_d, 0, 0, 0]$	Solution ignoring the original image colors and region adjacency. Suggested when only a set of maximally contrasting colors is needed (e.g. Bianco and Citrolo[37]).

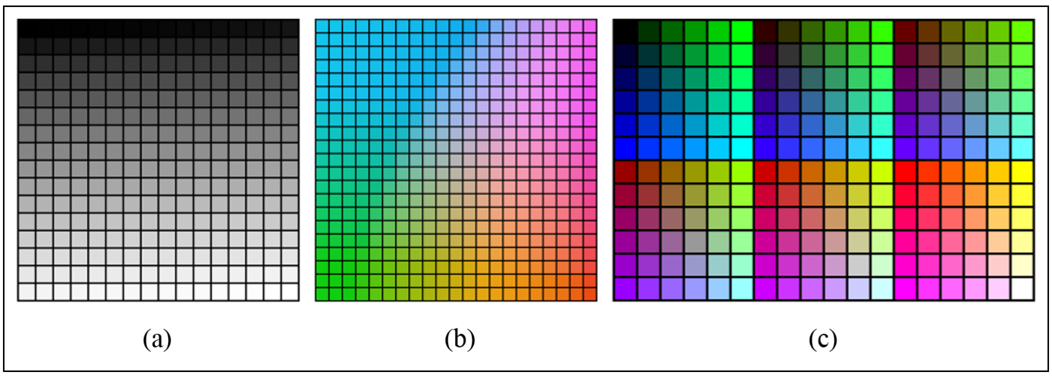

Figure 3. Color palettes used in this work: (a) gray-scale, (b) iso-luminance, and (c) web-safe.

in Figure 3. These palettes have different color characteristics: the first one varies only in luminance, and thus it is inherently one-dimensional (1D); the second one is two-dimensional (2D) as representing a sampling of an iso-luminance plane in the CIELab color space;[38] and the third one is three-dimensional (3D) as spanning the whole RGB cube. The first palette is a gray-scale composed of 256 shades of gray, that is, $|C| = 256$. The second one contains iso-luminance colors, with $|C| = 441$. The third one corresponds to the palette of web-safe colors, with $|C| = 216$. The web-safe palette was defined as a set of colors that could be shown without dithering on 256-color displays. In fact, on 256-color displays, applications can set a palette of any selection of colors that they choose, dithering the rest.

We report in Figure 4 the colors selected for the three palettes for different set cardinalities $n = 3, 4, 6, 9, 12$ without considering the spatial layout, that is, setting $[w_d, w_A, w_I, w_O] = [1, 0, 0, 0]$ in equation (5). It is possible to see that when n increases, the selected colors tend to be more similar. This explains the importance of the spatial layout of the colors on the image, that is, the way in which the colors are associated with the image regions, which is captured by the second term in equation (5).

Some example results taking into account the image spatial layout and color-class associations are reported for different application domains: Pascal VOC 2012 segmentation challenge,[39] texture classification, images annotation, and webpage layout segmentation. For all examples, the distance $d(\cdot, \cdot)$ used is the CIE ΔE which is an Euclidean distance computed in the CIELab color space. The CIELab color space is chosen in this work for its popularity and simplicity,[40] but other color spaces could be used in the proposed system as well, as for example the CIECAM02[41] and its extensions[42] which are particularly indicated when the final viewing conditions are known.

Concerning the superpixel segmentation, the threshold t in equations (1) and (4), that is, the one to decide if two colors are different enough to be both added to the respective sets I_i and O_i or they have to be averaged into a single color, is set to 3 ΔE units. Below this value, humans hardly perceive color differences. Concerning class labels, in all the examples equal-length labels are assumed. The size for the box filer b_i and thus of the space reserved to the placement of the label of each region has been set to $[w, h] = [75, 21]$.

In Figure 5 for each example, we report the original image, its segmentation with number overlaid representing the class id, and different proposals: the results using the gray-scale and iso-luminance palettes (with $[w_d, w_A, w_I, w_O] = [w_d, w_A, 0, 0]$ and no labels overlaid), the results using the web-safe palette (with $[w_d, w_A, w_I, w_O] = [w_d, w_A, w_I, w_O]$ and labels overlaid) coloring boundaries only, or overlaying region colors

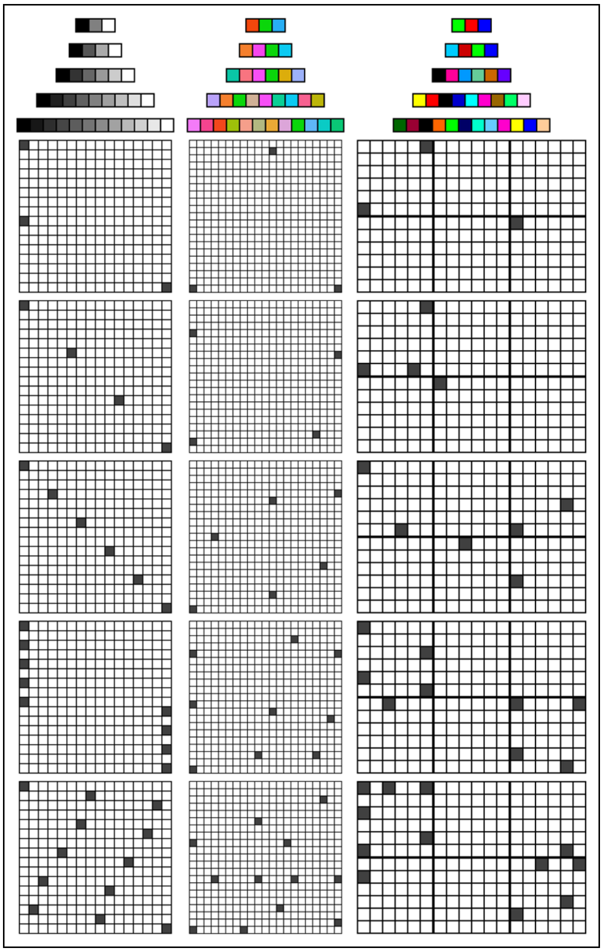

Figure 4. Colors selected for the three palettes for different set cardinalities $n = 3, 4, 6, 9, 12$ without considering the spatial layout (top row) and their position inside the respective palettes (remaining rows). The sRGB coordinates of the selected colors are reported in Appendix 1.

on the original image. The RAG of each image is reported in Figure 6. Notice how for classes composed of multiple regions (i.e. class 4 in the first image, class 2 in the second one, and class 5 in the third one) the labels are placed on the region that can contain the whole label. Together with the different proposals, the user can also set the transparency level of the overlay solutions as well as the saturation of the underlying image. Some examples of the effect of these two parameters are shown in Figure 11 in Appendix 1.

In Figure 7, we report the chromaticity plane of the CIELab color space showing the position of the chosen colors with respect to image colors for the first image in Figure 5 using the web-safe palette. The image colors I_i and O_i for each class r_i are represented as blue dots. The colors themselves are represented as red crosses. The gray line represents the boundary of the CIELab color space and is a superset of the palette. From the figure, it is possible to see how the four selected colors are both very far from each other and far from the colors of the corresponding underlying classes.

Finally, we numerically compare the results obtained by the proposed algorithm with those obtained by two alternative solvers to the LS proposed in this article: the first one is a color selection and placement solution obtained by multiple random selections; the second one is based on GAs that are often used to solve high-contrast color selection problems.[26] Concerning random selection, we performed 1,000,000 random independent runs and report the results of both the best solution found and the average solution. Concerning GA, we performed 100 independent runs with 200 individuals each and report the results of both the best solution found and the average solution. The results are reported both in terms of fitness and in terms of the four elements of which it is composed of: the distance between colors (d_D), the distance among adjacent colors (d_A), and the distance among the color associated with each class and the colors, respectively, inside and outside of it (d_I and d_O). Results are visualized as radar plots in Figure 8. Note that $d_D \leqslant d_A$, and the equality holds only when the nearest colors are assigned to adjacent regions. From the plots, it is possible to see that the proposed method is able to outperform the compared solvers on all the sample images and on all the different terms composing the fitness function. The second best solution is always obtained by the best GA solution, with the average random solution obtaining the worst result. Numerical values for the fitness values are also reported in Table 2 together with the improvement of our solutions with respect to the best compared solver, that is, the best GA solution. In particular, we can observe that when the number of classes is small (i.e. 4 to 5) the average improvement of the proposed method with respect to the best GA solution is 2.43%; when the number of classes increases, instead, the average improvement increases to 26.06%. Computational times are reported in Table 3. For the best random solution, the total time for the 1,000,000 guesses is reported, while for the average random one, the time for a single guess is reported. For the best GA solution, the total time for the 100 runs is reported, while for the average GA one, the time for a single run is reported. For the proposed method, being it not based on randomness, a single value is reported.

To test the robustness of the proposed method to the initialization, 100 independent runs are performed with random initialization. The radar plots of the solution obtained with the proposed initialization and the average solution obtained with random initialization

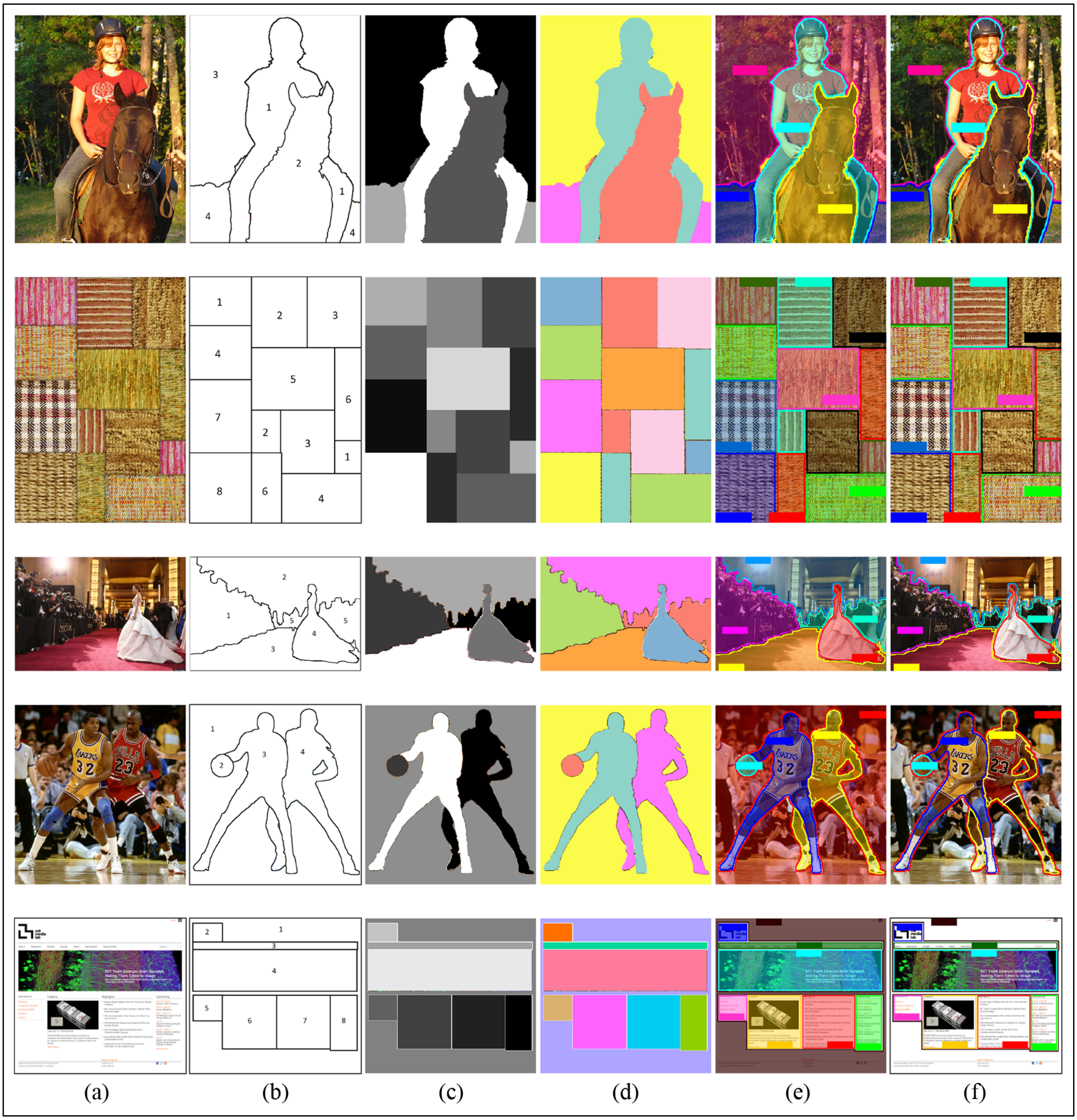

Figure 5. Examples of results obtained by our method: (a) original image, (b) its segmentation with overlaid region identification numbers, pseudo-color results using the (c) gray-scale palette and (d) iso-luminance palettes, and region boundaries (e) with and (f) without overlaid colors using the web-safe palette (both with labels overlaid).

are reported in Figure 9. The average drop in performance using a random initialization is less than 6.5%, which is one order of magnitude lower than the one that we have passing from the best random solution to the average random one (i.e. 66.8%), and almost half of the one that we have passing from the best GA solution to the average GA one (i.e. 10.0%). Furthermore, even with random initialization, the results obtained by the proposed method are on average almost 4.4% better than those obtained by the best GA solution, confirming the robustness of the proposed method to the choice of the initial conditions.

As a final experiment, we run the same experiments as before changing the working color space used from CIELab to CIECAM02-UCS.[42] The CIECAM02-UCS is an extension of CIECAM02 that better fits both small and large color differences. As mentioned at the beginning of this section, to be properly used,

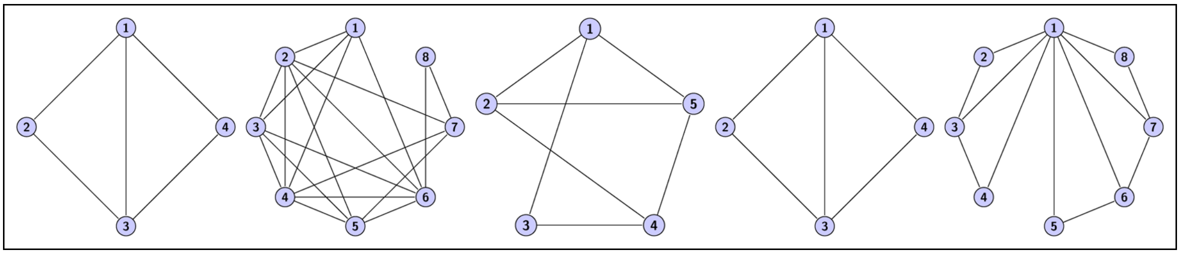

Figure 6. RAGs of the images reported in Figure 5.

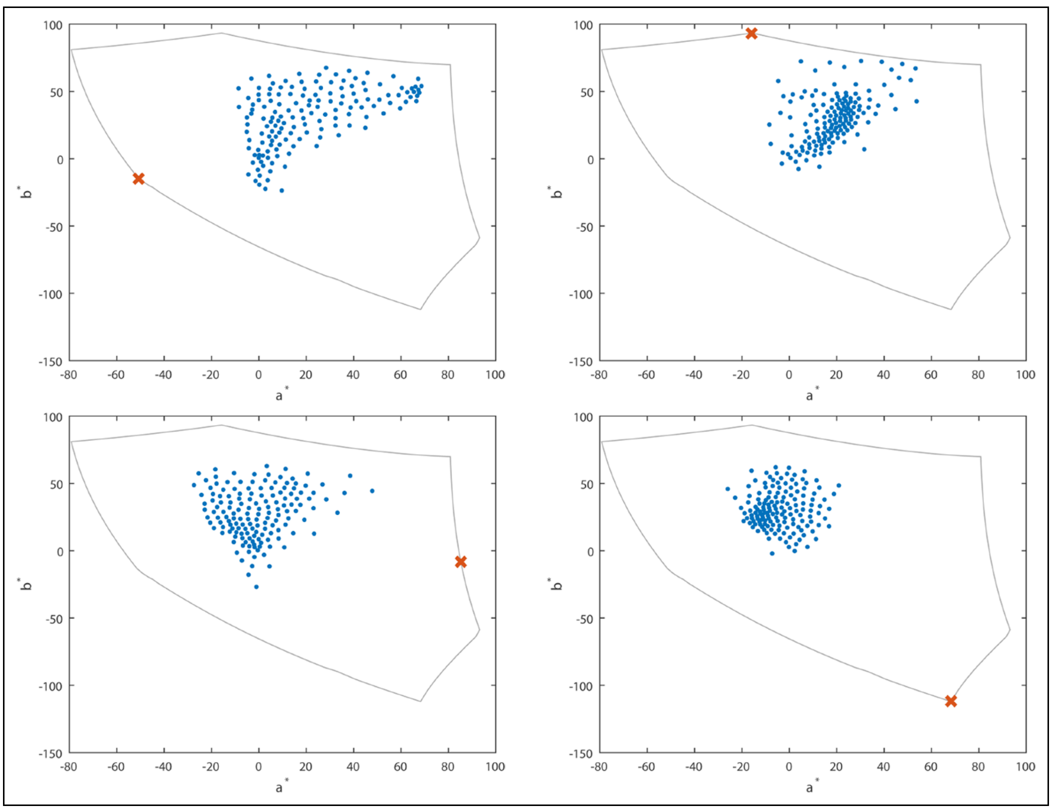

Figure 7. Chromaticity plane of the CIELab color space showing the position of the chosen colors with respect to image colors for the first image in Figure 5 using the web-safe palette. Image colors I_i and O_i for each class r_i are represented as blue dots. The colors themselves are represented by red crosses. The gray line represents the boundary of the CIELab color space.

Table 2. Comparison of fitness values obtained in terms of CIELab ΔE by the random selection, the GA selection, and the proposed method on the five images in Figure 5.

Image	No. of classes	No. of regions	Random (best)	GA (best)	Random (average)	GA (average)	Proposed	Improvement
Image 1 (horse)	4	7	60.95	70.10	17.76	61.68	70.10	0.00%
Image 2 (texture)	8	13	45.15	47.45	17.78	43.75	60.63	27.78%
Image 3 (Oscar)	5	6	56.44	59.76	18.74	54.74	63.50	6.26%
Image 4 (basketball)	4	5	61.90	69.02	18.34	62.12	69.74	1.03%
Image 5 (webpage)	8	8	42.93	50.13	14.89	44.15	62.33	24.34%

GA: genetic algorithm.For the random selection, the higher and the average fitness over 1,000,000 independent runs are reported. For the GA selection, the higher and the average fitness over 100 independent runs with 200 individuals each are reported.

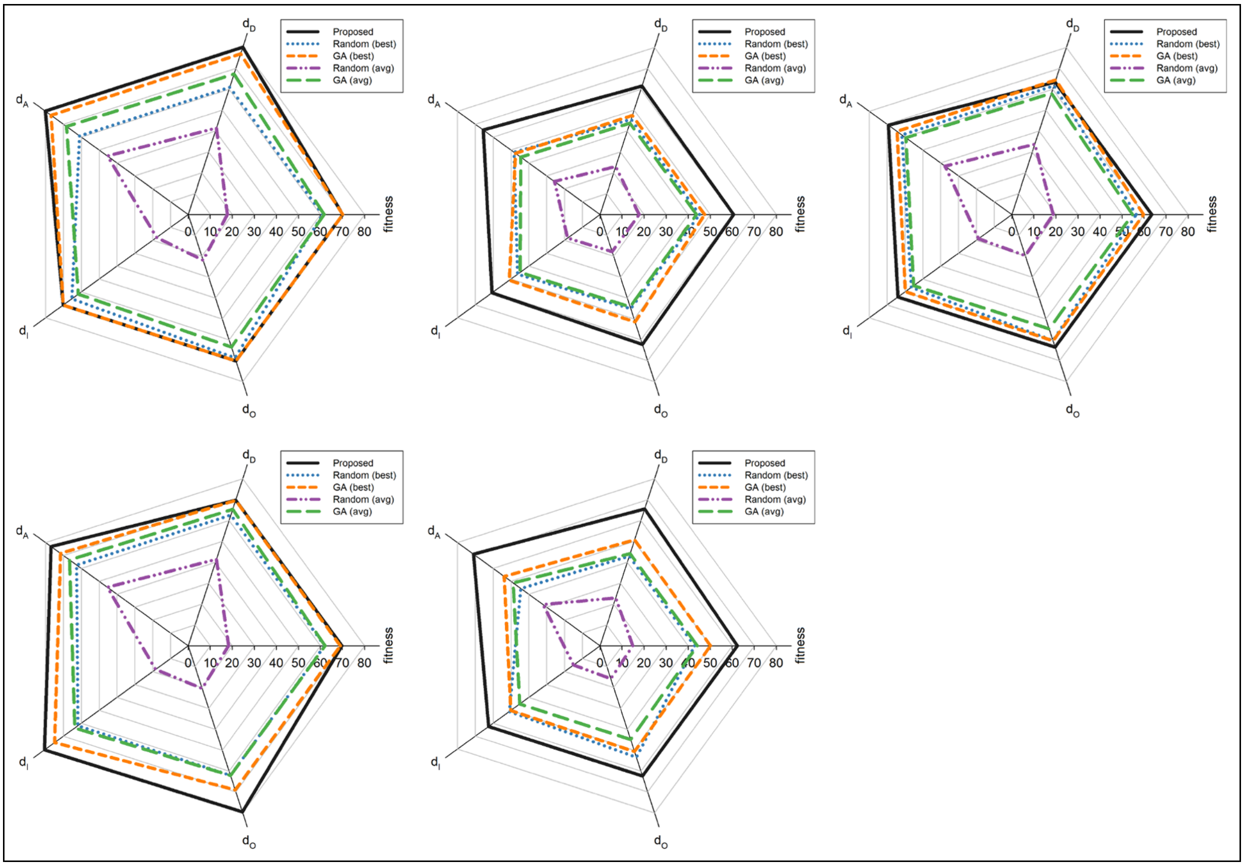

Figure 8. Radar plot visualizations of the fitness values and the four terms of which it is composed, obtained by the random selection (red) and the proposed method (blue) on the five images in Figure 5. For the random selection, the higher fitness over 1000 independent runs is reported.

Table 3. Comparison of execution times for the random selection, the GA selection, and the proposed method on the five images in Figure 5.

Image	No. of classes	No. of regions	Random (best)	GA (best)	Random (average)	GA (average)	Proposed
Image 1 (horse)	4	7	1305.8 s	1507.6 s	0.0013 s	15.08 s	3.03 s
Image 2 (texture)	8	13	2073.4 s	2947.8 s	0.0021 s	29.48 s	18.87 s
Image 3 (Oscar)	5	6	1529.4 s	1793.4 s	0.0015 s	17.93 s	5.56 s
Image 4 (basketball)	4	5	1349.2 s	1481.4 s	0.0013 s	14.81 s	3.22 s
Image 5 (webpage)	8	8	2048.5 s	3036.4 s	0.0020 s	30.36 s	18.99 s

GA: genetic algorithm.For the random selection, the average and total time for 1,000,000 independent runs are reported. For the GA selection, the average and total time for the 100 independent runs with 200 individuals each are reported.

CIECAM02 and its extensions require the final viewing conditions to be known and specified. We run the experiments considering two different viewing conditions: self-luminous display with CIE D65 white point under office illumination and print evaluation in a light booth under CIE D50 standard illuminant. The relevant parameters of both the viewing conditions are reported in Table 4. The distance $d(\cdot,\cdot)$ used is the $\Delta E'$, and the threshold t is set to 3 units. The fitness values are reported in Table 5, where it can be seen that algorithms' ranking are the same as those in Table 2 and the improvements are similar. Additionally, we replicate the results reported in Figure 4 by considering the CIECAM02-UCS color space and the viewing condition 1 (see Table 4). The results are reported in Figure 10 and represent the colors selected for the three palettes for different set cardinalities $n = 3, 4, 6, 9, 12$ without considering the spatial

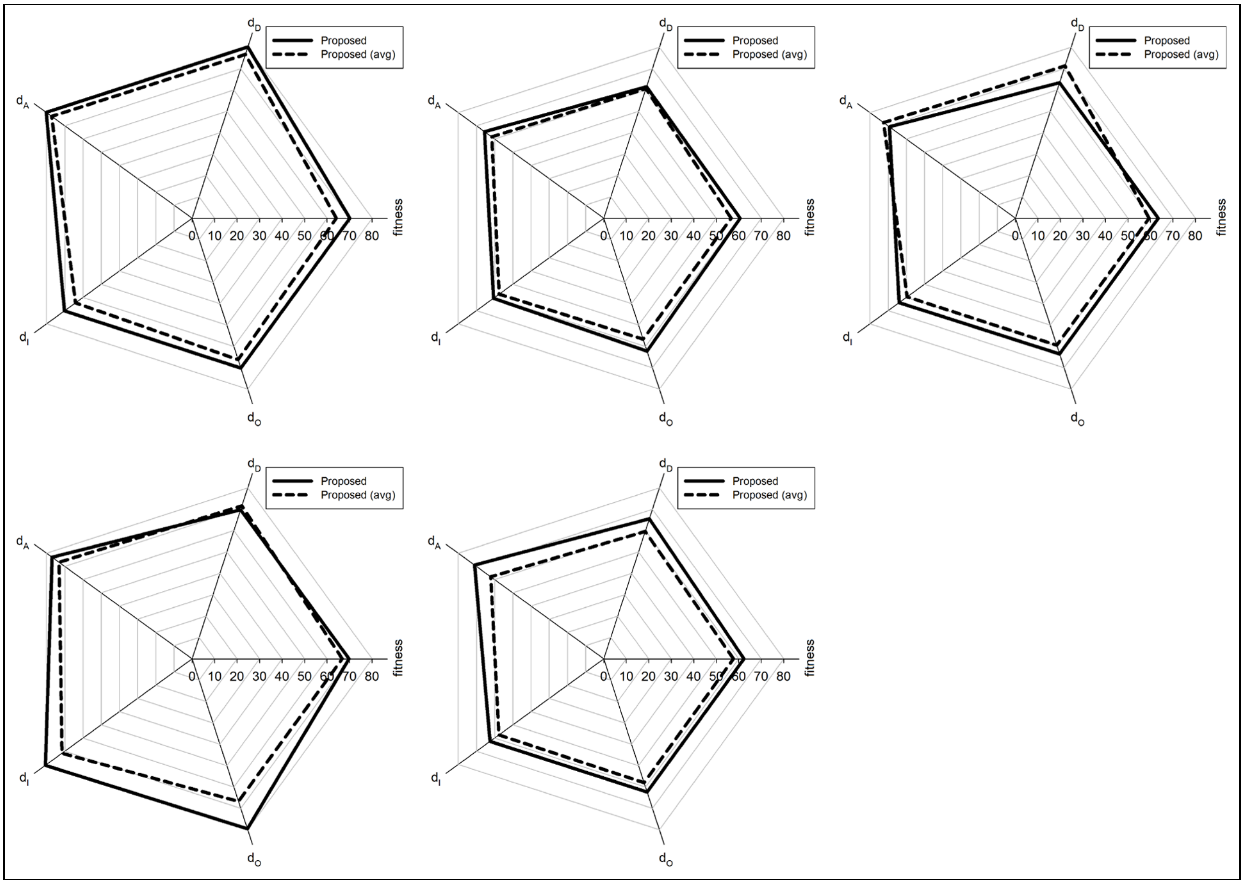

Figure 9. Radar plot visualizations of the fitness values and the four terms of which it is composed, obtained of the proposed method with the proposed initialization (solid line) and the average solution obtained with random initialization (dashed line) on the five images in Figure 5. For the random initialization, the average values over 100 independent runs are reported.

Table 4. Relevant parameters for the two viewing conditions considered in Table 5.

Parameters	Viewing conditions	
	1	2
Ambient illumination in lux (or cd/m^2)	500 (159.2)	1000 (318.3)
Scene or device white luminance	80	318.30
L_A in cd/m^2	15	60
Adopted white point	Display	Light booth
S_R	2	1
Surround	Average	Average

layout, that is, setting $[w_d, w_A, w_I, w_O] = [1, 0, 0, 0]$. The sRGB coordinates of the selected colors are also reported in Appendix 1 to allow their further visual appreciation and their immediate re-use.

Conclusion

Color is pre-attentively observed and used to segment the visual environment. This characteristic makes it particularly effective in visualizing image classification and segmentation results. A problem that may arise is that we often have more items to be represented in an image than easily discriminable colors. Moreover, selecting a high-contrast color subset is only part of the solution since the colors must be assigned to the image classes and displayed together. In this article, we proposed a general purpose system that given as input a segmented/classified image automatically provides

Table 5. Comparison of fitness values obtained in terms of CIECAM02-UCS $\Delta E'$ by the random selection, the GA selection, and the proposed method on the five images in Figure 5.

Image	Viewing condition 1				Viewing condition 2			
	Random (best)	GA (best)	Proposed	Improvement	Random (best)	GA (best)	Proposed	Improvement
Image 1 (horse)	72.64	72.09	74.76	2.92%	74.46	75.61	75.61	0.00%
Image 2 (texture)	36.22	41.02	50.23	22.45%	37.63	43.18	51.75	19.85%
Image 3 (Oscar)	55.25	60.41 63.15	4.54%	58.22	63.02	66.33	5.25%	
Image 4 (basketball)	73.32	71.44	74.05	1.00%	74.96	74.52	75.12	0.21%
Image 5 (webpage)	34.98	43.43	52.87	21.74%	35.07	45.98	53.82 17.95%	

GA: genetic algorithm.For the random selection, the higher fitness over 1,000,000 independent runs is reported. For the GA selection, the higher fitness over 100 independent runs with 200 individuals each is reported.

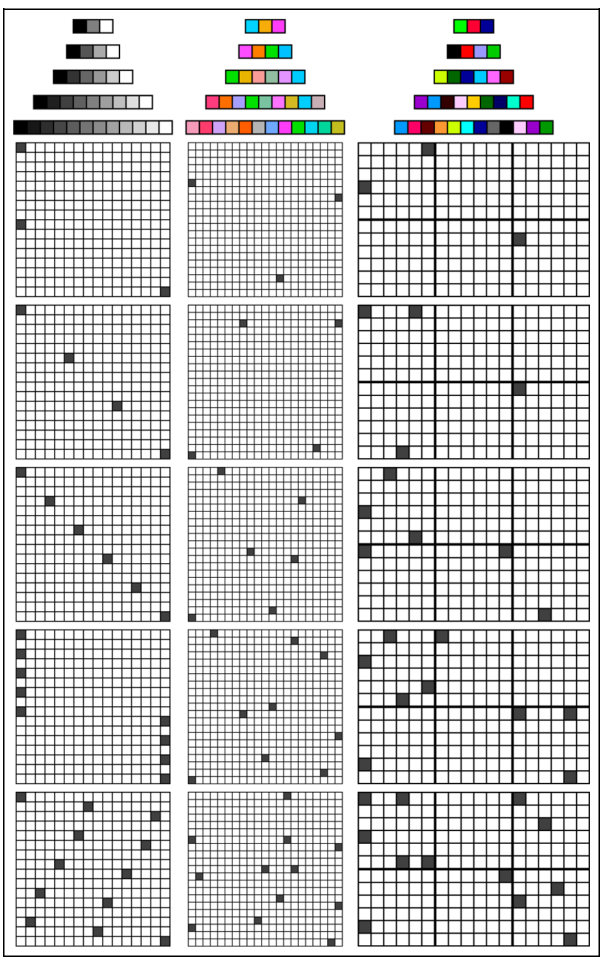

Figure 10. Colors selected for the three palettes for different set cardinalities $n = 3, 4, 6, 9, 12$ without considering the spatial layout (top row) and their position inside the respective palettes (remaining rows). The color space considered is the CIECAM02-UCS under the viewing condition 1 (see Table 1). The sRGB coordinates of the selected colors are reported in Appendix 1.

different types of outputs ranging from solutions to be used when a pseudo-color representation of the image is needed, to solutions to be used when region boundaries have to be displayed or transparent colors have to be overlaid on the original image, specifically: different pseudo-color representations (obtained using different color palettes), colored region boundaries displayed on the original image, transparent colors overlaid on the original image, and if available, the automatic placement of the class labels. Both the color selection and class-color association are formulated as an optimization problem and heuristically solved using a LS procedure. Experimental results show the effectiveness of the proposed method on images having different content and different number of annotated classes. Results are compared with the best solution obtained in 1000 independent random selections, showing an average improvement of 51.6% in the fitness values. Finally, we also included a method for the automatic placement of class labels on the image exploiting an image description in terms of saliency.

The proposed system is available at http://www.ivl.disco.unimib.it/activities/cc4visual-im-class-results/ and could be used in existing image and video annotation software such as LabelMe,[9] Ilastik,[43] Sloth,[44] VATIC,[10] and iVat.[11]

As future work we plan to investigate the use of a combination of different image descriptions to better identify the less prominent sub-regions of the image. We also plan to conduct a user study to investigate where users prefer to place class labels and integrate these preferences directly in the fitness function.[45] The whole system could be validated in specific use cases, for example, remote sensing where the color-class association should take into account the semantic and therefore the colors could be a priori fixed or limited to a certain portion of the gamut (e.g. we could be interested to use a particular green for fields or limit the associated color to the gamut of greens). In the actual system, it is already possible to fix one or more colors, but this functionality should be extended. The system could be further extended to include also options for visually impaired and color-blind users. We also plan to conduct a user study to evaluate the proposed system and to integrate more complex relationships among the chosen colors such as color harmony.[46]

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Bell B, Feiner S and Höllerer T. View management for virtual and augmented reality. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*, Orlando, FL, 11–14 November 2001, pp. 101–110. New York: ACM.
2. Wang C, Reese JP, Zhang H, et al. Similarity-based visualization of large image collections. *Inform Visual* 2013; 14: 183–203.
3. Kim SY, Maciejewski R, Ostmo K, et al. Mobile analytics for emergency response and training. *Inform Visual* 2008; 7(1): 77–88.
4. Fuchs H, Livingston MA, Raskar R, et al. Augmented reality visualization for laparoscopic surgery. In: *Proceedings of the international conference on medical image computing and computer-assisted intervention*, Cambridge, MA, 11–13 October 1998, pp. 934–943. Berlin, Heidelberg: *Springer*.
5. Ward RC, Pouchard LC, Beckerman BG, et al. The *HotBox*: a visual user interface to medical data. *Inform Visual* 2006; 5(4): 290–295.
6. Hilton BN, Horan TA, Burkhard R, et al. SafeRoadMaps: communication of location and density of traffic fatalities through spatial visualization and heat map analysis. *Inform Visual* 2011; 10(1): 82–96.
7. Tateosian L, Mitasova H, Thakur S, et al. Visualizations of coastal terrain time series. *Inform Visual* 2013; 13: 266–282.
8. Tonnis M, Sandor C, Lange C, et al. Experimental evaluation of an augmented reality visualization for directing a car driver's attention. In: *Proceedings of the 4th IEEE/ACM international symposium on mixed and augmented reality*, Vienna, 5–8 October 2005, pp. 56–59. New York: IEEE.
9. Russell CB, Torralba A, Murphy KP, et al. LabelMe: a database and web-based tool for image annotation. *Int J Comput Vision* 2008; 77(1–3): 157–173.
10. Vondrick C, Patterson D and Ramanan D. Efficiently scaling up crowdsourced video annotation. *Int J Comput Vision* 2013; 101(1): 184–204.
11. Bianco S, Ciocca G, Napoletano P, et al. An interactive tool for manual, semi-automatic and automatic video annotation. *Comput Vis Image Und* 2014; 131: 88–99.
12. Christensen J, Marks J and Shieber S. Placing text labels on maps and diagrams. *Gr Gems* 1994; 4, https://www.eecs.harvard.edu/shieber/Biblio/Papers/jc.label.pdf
13. Campadelli P, Posenato R and Schettini R. An algorithm for the selection of high-contrast color sets. *Color Res Appl* 1999; 24(2): 132–138.
14. Kelly KL. Twenty-two colors of maximum contrast. *Color Eng* 1965; 3(26): 26–27.
15. Carter RC and Carter EC. High-contrast sets of colors. *Appl Optics* 1982; 21(16): 2936–2939.
16. Carter RC and Carter EC. Color coding for rapid location of small symbols. *Color Res Appl* 1988; 13(4): 226–234.
17. Corte WD. Ergonomically optimal CRT colours for nonfixed ambient illumination conditions. *Color Res Appl* 1988; 13(5): 327–331.
18. Corte WD. Recent developments in the computation of ergonomically optimal contrast sets of CRT colours. *Displays* 1990; 11(3): 123–128.
19. Louis DS, James SL, Robert CC, et al. Modeling of display color parameters and algorithmic color selection. In: *Proceedings of the OE/LASE'86*, Los Angeles, CA, 23–24 January 1986, pp. 26–35. Bellingham, WA: International Society for Optics and Photonics.
20. Van Laar D and Flavell R. A name based algorithm for generating maximally discriminable colour sets. *J Photogr Sci* 1994; 42(3): 83–84.
21. Campadelli P, Mora P and Schettini R. Color set selection for nominal coding by hopfield networks. *Visual Comput* 1995; 11(3): 150–155.
22. Carter R and Huertas R. Ultra-large color difference and small subtense. *Color Res Appl* 2010; 35(1): 4–17.
23. Glasbey C, Van der Heijden G, Toh VFK, et al. Colour displays for categorical images. *Color Res Appl* 2007; 32(4): 304–309.
24. Breslow LA, Gregory Trafton J, Malcolm McCurry J, et al. An algorithm for generating color scales for both categorical and ordinal coding. *Color Res Appl* 2010; 35(1): 18–28.
25. Rodríguez-Pardo EC and Sharma G. Adaptive color visualization for dichromats using a customized hierarchical palette. In: *Proceedings of the IS& T/SPIE electronic imaging*, San Francisco, CA, 23 January 2011, paper no. 786603. International Society for Optics and Photonics, http://www.ece.rochester.edu/~gsharma/papers/PardoAdaptColorVisualizeDichromat_EI2011.pdf
26. Radlak K and Smolka B. Visualization enhancement of segmented images using genetic algorithm. In: *Proceedings of the international conference on multimedia computing and systems*, Marrakech, Morocco, 14–16 April 2014, pp. 391–396. New York: IEEE.
27. Rheingans P and Tebbs B. A tool for dynamic explorations of color mappings. In: *Proceedings of the 1990 symposium on interactive 3D graphics*, Snowbird, UT, 25–28 March 1990, pp. 145–146. New York: ACM.
28. Bergman LD, Rogowitz BE and Treinish LA. A rule-based tool for assisting colormap selection. In:

Proceedings of the 6th conference on visualization'95, Atlanta, GA, 29 October–3 November 1995, p. 118. New York: IEEE.
29. MacDonald LW. Using color effectively in computer graphics. *IEEE Comput Graph* 1999; 19(4): 20–35.
30. Campadelli P, Schettini R and Zuffi S. A system for the automatic selection of conspicuous color sets for qualitative data display. *IEEE T Geosci Remote* 2001; 39(10): 2283–2286.
31. Harrower M and Brewer CA. Colorbrewer.org: an online tool for selecting colour schemes for maps. *Cartogr J* 2003; 40(1): 27–37.
32. Ren X and Malik J. Learning a classification model for segmentation. In: *Proceedings of the ninth IEEE international conference on computer vision*, Nice, 13–16 October 2003, pp. 10–17. New York: IEEE.
33. Mori G, Ren X, Efros AA, et al. Recovering human body configurations: combining segmentation and recognition. In: *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition*, Washington, DC, 27 June–2 July 2004, vol. 2, pp. II-326–II-333. New York: IEEE.
34. Shi J and Malik J. Normalized cuts and image segmentation. *IEEE T Pattern Anal* 2000; 22(8): 888–905.
35. Borji A and Itti L. State-of-the-art in visual attention modeling. *IEEE T Pattern Anal* 2013; 35(1): 185–207.
36. Zhang J and Sclaroff S. Saliency detection: a Boolean map approach. In: *Proceedings of the IEEE international conference on computer vision (ICCV)*, Sydney, NSW, Australia, 1–8 December 2013, pp. 153–160. New York: IEEE.
37. Bianco S and Citrolo AG. High contrast color sets under multiple illuminants. In: Tominaga S, Schettini R and Trémeau A (eds) *Computational color imaging*. Berlin, Heidelberg: Springer, 2013, pp. 133–142.
38. Wyszecki G and Stiles WS. *Color science: concepts and methods, quantitative data and formulas*. Oxford: John Wiley & Sons, 1967.
39. Everingham M, Van Gool L, Williams CKI, et al. The PASCAL visual object classes challenge 2012 (VOC2012) results, http://www.pascal-network.org/?q=node/874
40. Fairchild MD. *Color appearance models*. New York: John Wiley & Sons, 2013.
41. Nathan M, Fairchild M, Hunt R, et al. The CIECAM02 color appearance model. In: *Proceedings of the color and imaging conference*, Scottsdale, AZ, 12–15 November 2002, pp. 23–27. Society for Imaging Science and Technology, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.844.5693&rep=rep1&type=pdf
42. Luo MR, Cui G and Li C. Uniform colour spaces based on CIECAM02 colour appearance model. *Color Res Appl* 2006; 31(4): 320–330.
43. Sommer C, Straehle C, Kothe U, et al. Ilastik: interactive learning and segmentation toolkit. In: *Proceedings of the 2011 IEEE international symposium on biomedical imaging: from nano to macro*, Chicago, IL, 30 March–2 April 2011, pp. 230–233. New York: IEEE.
44. A universal labeling tool: Sloth, https://cvhci.anthropomatik.kit.edu/~baeuml/projects/a-universal-labeling-tool-for-computer-vision-sloth/
45. Bianco S and Ciocca G. User preferences modeling and learning for pleasing photo collage generation. *ACM Trans Multimed Comput Comm Appl* 2015; 12(1): 1–23.
46. Moon P and Spencer DE. Geometric formulation of classical color harmony. *J Opt Soc Am* 1944; 34(1): 46–59.

Appendix 1

In the following, we report the sRGB coordinates of the high-contrast colors selected in Figures 4 and 10 (Tables 6–11). We also report in Figure 11 some examples of the effect of setting the transparency level of the overlay solutions as well as the saturation of the underlying image.

Table 6. sRGB coordinates of the high-contrast colors selected in Figure 4 for the gray-scale palette using CIELab as the working color space.

no.	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	128	128	128	85	85	85	51	51	51	32	32	32	23	23	23
3	255	255	255	170	170	170	102	102	102	64	64	64	46	46	46
4				255	255	255	153	153	153	96	96	96	70	70	70
5							204	204	204	128	128	128	93	93	93
6							255	255	255	159	159	159	116	116	116
7										191	191	191	139	139	139
8										223	223	223	162	162	162
9										255	255	255	185	185	185
10													209	209	209
11													232	232	232
12													255	255	255

Table 7. sRGB coordinates of the high-contrast colors selected in Figure 4 for the iso-luminance palette using CIELab as the working color space.

no.	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	255	66	0	255	126	34	0	208	174	183	158	255	251	117	255
2	0	225	0	255	61	247	255	109	122	255	126	34	255	59	143
3	29	178	255	0	224	0	255	70	255	0	225	0	255	65	17
4				0	212	255	0	225	0	211	176	147	165	202	0
5							231	180	1	255	70	255	251	155	133
6							152	176	255	0	217	155	180	187	126
7										0	212	255	244	172	44
8										255	89	142	223	162	220
9										198	191	0	0	225	0
10													87	185	255
11													0	214	208
12													0	210	120

Table 8. sRGB coordinates of the high-contrast colors selected in Figure 4 for the web-safe palette using CIELab as the working color space.

no.	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	0	255	0	0	204	255	0	0	0	255	255	0	0	102	0
2	255	0	0	204	0	0	255	0	153	255	0	0	153	0	51
3	0	0	255	0	255	0	0	153	255	0	0	0	0	0	0
4				0	0	255	102	204	153	0	0	204	255	102	0
5							204	102	0	0	255	255	0	255	0
6							102	0	255	255	0	204	0	0	102
7										153	102	0	0	255	204
8										0	255	102	102	204	255
9										255	204	255	255	0	204
10													255	255	0
11													0	0	255
12													255	204	153

Table 9. sRGB coordinates of the high-contrast colors selected in Figure 10 for the gray-scale palette using CIECAM02-UCS as the working color space under the viewing condition 1 (see Table 4).

no.	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	128	128	128	85	85	85	51	51	51	32	32	32	23	23	23
3	255	255	255	170	170	170	102	102	102	64	64	64	46	46	46
4				255	255	255	153	153	153	96	96	96	70	70	70
5							204	204	204	128	128	128	93	93	93
6							255	255	255	159	159	159	116	116	116
7										191	191	191	139	139	139
8										223	223	223	162	162	162
9										255	255	255	185	185	185
10													209	209	209
11													232	232	232
12													255	255	255

Table 10. sRGB coordinates of the high-contrast colors selected in Figure 10 for the iso-luminance palette using CIECAM02-UCS as the working color space under the viewing condition 1 (see Table 4).

no.	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	0	212	255	255	77	255	0	225	0	255	60	126	246	155	186
2	247	173	9	255	126	0	233	180	0	255	111	0	255	61	109
3	255	61	247	0	225	0	249	155	151	167	159	255	207	163	247
4				0	194	255	146	191	161	0	225	0	236	171	112
5							228	148	255	121	195	160	255	93	0
6							0	204	255	254	115	255	181	181	181
7										212	185	35	110	168	255
8										0	206	255	255	61	247
9										200	175	182	0	223	0
10													0	213	240
11													0	214	156
12													195	190	32

Table 11. sRGB coordinates of the high-contrast colors selected in Figure 10 for the web-safe palette using CIECAM02-UCS as the working color space under the viewing condition 1 (see Table 4).

no.	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	0	255	0	0	0	0	204	255	0	153	0	204	0	153	255
2	255	0	51	255	0	0	0	102	0	0	153	255	255	0	102
3	0	0	153	153	153	255	0	0	153	51	0	0	102	0	0
4				0	204	0	0	204	255	255	204	255	255	153	51
5							255	102	255	255	204	0	204	255	0
6							153	0	0	0	102	0	0	255	255
7										0	0	102	0	0	153
8										0	255	204	102	102	102
9										255	0	0	0	0	0
10													255	204	255
11													153	0	204
12													0	153	0

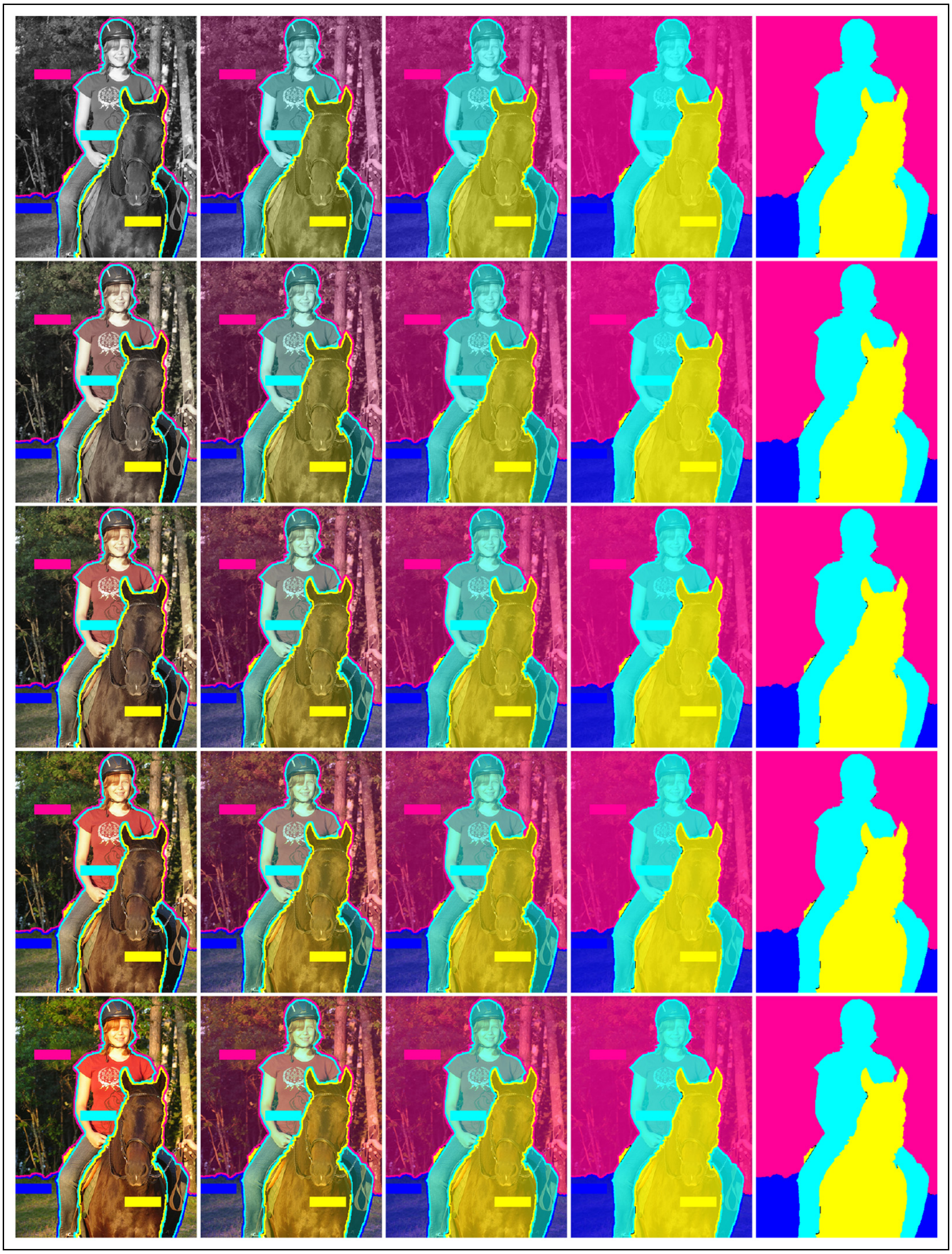

Figure 11. Different proposals for the overlay solution using the web-safe palette. The transparency of the overlaid colors is changed in five steps (from 100% to 0%) along the different columns; the saturation of the underlying image is changed in five steps (from 0% to 100%) along the different rows.