Learning Color Names from Real-World Images

Joost van de Weijer, Cordelia Schmid, Jakob Verbeek
INRIA, LEAR
655 Avenue de l'Europe, Montbonnot 38330, France
{Joost.van-de-weijer, Cordelia.Schmid, Jakob.Verbeek}@inrialpes.fr

Abstract

Within a computer vision context color naming is the action of assigning linguistic color labels to image pixels. In general, research on color naming applies the following paradigm: a collection of color chips is labelled with color names within a well-defined experimental setup by multiple test subjects. The collected data set is subsequently used to label RGB values in real-world images with a color name. Apart from the fact that this collection process is time consuming, it is unclear to what extent color naming within a controlled setup is representative for color naming in real-world images. Therefore we propose to learn color names from real-world images. Furthermore, we avoid test subjects by using Google Image to collect a data set. Due to limitations of Google Image this data set contains a substantial quantity of wrongly labelled data. The color names are learned using a PLSA model adapted to this task. Experimental results show that color names learned from real-world images significantly outperform color names learned from labelled color chips on retrieval and classification.

1. Introduction

Color names are linguistic labels that humans attach to colors. We use them routinely and seemingly without effort to describe the world around us. They have been primarily studied in the fields of visual psychology, anthropology and linguistics [1]. Color naming is different from the thoroughly explored field of color imaging, where the main goal is to decide, given an acquisition of an object with a certain color, if objects in other acquisitions have the same (or a different) color. Based on physical or statistical models of reflection and acquisition systems [2, 3, 4] object colors can be described independent of incidental scene events such as illuminant color and viewing angle. The research question of color naming is different: given a color measurement, the algorithm should predict with which color name humans would describe it. It also allows for different functionalities, for example within a content based retrieval context it allows to steer the search to objects of a certain color name. The user might query an image search engine for "red cars".The system recognizes the color name "red", and orders the retrieved results on "car" based on their resemblance to the human usage of "red'. Apart from the retrieval task color names are applicable in automatic content labelling of images, colorblind assistance, and linguistic human-computer interaction [5].

One of the most influential works in color naming is the linguistic study of Berlin and Kay [6] on basic color terms. They are defined as those color names in a language which are applied to diverse classes of objects, whose meaning is not subsumable under one of the other basic color terms, and which are used consistently and with consensus by most speakers of the language. Basic color names were found to be shared between languages. However the number of basic terms varies from two in some Aboriginal languages to 12 in Russian. In this paper, we use the 11 basic color terms of the English language: black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow.

To use color naming for computer vision a mapping between the RGB values and color names is required. Generally this mapping is inferred from a labelled set. Multiple test subjects are asked to label hundreds of color chips within a well-defined experimental setup [7, 8, 9, 10]. The colors are to be chosen from a preselected set of color names (predominantly the set of 11 basic color terms [8, 10]). From this labelled set of color chips the mapping from RGB values to color names is derived. Throughout the paper we will refer to this methodology of color naming as *chip-based color naming*.

We do not wish to cast doubt on the usefulness of chip-based color naming within the linguistic and color science fields, however it might be questioned to what extent the labelling of isolated color chips resembles color naming in the real-world. Color naming chips under ideal lighting on a color neutral background greatly differs from the challenge of color naming in the real-world without a neutral reference color and with physical variations such as shading effects and different light sources. Another disadvan-

1-4244-1180-7/07/$25.00 ©2007 IEEE

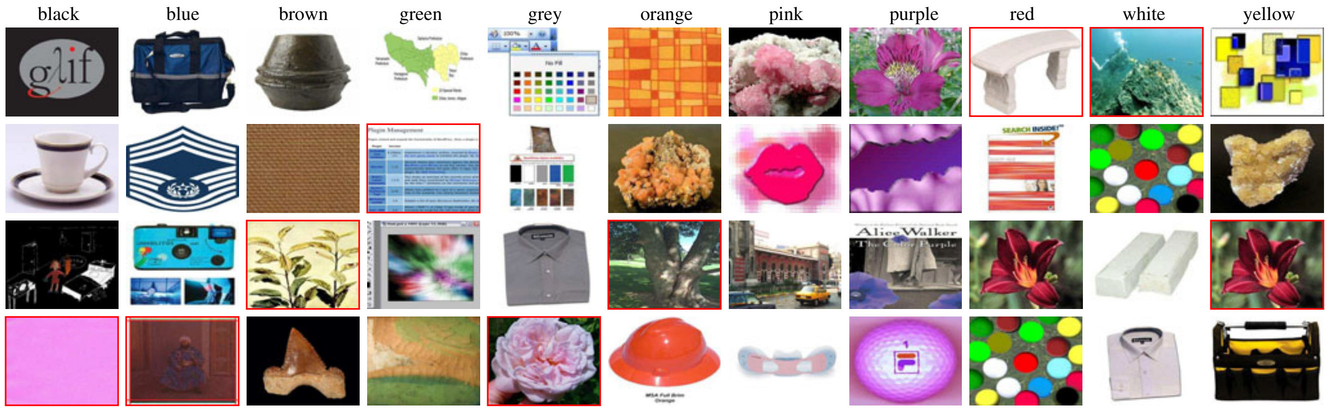


Figure 1. Google-retrieved examples for color names. The red bounding boxes indicate false positive images. The same images can be retrieved with various color names, such as the flower image which appears in the red and the yellow set.

tage of chip-based color naming is its inflexibility with respect to changes of the used color name set. This is caused by the demanding experimental setup necessary for chip-based color naming. Changing the set by adding for example beige, violet or olive, would imply rerunning the experiment for all patches.

We propose an alternative method to color naming. To overcome, at least to some extent, the limitations of chip-based color naming, we propose to learn color names from real-world images. Furthermore, to design a flexible system with respect to variations in the color name set, we propose to automatically learn the color names from images retrieved from Google image search (see Fig. 1). The use of image search engines to avoid hand labelling was pioneered by Fergus et al. [11]. Retrieved images from Google search are known to contain many false positives. To learn color names from the Google images we propose to use Probabilistic Latent Semantic Analysis (PLSA), a generative model introduced by Hofmann [12] for document analysis. This model was recently applied to computer vision by [13], [14], [15]. We model RGB values (words) in images (documents) with mixtures of color names (topics), where mixing weights may differ per image but the topics are shared among all images. In conclusion, by learning color names from real-world images, we aim to derive color names which are applicable on challenging images typical in computer vision applications. In addition, since its knowledge on color names is derived from an image search engine, the method can easily vary the set of color names.

Our method is closely related with work on finding relations between words and image regions. Barnard et al. [13] proposed a method to learn the joint distribution between words and image regions. The original work which was limited to nouns was later extended to also include adjectives by Yanai and Barnard [16]. They compute the "visualness" of adjectives, based on the entropy between adjectives and image features. The work shows among other adjectives results for color names: several of these are correctly found to be visual, however the authors also report failure for some color names. Contrary to this work, we start from the prior-knowledge that color names are "visual" and that they should be learned from the color distributions (and not for example from texture features), with the aim to improve the quality of the learned color names.

This paper is organized as follows. In Section 2, the color name data sets used for training and testing are presented. In Section 3, several color related issues are discussed. In Section 4, our approach to learning color names from images is presented. In Section 5, experimental results are given, and Section 6 finishes with concluding remarks.

2. Color Name Data Sets

For the purpose of learning color names from real-world images, a set of labelled images is required. We use three data sets, two of which were collected specifically for the work presented here and are made available online at http://lear.inrialpes.fr/data, we briefly describe them below.

Google color name set: Google image search uses the image filename and surrounding web page text to retrieve the images. As color names we choose the 11 basic color names as indicated in the study of Berlin and Kay [6]. We used Google Image to retrieve 100 images for each of the 11 color names. For the actual search we added the term "color", so for red the query is "red+color". Examples for the 11 color names are given in Fig. 1. Per color name there are on average 19 false positives, i.e., images which do not contain the color of the query. Furthermore in many cases only a small portion, as little as a few percent of the pixels, represents the color label. Our goal is to arrive at a color naming system based on the raw results of Google image, i.e., we used both false and true positives.

Figure 2. Examples for the four classes of the Ebay data set: blue cars, grey shoes, yellow dresses, and brown pottery. For all images masks with the area corresponding to the color name are hand segmented. One example segmentation per category is given.

Ebay color name set: To test the color names a human-labelled set of object images is required. We used images from the auction website Ebay. Users labelled their objects with a description of the object in text, often including a color name. We selected four categories of objects: cars, shoes, dresses, and pottery (see Fig. 2). Of each object category 110 images where collected, 10 for each color name. The images contain several challenges: the reflection properties of the objects differ from matt reflection of dresses to highly specular surfaces of cars and pottery. Furthermore, it comprises both indoor and outdoor scenes. For all images we hand-segmented the object areas which correspond to the color name. In the remainder of the article when referring to Ebay images, only the hand segmented part of the images is meant, and the background is discarded.

Chip-based color name set: In the experimental Section we compare our method to a chip-based approach. For this purpose we use the data set of color named chips of Benavente [10], which is available online. The set contains 387 patches which were classified into the 11 basic color terms by 10 subjects. If desired the color patch could be assigned to multiple color names. Thus every patch is represented by its sRGB values (standard default color space) and a probability distribution over the color names. To arrive at a probability over the color names, z, for all $L^*a^*b^*$-bins (we use the same discretization as is applied in our algorithm), we assign to each $L^*a^*b^*$-bin, w, the probability of the neighbors according to

$$P\left(z\left|w\right.\right) \propto \sum_{i=1}^{N} P\left(z\left|w_i\right.\right) g^{\sigma}\left(\left|w_i - w\right|^{LAB}\right) \qquad (1)$$

where the w_i's are the $L^*a^*b^*$-values for the color chips and N is total number of chips. $P\left(z\left|w_i\right.\right)$ is given for all the color chips. The distance between the color chips, w_i, and w is computed in $L^*a^*b^*$-space. For the weighting kernel we use a Gaussian with $\sigma = 5$, which has been optimized to get the best results on the retrieval task of Section 5.1.

3. Color Considerations

The images are represented in the form of color histograms to the learning algorithm. We consider the images from the Google and Ebay data sets to be in $sRGB$ format. Before computing the color histograms these images are gamma corrected with a correction factor of 2.4. Although images might not be correctly white balanced, we refrained from applying a color constancy algorithm. This is motivated by the fact that state-of-the-art color constancy often gives unsatisfying results [17]. Furthermore, many Google images lack color calibration information, and regularly break assumptions on which color constancy algorithms are based. For example many of the images consist of single colored objects on a background, for which most color constancy methods fail.

For the color histograms we considered several color spaces: *RGB*, *HSL*, and $L^*a^*b^*$. *HSL* is attractive because of its axis-alignment with photometric variations [7, 18]. Decision on chromatic versus achromatic colors can be based on luminance and saturation, whereas chromatic colors can be distinguished based on hue and saturation. Yet, some colors have the same hue but different intensities, e.g., orange and brown. To efficiently use *HSL* the subspace of the *HSL*-space in which the color name is located should be given as extra information. This is opposite to our aim to automatically learn the color names from Google images. The $L^*a^*b^*$ color space instead seems like an appropriate choice, as it is perceptually linear, meaning that similar differences between $L^*a^*b^*$ values are considered about equally important color changes to humans. This is a desired property because the uniform binning we apply for histogram construction implicitly assumes a meaningful distance measure. To compute the $L^*a^*b^*$ values we assume a D65 white light source. Note that the $L^*a^*b^*$ color space is not photometrically invariant. Changes of the intensity influence all three coordinates. By choosing to learn the color names in the $L^*a^*b^*$-space we hope that the "partial" photometric invariance of color names is acquired in the learning phase. In our experiments we show that within our context the $L^*a^*b^*$-space indeed outperforms both the RGB and HSL-space.

4. Learning Color Names

The learning of the color names is achieved with the PLSA model [12]. The PLSA model is appropriate since it allows for multiple "classes" in the same image, which is the case in our Google data set. In analogy with its use in text analysis, where the PLSA model is used to discover topics in a bag-of-word representation, we here apply it to discover colors in a bag of pixels representation, where every pixel is represented by its $L^*a^*b^*$ value. In order to use the PLSA model we discretize the $L^*a^*b^*$ values into a finite vocabulary by assigning each value by cubic interpolation to a regular $10 \times 20 \times 20$ grid in the $L^*a^*b^*$-space[1]. An image (document) is then represented by a histogram indicating how many pixels are assigned to each bin (word).

4.1. Generative Models: PLSA and PLSA-bg

In text analysis the PLSA model is used to find a set of semantic topics in a collection of documents. Here we use the model to find a set of color names (comparable to the topics in documents) in a collection of images. The use of generative models to learn the relation between images and words was first proposed by Barnard et al. [13]. They apply Latent Dirichlet Allocation (LDA) to learn relations between words and image blobs. We start by explaining the standard PLSA, after which we propose an adapted version suited to our problem. We follow the terminology of the text analysis community.

Given a set of documents $D = \{d_1, ..., d_N\}$ each described in a vocabulary $W = \{w_1, ..., w_M\}$, the words are taken to be generated by latent topics $Z = \{z_1, ..., z_K\}$. In the PLSA model the conditional probability of a word w in a document d is given by:

$$P\left(w|\, d\right) = \sum_{z \in Z} P\left(w|\, z\right) P\left(z|\, d\right). \tag{2}$$

Both distributions $P(z|d)$ and $P(w|z)$ are discrete, and can be estimated using an EM algorithm [12]. This standard PLSA model does not exploit the labels of the images. The topics are hoped to converge to the desired color names. As is pointed out in [19] this is rarely the case. To overcome this shortcoming we propose an adapted PLSA model.

PLSA-bg: We propose to model an image d as being generated by two distributions: the foreground distribution which is determined by its color name label l_d and the background distribution which is shared between all images:

$$P\left(w\,|d, l_d = z\right) = \alpha_d P\left(w\,|l_d = z\right) + \left(1 - \alpha_d\right) P\left(w\,|bg\right), \tag{3}$$

where $P\left(w\,|l_d = z\right)$ is the probability that the word is generated by topic l_d. We use the following shorthands: $P\left(w\,|d, l_d = z\right) = p_{wd}$, $P\left(w\,|z\right) = \beta_{wz}$ and $P\left(w\,|bg\right) = \theta_w$. To learn the model we need to estimate the mixing proportion of foreground versus background α, the color name distributions β, and the background model θ. With each word in each document we associate a hidden variable with two states, that indicates whether the word was drawn from the foreground topic or the background topic. The posterior on the states is calculated as:

$$\begin{array}{ll} q_{wd}^{fg} = \frac{\alpha_d \beta_{wz}}{p_{wd}} & \text{(foreground)}, \\ q_{wd}^{bg} = 1 - q_{wd}^{fg} = \frac{(1-\alpha_d)\theta_w}{p_{wd}} & \text{(background)}. \end{array} \tag{4}$$

By maximizing the complete data log-likelihood

$$Q = \sum_{w,d} c_{wd} \left(q_{wd}^{fg} \log \alpha_d \beta_{wl_d} + q_{wd}^{bg} \log\left(1 - \alpha_d\right) \theta_w \right), \tag{5}$$

and given the q's, we can re-estimate the parameters:

$$\begin{aligned} \alpha_d &= \left[\sum_w c_{wd} \right]^{-1} \sum_w c_{wd} q_{wd}^{fg} \\ &= \alpha_d \left[\sum_w c_{wd} \right]^{-1} \sum_w \frac{c_{wd}}{p_{wd}} \beta_{wz}, \end{aligned} \tag{6}$$

and after updating p_{wd} with the new α's:

$$\beta_{wz} \propto \sum_{d:\ l_d = t} c_{wd} q_{wd}^{fg} = \beta_{wz} \sum_{d:\ l_d = t} \alpha_d \frac{c_{wd}}{p_{wd}}, \tag{7}$$

$$\theta_w \propto \sum_d c_{wd} q_{wd}^{bg} = \theta_w \sum_d \left(1 - \alpha_d\right) \frac{c_{wd}}{p_{wd}}, \tag{8}$$

where D^t is the set of documents for which label l_d is equal to t, and c_{wd} is the normalized word count per document. The method can be extended to allow for multiple shared background topics. However, we found that increasing the number of background topics did not improve performance.

Predicting color names: We will apply the derived word-topic distributions to assign color name probability to image pixel values. Two ways to assign color names to *individual pixels* are considered: based only on the pixel value, indicated by PLSA-bg, or by also taking the region of the pixel into account, abbreviated with PLSA-bg*. The probability of a color name given a pixel is

$$\mathrm{PLSA-bg}: \quad P\left(z\,|w\right) \propto P\left(z\right) P\left(w\,|z\right), \tag{9}$$

where the prior over the color names is taken to be uniform. The probability of a color name given the region is computed with

$$\mathrm{PLSA-bg}^*: \quad P\left(z\,|w, d\right) \propto P\left(w\,|z\right) P\left(z\,|d\right), \tag{10}$$

where $P\left(z\,|d\right)$ is estimated using an EM algorithm by taking the word topic distribution $P\left(w\,|z\right)$ fixed. The background topic is disregarded in this phase, since we know

[1]The difference in bins is caused by the different domains. The intensity axis ranges from 0 to 100, the chromatic axes range from -100 to 100.

that the 11 color names describe the whole RGB cube. The difference between the two methods is that for PLSA-bg* first a distribution over the color names given the region is computed $P(z|d)$. This distribution is subsequently used as a prior over the color names in the computation of the probability of a color name given a pixel value and the region. In the context of scene classification, Quelhas et al. [15] also consider these two methods to compute the conditional probability of topics given a word. They found retrieval results to improve by taking $P(z|d)$ into account.

To arrive at a probability distribution over the color names for an *image region* (e.g., the segmentation masks in the Ebay image set) we use the topic distribution over the region $P(z\,|d)$ described above for PLSA-bg*. For PLSA-bg the probability over the color names for a region is computed by a simple summation over all pixels in the region of the probabilities $P(z|w)$, computed with Eq. 9 using a uniform prior. Both PLSA-bg and PLSA-bg* will be compared in the experimental section on their usefulness for retrieving colored objects and assigning color names to pixels.

5. Experimental Results

In the introduction we argued against learning color names in a different setup than in which they are applied. Therefore, we propose a method which learns color names from real-world images. Furthermore, to arrive at a flexible method with respect to variations in the color name set we propose to learn the color names from Google images. In the experiments the proposed method is compared to a chip-based method (see Section 2). We evaluate them based on color object retrieval, and on the assignment of color names to pixels. Furthermore, we illustrate the flexibility of our method when changing the set of color names.

To verify our learning approach we compare PLSA-bg learning with two alternatives. Firstly, a standard PLSA with 11 topics. With random initialization the topics rarely coincided with the color names, and the method performed poorly. A better way to initialize the word-topic distributions $P(w|z)$ is to average for each topic the empirical distribution over words of all documents labelled with the class associated with that topic. Secondly, a linear support vector machine [20], which is trained on the $L^*a^*b^*$ histograms of the Google images. In retrieval we classify the histograms of the segmented regions, and derive a probability over the color names from the SVM scores (Section 5.1). For individual pixel classification (Section 5.2), the SVM classifies histograms of individual pixels. [2]

5.1. Retrieving Objects by Color

The different approaches to color naming are evaluated on retrieval of colored objects within each category. For example, the car category is queried for ”red cars”. We query the four categories of the Ebay set (see Section 2) for the 11 color names. The images are retrieved based on the probability of the query color given the Ebay images. For the Ebay images only pixels within the segmentation masks are considered. To assess the performance we compute the equal error rate (EER) for each query. The average EER's over the 11 color names are reported for each category in Table 2. The learning of the color names is performed on the weakly labelled Google images, or a combination of the Google and Ebay images. In the case of the combined Google and Ebay images the Ebay category which is queried is left out in training.

method	RGB	HSL	L*a*b*
EER	94	93	95

Table 1. Equal error rates on Ebay set of the PLSA-bg* method, learned on Google images based on three different color spaces.

method	train-set	cars	shoes	dresses	pottery	overall
chip-based	-	88	93	94	91	92
SVM	Google	91	96	96	91	94
PLSA	Google	89	95	94	92	93
PLSA-bg*	Google	91	96	99	93	95
PLSA-bg	Google	92	97	99	95	96
PLSA-bg*	Google+Ebay	92	96	99	95	96
PLSA-bg	Google+Ebay	92	97	100	94	96

Table 2. Average equal error rates for retrieval on Ebay images.

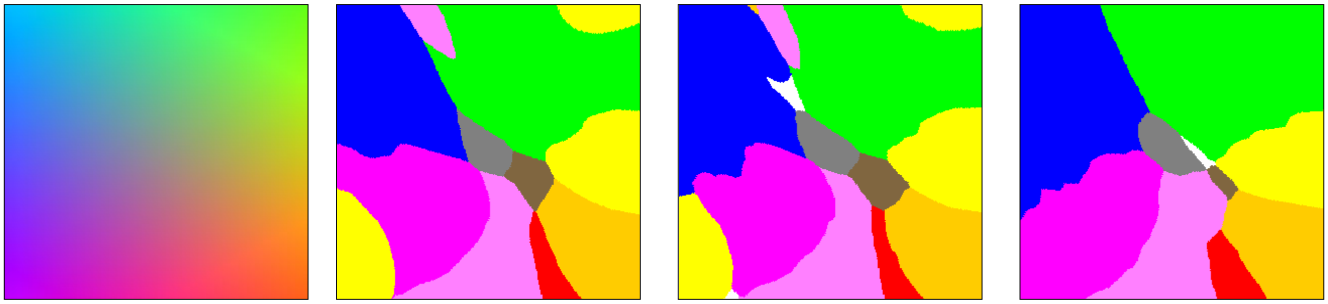
We first verify our choice of color space. The results for retrieval based on three color spaces are given in Table 1. As expected the $L^*a^*b^*$-space slightly outperforms the other color spaces. In the remainder of the article only results based on the $L^*a^*b^*$-space are reported.

The results of the various color naming methods are summarized in Table 2. The results support our idea that learning color names from real-world images is sensible: all learning methods outperform the chip-based method for all four categories. Secondly, results show that it is possible to learn the color names from highly polluted images retrieved with an image search engine. Thirdly, the proposed adaptations of the PLSA model are beneficial: they obtain significantly better results than both standard PLSA and SVM. Extending the training set with the Ebay images did not improve results over training on Google images only.

5.2. Pixelwise Color Name Classification

As a second experiment, the color naming methods are compared on classification of pixels in the Ebay images. All pixels within the segmentation masks are assigned to their

[2] Cubic interpolation maps the color value of a pixel to multiple bins.



(a) (b) (c) (d)

Figure 3. (a) A challenging synthetic image: the RGB values at the border rarely occur in natural images. Results obtained with (b) PLSA-bg learned on Google images, (c) SVM learned on Google and Ebay, and (d) PLSA-bg learned on Google and Ebay.

method	train-set	cars	shoes	dresses	pottery	overall
chip-based	-	39	60	62	50	53
SVM	Google	45	61	68	56	62
PLSA	Google	48	69	71	62	63
PLSA-bg*	Google	63	84	89	74	78
PLSA-bg	Google	51	71	81	66	67
PLSA-bg*	Google+Ebay	68	88	93	80	82
PLSA-bg	Google+Ebay	53	73	84	71	70

Table 3. Pixel classification in percentages for Ebay images.

	black	blue	brown	grey	green	orange	pink	purple	red	white	yellow
black	78	1	2	19							
blue	3	90		4				3			
brown	6		67	16			1		8	2	
grey	3		5	87						5	
green	4	1	1	9	81						4
orange			5			80	2		13		
pink			6	1			78	9	2	4	
purple	12	1	1	13			4	67	0	2	
red			2				4		94		
white				12	1					87	
yellow	1		1							2	96

Table 4. Confusion matrix for pixel classification based on PLSA-bg* learned on Google and Ebay. The overall classification rate is 82% as given in Table 3.

most likely color name according to $\arg\max_z P(z\,|w)$. Only for PLSA-bg* we take the surrounding of the pixel into account: we use $\arg\max_z P(z\,|w,d)$ to classify the pixel, where the segmentation mask forms the document.

In Table 3 the results are presented. For this task the gain in learning color names from real-world images is impressive: where the chip-based method classifies only 53% of the pixels correctly, the PLSA-bg* obtains a result of 82%. Again the proposed method outperforms standard PLSA and linear SVM. For this task, adding the Ebay images to the training data improved the overall classification rate for PLSA-bg* from 78% to 82%.

We believe that the difference in performance of PLSA-bg* and PLSA-bg methods in retrieval and pixel classification is caused by the fact that the PLSA-bg* couples the topic assignment of pixels within an image. This proved to be beneficial for pixel classification. On the other hand when using an image specific topic distribution, the correct classification of small regions in an image may be hindered in the presence of a large region of a similar color in the same image (e.g., a small orange region within a dominantly yellow image). Retrieving this image on the color name of the small color region (in the example orange) will result in deteriorated retrieval performance. This explains the slightly worse retrieval results for PLSA-bg*. This contrasts results in [15], where taking the topic-document distribution into account proved beneficial for retrieval of man-made versus landscape images.

To give some insight into the errors made in color naming we show the confusion matrix of the pixel classifications based on PLSA-bg* in Table 4. Most confusions are reasonable: confusions between achromatic colors (white-grey and black-grey), between colors and achromatic colors (purple-black, purple-grey, brown-grey) or between very similar colors (red-orange, pink-purple).

In Fig. 3 a challenging image shows that the color names learned from the Google images alone are still inaccurate for some highly saturated colors (see Fig. 3b). These colors occur rarely in natural images, because they require one of the color channels to be near zero. Color names learned from the combined set of Google and Ebay images provide reliable results even for highly saturated colors (see Fig. 3d).

The PLSA-bg model learned on the Google and Ebay images is available online at http://lear.inrialpes.fr/people/vandeweijer/color_names.html, in the form of a $32 \times 32 \times 32$ lookup table which maps sRGB values to probabilities over color names.

Limitations of Chip-Based Color Naming: For both retrieval and pixel classification color names learned from real-world images outperformed chip-based color naming. Here we analyze the reasons underlying this difference.

We used the PLSA-bg color naming model learned on Google and Ebay images to classify the color chips from [10]. Amazingly, for 77 out of 387 chips (20%) our method appointed a color name to the chip which was not matched by any of the test subjects. However, 24 of these concern disagreement between achromatic colors, primarily caused by grey-labelled chips which were classified by PLSA-bg as white. For 51 color chips, labelled with chromatic color names, our method appointed achromatic color names. Only in two cases there was disagreement between chromatic colors (orange-yellow, pink-brown). In conclusion, differences in color naming mainly occur for low saturated colors, which in a laboratory setup on a neutral background are often considered as chromatic. However, in real-world images these colors are more often caused by inter-reflection, or variations of the light source. By learning the color names from real-world images, slight deviations from perfect grey are not attributed to chromatic colors. Several examples of color name assignments for chip-based, PLSA-bg and PLSA-bg* are depicted in Fig. 5. Other than in the earlier experiment we classify all pixels in the image.

5.3. Flexibility Color Name Data Set

Another drawback of chip-based color naming is the inflexibility with respect to changes of the color names set. An search engine based method is more flexible, since the collection of data is only several minutes of work. In Fig. 4 we show prototypes of the 11 basic color terms based on the patches of the chip-based data set. The first row shows the prototypes computed using the human assigned labels and the second row the prototypes based on the labels assigned by our method. The prototype of a color name is computed by averaging the RGB values of all color chips for which the color name is the most likely. The learned color names from Google have a great visual resemblance with the prototypes based on the human labelled color chips.

Next, we give two examples of varied color name sets. Mojsilovic, in her study on color naming [9], mentions the use of the color names beige, violet and olive, in addition to the 11 basic color terms. For these three classes we retrieved 100 images each, and re-learned the color names for the 14 color terms. The prototypes of the three newly added colors together with a few of the closely related color names are depicted in the bottom row of Fig.4. The newly injected color names also influence the position of the old color names, as can be seen in the slight color shift of the brown and the pink prototype.

As a second example of flexibility we look into inter-

Figure 4. First row: prototypes of the 11 basic color terms based on chip-based color naming. Second row: prototypes of the 11 basic color terms learned from Google images based on PLSA-bg. Third row shows results on a varied set of basic color terms (left and middle group): prototypes of several of the color names learned from Google images using 14 color names: the 11 basic color terms extended with beige, olive and violet. Third row (group to the right): prototypes of the two Russian blues learned from Google images.

linguistic differences in color naming. The Russian language is often mentioned as one of the few languages which has 12 basic color terms. The color term blue is split up in two: goluboi (голубой), and siniy (синий). We ran the system on 30 images for both blues, returned by Google image. Results are given in Fig.4, and correspond with the fact that goluboi is a light blue and siniy a dark blue. The example shows internet as a potential source of data for the examination of linguistic differences in color naming.

6. Discussion and Conclusions

We have presented a new method for color naming. It breaks with the generally accepted approach to learn color names from isolated color chips in a laboratory setting. Furthermore, to obtain a method for which it is easy to vary the set of desired color names, we proposed to learn the color names from Google image. Results show that within the context of computer vision the learning of color names from real-world images is beneficial. The improvement is especially striking in classification of pixels with color names where results go up from 53% to 82%. Furthermore, the flexibility of the method with respect to varying color name sets has been illustrated.

In a wider context this article can be seen as a case study for the automatic learning of visual attributes. In recent years the computer vision community has achieved significant progress in the field of object and object category recognition. Now that it is possible to detect objects such as people, cars, and vases in images, the question arises if we are able to retrieve *small* people, *striped* vases, and *red* cars. The scope of these so called visual attributes is vast: they range from size descriptions, such as large, elongated, and contorted, to texture descriptions such as striped, regular, and smooth, further on to color descriptions, such as red, cyan and pastel. The challenges which arise in the development of an automatic color naming system can be seen as exemplar for the problems which arise for visual attribute

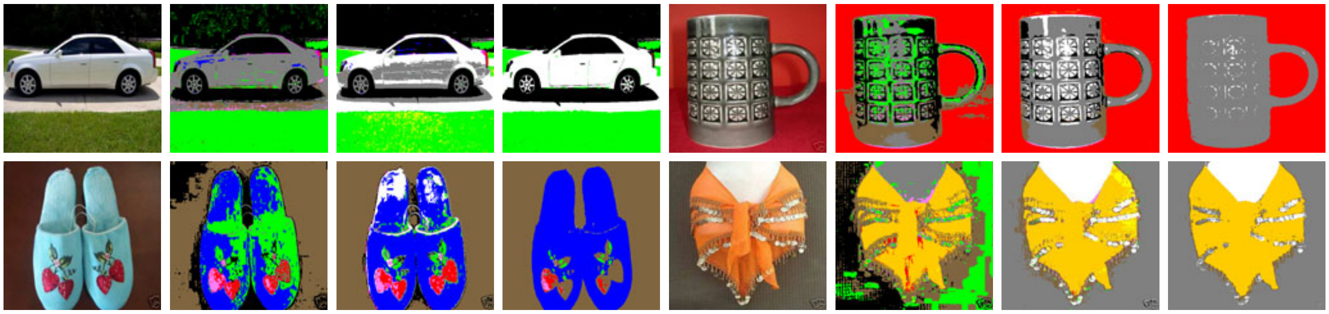


Figure 5. Four examples of pixelwise color name classification. For each example the results of chip-based, PLSA-bg, PLSA-bg* are given successively. Left top: without reference color it is very hard to classify the achromatic colors black, grey and white. As a result the white car is considered grey by the chip-based method. Right top: although the pixel values are slightly greenish, the mug is human labelled as being grey. The slight deviation from grey led to the wrong color label green for the chip-based method. Bottom left: Here some blue pixels were wrongly classified as green, because the 387 color chips sample some areas of the RGB-cube only sparsely. In this example, PLSA-bg* suppresses the small red topic, consequently parts of the strawberries are wrongly considered brown. Bottom right: all methods correctly classify the dress pixels. The chip-based method appoints chromatic color names to the low saturation background.

learning at a large. The labelling of images with the inexhaustible set of existing visual attributes will be unfeasible, and automatic ways to learn them are needed.

7. Acknowledgements

This work is supported by both the Marie Curie Intra-European Fellowship Program and the European funded CLASS project.

References

[1] C.L. Hardin and L. Maffi, editors. *Color Categories in Thought and Language*. Cambridge University Press, 1997.

[2] D.A. Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5(1):5–36, 1990.

[3] B.V. Funt and G.D. Finlayson. Color constant color indexing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:522–529.

[4] Th. Gevers and A. Smeulders. Color based object recognition. *Pattern Recognition*, 32:453–464.

[5] L. Steels and T. Belpaeme. Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Science*, 28:469–529, 2005.

[6] B. Berlin and P. Kay. *Basic color terms: their universality and evolution*. Berkeley: University of California, 1969.

[7] D.M. Conway. An experimental comparison of three natural language colour naming models. In *Proc. east-west int. conf. on human-computer interaction*, pages 328–339, 1992.

[8] J.M. Lammens. *A computational model of color perception and color naming*. PhD thesis, University of Buffalo, 1994.

[9] A. Mojsilovic. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image Processing*, 14(5):690–699, 2005.

[10] R. Benavente, M. Vanrell, and R. Bladrich. A data set for fuzzy colour naming. *COLOR research and application*, 31(1):48–56, 2006.

[11] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. of the IEEE Int. Conf. on Computer Vision*, Beijing, China, 2005.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proc. ACM SIGIR conf. on Research and development in information retrieval*, pages 50–57, 1999.

[13] K. Barnard, P. Duygulu, D.Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.

[14] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *Proc. of the IEEE Int. Conf. on Computer Vision*, 2005.

[15] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van-Gool. Modeling scenes with local descriptors and latent aspects. In *Proc. of the IEEE Int. Conf. on Computer Vision*, 2005.

[16] K. Yanai and K. Barnard. Image region entropy: a measure of "visualness" of web images associated with one concept. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 419–422, New York, NY, USA, 2005. ACM Press.

[17] B. Funt, K. Barnard, and L. Martin. Is machine colour constancy good enough? *Lecture Notes in Computer Science*, 1406:445–459, 1998.

[18] Y. Liu, D. Zhang, G. Lu, and W-Y. Ma. Region-based image retrieval with high-level semantic color names. In *Proc. 11th Int. Conf. on Multimedia Modelling*, 2005.

[19] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization. In *British Machine Vision Conference*, 2006.

[20] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.