Deriving a Discriminative Color Model for a Given Object Class from Weakly Labeled Training Data

Christian X. Ries
Augsburg University
Universitätsstr. 6a
86150 Augsburg, Germany
ries@informatik.uni-augsburg.de

Rainer Lienhart
Augsburg University
Universitätsstr. 6a
86150 Augsburg, Germany
lienhart@informatik.uni-augsburg.de

ABSTRACT

This paper presents a method for creating a discriminative color model for a given object class based on color occurrence statistics. A discriminative color model can be used to classify individual pixels of images with regards to whether they may belong to the wanted object. However, in contrast to existing approaches, we do not exploit pixel-wise object annotations but only global negative and positive image labels. Therefore our approach requires significantly less manual effort. We quantitatively evaluate the performance of our approach on two publicly available datasets and compare it to a baseline approach, which utilizes pixel annotations. The experimental results show that our approach is on par with pixel-wise approaches although requiring only a single global image label.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.4.0 [**Image Processing and Computer vision**]: General

General Terms

Theory

Keywords

Discriminative Color Models, Weakly Labeled Data, Color Histogram, Bayesian Model

1. INTRODUCTION

In this paper we introduce a method for learning a discriminative color model for a given object class from weakly labeled training data. A discriminative color model assigns a Boolean value to each pixel of a query image indicating whether the pixel may belong to an object of the given object class. Thus, the color model yields a binary map of positive and negative pixels. One example of such a map can be seen in Figure 1 on the right where the binary map was filled with the original image content for illustration.

Obviously, a discriminative color model can only be created for objects, which occur in a limited number of different color combinations. Therefore we only consider such objects for the remainder of this paper.

Figure 1: An example result after applying our discriminative color model which was created using weakly labeled training sets. The image shows an example from the Oxford Flowers dataset [8] on the left and the pixels detected as belonging to the flower by our color model on the right.

Color models can be used for pre-filtering images or for creating regions of interest for more sophisticated classification systems. In other words, color models can be useful for identifying or improving positive training examples among large datasets for image classification systems. Also, color models can be used for quickly rejecting unambiguously negative images prior to applying more sophisticated (and thus computationally more expensive) classifiers.

The model we present in this paper relies on color occurrence statistics and can thus be computed efficiently. It is straightforward, yet its performance is on par with existing approaches for almost all of our test classes. Furthermore, to our knowledge, existing approaches, which are comparable with regards to complexity, require pixel-wise annotations. In contrast, our approach only requires positive and negative image labels for the training set. No hints are given as to where the objects of interest are in the positive images.

Thus, the major advantage of our approach over existing approaches is the small amount of manual annotation effort required. Besides reducing the human effort, our approach is applicable to situations where annotating images manually is

ICMR '12, June 5-8, Hong Kong, China
https://doi.org/10.1145/2324796.2324848

not acceptable due to the size of the dataset or the contents of the positive images. Furthermore, pixel-wise annotation is often a tedious and difficult task and sometimes also error-prone at the boundary of the objects.

2. RELATED WORK

Discriminative color models, which are usually based on thresholds either derived from color histograms or parametric functions such as Gaussian mixtures, are among the earliest concepts in the field of computer vision. Yet they are still used successfully in several different contexts today. Especially for detecting and segmenting objects, which have a constant color scheme such as road signs, discriminative color models are useful [2, 4, 11].

Discriminative color models can also be applied in order to efficiently determine regions of interest, which can be used for video coding [3] or image classification [10]. The latter two approaches both use a discriminative color model for detecting human skin, which is a research field where such models are popular [7, 1, 5]. Skin detection, however, is still a difficult problem due to the variance of human skin colors.

Jones and Rehg [7] also employ a discriminative color model to identify human skin in images. Their approach is based on ground truth labeling of pixels. In the survey on skin color detection by Vezhnevets et al. [12] Jones and Rehg's approach is compared to various other approaches on skin color detection. It is among the approaches which yield the best results being only slightly inferior to the maximum entropy approach by Jedynak et al. [6]). To our knowledge, the approach by Jones and Rehg is still among the state-of-the-art methods for creating discriminative color models. Thus we use their approach as a baseline for evaluating the results of our approach. Note, however, while the approach of Jones and Rehg require pixel-wise labeled training images, we only require a single image label.

3. CREATING THE COLOR MODEL

In general, a discriminative color model assigns a classification value $h(c) \in \{0, 1\}$ to each color value c of a given (quantized) color space.

Our color model is computed from color occurrence statistics, which we determine from a set P of positive images (images containing the wanted object) and a set N of negative images (images not containing the wanted object). Usually in practice the set of positive images is significantly smaller than the set of negative images, since (almost) random images can be used as negative images while collecting positive images is often time-consuming. The main idea is to determine discriminative object colors, which appear significantly more often in positive images than in negative images, first. They serve as a seed in a second step, where we use an flood-fill algorithm to identify pixels of similar colors, which are spatially close to our seed pixels identified in the first step.

Our approach requires that two assumptions hold:

1. Objects of the desired object class occur in a limited number of different color schemes.
2. Background areas of positive images must be similar to the negative images and with respect to color more diverse than the wanted objects.

The first assumption is an obvious prerequisite since objects, which can assume many different color schemes, e.g. cars, cannot be represented by a color model. The second assumption must hold since we are searching for regularities among the positive images, which are not present among the negative images. Furthermore, the negative set must be reasonably large (i.e. larger than the positive set), since we need it to estimate a background model, which we want to be as general as possible. Since the Internet provides a huge number of publicly available random images and our model is tolerant to some noise, it is not difficult to provide large negative (background) sets for many object classes.

3.1 Identifying Distinctive Object Colors

We first determine which colors appear more often in positive images than we would expect given the number of their appearances in background images as these colors are likely to indicate an object of interest. Thus we compute the relative frequency with which each color c is present in the positive and negative image set, respectively. Let $n = |P|$ and $m = |N|$ be the numbers of images in the positive and negative set, respectively. Then the respective relative occurrence frequencies are given by

$$f_P(c) = \frac{1}{n} k_c \tag{1}$$

$$f_N(c) = \frac{1}{m} {k'}_c \tag{2}$$

where k_c and ${k'}_c$ are the absolute numbers of positive and negative images, respectively, in which c is present at a minimum of ϵ pixels for a very small value of ϵ (in our experiments we simply set $\epsilon = 0$).

We now discriminate two cases:

I. We consider all colors c as background colors (and thus $h'(c) = 0$) if they appear at least as frequent in negative images as in positive images, i.e. if $f_N(c) \geq f_P(c)$. These colors are very likely to be common background colors, since colors which appear in relatively fewer positive images than background images are obviously not indicative of the desired object if our assumptions hold.

II. If a color appears more frequent in positive than negative images (i.e. $f_P(c) > f_N(c)$), it is a candidate for being a distinctive object color for the object class of interest. However, it could still represent a background color, since our positive images also contain background regions. Thus we probabilistically model these colors to decide which colors should finally be considered as distinctive colors for the object class of interest.

Let $P(object|k_c)$ be the probability of observing color c in k_c out of n positive images given that c is a candidate for being a distinctive object color for the object class of interest. Let $P(\neg object|k_c)$ be the opposite probability. The latter probability is small, if it is unlikely that a color we observed k_c times among our n positive images is a background color. Therefore we compare the ratio of $P(object|k_c)$ and $P(\neg object|k_c)$ against a threshold T in order to decide if c is an object color:

$$\frac{P(object|k_c)}{P(\neg object|k_c)} > T' \tag{3}$$

Using Bayes' rule this equation can be written as

$$\frac{P(k_c|object)}{P(k_c|\neg object)} > \theta' \tag{4}$$

with

$$\theta' = T' \cdot \frac{1 - P(object)}{P(object)} \tag{5}$$

$P(object)$ is a constant value which we either have to determine empirically or intuitively. We discuss the selection of $P(object)$ in the section on our method of evaluation. Next we define the probability distributions $P(k_c|object)$ and $P(k_c|\neg object)$. Thus, we have to model the probability of observing a color c in k_c positive images whereas c is an object color or non-object color, respectively.

3.1.1 Probability Models

With $P(k_c|\neg object)$ we want to model the probability that a background color c appears k_c times in n positive images. Recall that we observed c at a rate of $f_N(c)$ among the background images. We assume that our positive images also contain common background, and N is a representative set of background images. We thus expect that colors appearing in N with relative frequency $f_N(c)$ appear among the positive images P with a frequency $f_P(c)$ close to $f_N(c)$.

In other words, we assume that N is a representative set of background images. Thus if we obtain a relative color occurrence frequency $f_N(c)$, we expect to observe a similar frequency among any set of n images containing background. Therefore $n \cdot f_N(c)$ is our expected value for k_c (because this implies that $f_P(c) = f_N(c)$). The variance we want to allow depends on the size of the positive set of images since the smaller the positive set, the more relative deviation can be explained by the disparity of the two image sets' respective sizes.

We model $P(k_c|\neg object)$ using a binomial distribution $b(k_c; n, f_N(c))$ with expectation $E[k_c|\neg object] = n \cdot f_N(c)$ and variance $Var[k_c|\neg object] = n \cdot f_N(c) \cdot (1 - f_N(c))$. Hence the probability $P(k_c|\neg object)$ of c being observed k_c times among the positive images and being a background color is given by

$$P(k_c|\neg object) = \binom{n}{k_c} \cdot f_N(c)^{k_c} \cdot (1 - f_N(c))^{n-k_c} \tag{6}$$

Note that the maximum value and variance of this distribution depends on the value of $f_N(c)$. For small and large values of $f_N(c)$ we obtain distributions with higher maximum probabilities and lower variances compared to distributions with $f_N(c)$ close to 0.5. Intuitively, we however want the probability value to only depend on the difference between $f_N(c)$ and $f_P(c)$ and not on the actual value of $f_N(c)$. Thus, we enforce the same shape for all distributions, i.e. we use the same maximum probability value and variance for each value of $f_N(c)$ by shifting the distribution for $f_N(c) = 0.5$ to the respective expected value $n \cdot f_N(c)$. $P(k_c|\neg object)$ is shown in Figure 2 for a positive set size of $n = 40$ and observed relative occurrence frequencies of $f_N(c) = 0.5$ and $f_N(c) = 0.75$.

As illustrated in Figure 2, $P(k_c|\neg object)$ is large if the relative number of positive images in which color c appears is similar to the relative number of negative images in which color c appears. Thus we obtain large values for $P(k_c|\neg object)$ for colors whose appearance in positive images can be explained by their frequent appearance in background.

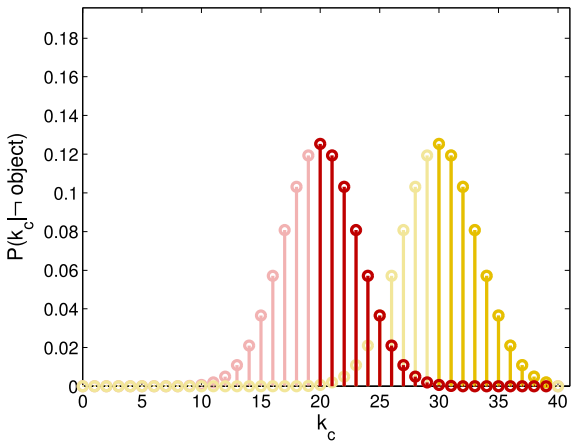


Figure 2: Visualization of our model for $P(k_c|\neg object)$ for $n = 40$ and example relative frequencies on N of $f_N(c) = 0.5$ (red) and $f_N(c) = 0.75$ (yellow). Note that both distributions have the same maximum probability and variance.

If, however, the relative number of positive images in which color c occurs is significantly larger than the number of negative images in which c occurs, the value of $P(k_c|\neg object)$ will decline. Intuitively this reflects the fact that color c is unlikely to appear in a significantly higher relative number among positive images than among background images if it is a background color. As a consequence we can deduce that color c is an object color if inequation 4 holds, since inequation 4 becomes true if $P(k_c|\neg object)$ is below $\theta \times P(k_c|object)$. Note that $P(k_c|\neg object)$ will also decline if $f_P(c)$ is significantly smaller than $f_N(c)$. However, this situation is already covered by case I, which already excludes colors which appear less often than expected. This is indicated by the brighter bars in Figure 2.

For $P(k_c|object)$ we do not have an estimation of the relative frequency from a larger set as is the case for $P(k_c|\neg object)$. Therefore we define a model for $P(k_c|object)$ which simply assigns high probabilities to all colors which appear in positive images and thus we use a constant value which only depends on n.

3.1.2 Initial Color Model

Combining the cases I. and II. yields the following decision rules that define the initial color model $h'(c)$:

if $f_N(c) \geq f_P(c)$:

$$h'(c) = 0 \tag{7}$$

if $f_N(c) < f_P(c)$:

$$h'(c) = \begin{cases} 1 & \text{if } \dfrac{P(k_c|object)}{P(k_c|\neg object)} > \theta' \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Thus each color c whose relative frequency of appearance is higher among negative images than among positive images is considered a non-object color since it appears less often in positive images than expected.

If the relative frequency is higher in positive images, we determine the ratio between the probabilities $P(k_c|object)$

and $P(k_c|\neg object)$ for the number k_c of positive images containing c. Recall that according to our definition $P(k_c|object)$ is constant for a given n.

Note that our initial decision rules may exclude colors which actually belong to the desired object if the same color appears in many background images or only on a few object instances. However, we are only interested in colors which are indicative of the object and at the same time do not regularly appear on random images which do not include the object. We try to reduce this effect by adding spatially and chromatically related colors as explained in the next section.

3.2 Identifying Object Color by Adding Spatially and Chromatically Related Colors

So far, our initial color model h' is only suitable for identifying pixels, which with high probability are located on objects of interest. However, the set of identified pixels will by no means cover the entire objects. Thus, the color model cannot yet be used to 'segment' the objects of interest as a standard color model would do. Thus, based on our seed pixels we need to identify, which other pixels might belong to wanted objects. These pixels should be neighbors of distinctive colors as identified by h'. Therefore, we will apply a flood-fill algorithm, which is seeded at pixels with object colors detected by h' in all images of P.

In detail, we first determine the set $\mathfrak{P}$ of all pixels p in all images of P whose color c_p is distinctive for the object class:

$$\mathfrak{P} = \{p|h'(c_p) = 1\} \tag{9}$$

Outliers are removed by applying a median filter to each image's binary map obtained by applying $h'(c)$. Now let $N(p)$ be the $4-$neighborhood of pixel p in its respective image. We define $N'(p)$ as the pixels p' from the neighborhood of p which do not deviate more than a given threshold θ_{c^i} from the color of p in any color channel i:

$$N'(p) = \{p' \in N(p)|\forall i : |c^i_{p'} - c^i_p| < \theta_{c^i}\} \tag{10}$$

where c^i_p is the value of the ith channel of color c_p. We conservatively define $\theta_{c^i} = 5$ for all color channels i in our experiments. Note that $N'(p)$ only includes pixels which are both, chromatically and spatialy close to pixels for which we can safely assume that they ae part of the wanted object. After determining the pixels of $N'(p)$ in each image of P we include them in $\mathfrak{P}$:

$$\mathfrak{P} = \mathfrak{P} \cup \left(\bigcup_{p \in \mathfrak{P}} N'(p) \right) \tag{11}$$

We repeat this procedure until $\mathfrak{P}$ does not grow anymore. Note that in further iterations, we keep the color c_p of the pixel p from the original set $\mathfrak{P}$ in equation 11. Thus, all pixels we add to $\mathfrak{P}$ may not deviate too much from colors found by our initial model. This way we prevent "leaking" into background across smooth object borders. As a result we obtain regions of similarly colored pixels, which we consider object pixels and thus positive examples. We use these regions to compute two color histograms, which represent $P(c|object)$ and $P(c|\neg object)$ as suggested in [7]. Since we consider pixels as positive or negative examples now (as opposed to whole images as in the previous subsection), we can simply use relative frequencies to define these conditional probabilities:

$$P(c|object) = \eta_P^{-1}|\{p \in \mathfrak{P}|c_p = c\}| \tag{12}$$

$$P(c|\neg object) = \eta_N^{-1}|\{p \in \mathfrak{P}_N|c_p = c\}| \tag{13}$$

where η_P and η_N are the sums of all histogram bin values of the respective histograms and $\mathfrak{P}_N$ are all pixels in all images of N.

Analogous to the previous subsection (see equations 3 and 4), we decide whether a color c belongs to the object by the following inequation:

$$\frac{P(c|object)}{P(c|\neg object)} > T \cdot \frac{1 - P(object)}{P(object)} \tag{14}$$

The prior $P(object)$ is the probability that any given pixel is part of the object and thus also depends on the size of the object in the image. Again, this value has to be determined empirically. In the evaluation section below we set $T = 1$ for creating the ROC curves.

Evaluating equation 14 for each color c yields our final object color model h:

$$h(c) = \begin{cases} 1 & \text{if } \frac{P(c|object)}{P(c|\neg object)} > \theta \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

with

$$\theta = T \cdot \frac{1 - P(object)}{P(object)} \tag{16}$$

Now we can decide whether a pixel is likely to belong to the wanted objects.

4. EXPERIMENTAL EVALUATION

In this section we first determine the color space parameters of our approach and then compare its accuracy to the baseline model.

4.1 Evaluation on Logos Dataset

For our first set of experiments we use the FlickrLogos-32 [9] dataset. This dataset consists of 32 sets of images containing brand logos. Each set contains 70 images showing an instance of a logo of the same brand. However, almost every image also features background. In some images the brand logo is not even featured very prominently. We pick 6 of the 32 classes for which we reckon that the brand logo has a fairly constant color scheme: "DHL", "Coca Cola", "Esso", "Aldi", "Pepsi", "Shell". Figure 3 shows a few sample images of the sets we have chosen.

We use the given partitioning of 10, 30, and 30 images for the training set, validation set, and test set, respectively. The FlickrLogos-32 dataset also provides a set of negative images, which do not contain any logos. We use $3,000$ negative images for training and $1,000$ negative images for testing. Note that we evaluate our approach pixel-wise on the test images, thus the number of test examples is reasonably large. For the following experiments we use the training and validation set combined to create the color model as described in the previous section and compute our results on the test sets. Thus, we have a positive training set per brand logo of $n = 40$ and a negative training set of $m = 3000$ images. The test sets consists of 30 logos per brand and 1000

Figure 3: Sample training images for the 6 logo classes we have used from the FlickrLogos-32 dataset [9].

negative images. For evaluating the quality of our color model we manually label the pixels, which belong to the logos, in the test images. Thus, for each pixel p we have a ground truth label $t(p) \in \{0, 1\}$.

4.1.1 Method of Evaluation

For all of the following experiments on the FlickrLogos-32 dataset we use the same method of evaluation:

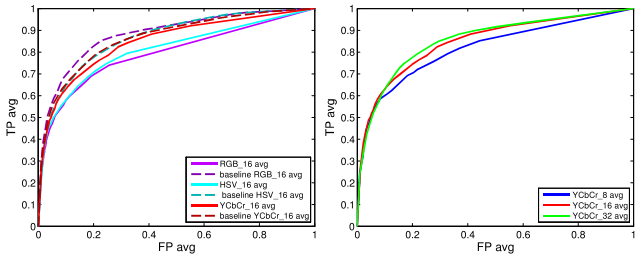
We measure the effectiveness of our approach by constructing a ROC curve over the threshold θ. We obtain the true positive (TP) rate for each tested threshold value from the pixel-wise annotations. Note that for each pixel of each negative image $p \in \mathfrak{P}_N$ we know that $t(p) = 0$ since the negative images never show a logo. Therefore we use the negative images to determine the false positive (FP) rate. We do not count FP among positive images since the logos are often found on surfaces of the same color as the logo (e.g. billboards, wrappings, or vehicles), which were annotated as not belonging to the logo. Thus, the annotation in positive images is ambiguous regarding negative pixels since the logos were simply annotated as narrowly as possible.

As we do not exploit any additional information, we have to empirically choose an appropriate value for θ' in formula 8. Since the magnitude of the conditional probability values depends on the "uniqueness" of the positive colors and is thus class-specific, it is difficult to select a universal threshold. We therefore set θ' (by adjusting T' in formula 5) to the 97%-quantile of all colors for the given class, which does not depend on the magnitude of the probability values. We use this relatively restrictive threshold, since we only want the most unique colors as seeds for the flood-fill algorithm explained in section 3.2.

4.1.2 Comparison of Color Spaces and Numbers of Bins

We first compare the performances of our model in three different color spaces: RGB, HSV, and YCbCr. For this experiment, we divide each color channel into 16 equally sized bins.

As a baseline approach, we also implement the approach of Jones and Rehg [7], which follows equations 12 to 16. However, for the baseline, the set of positive pixels $\mathfrak{P}$ are manually annotated pixels which are used to deduce relative frequencies $p(c|object)$ and $P(c|\neg object)$for positive and negative colors, respectively. Based on these frequencies, the Bayesian model of equation is used. Since the baseline approach is based on pixel-wise ground truth (except for some inaccuracies of the manual annotations), we consider it a reasonable reference.



(a) Color spaces (b) # bins per channel

Figure 4: Average ROC curves over all six logo classes over θ for (a) three different color spaces; (b) three different numbers of bins per channel in YCbCr color space.

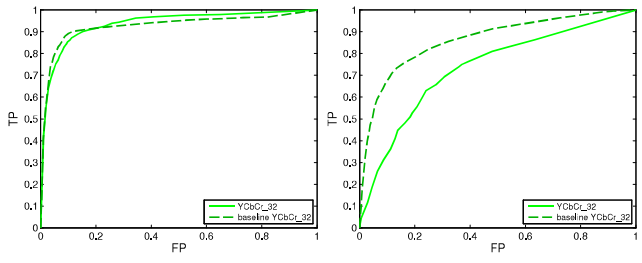
We compute an average ROC curve over all 6 logo classes by simply averaging the respective TP and FP values over all classes for each value of θ. The average ROC curves for RGB, HSV, and YcbCr color spaces are shown in Figure 4a as well as the corresponding ROC curves for the average results of the baseline approach.

The difference between our approach with the best color space and the baseline approaches for all color spaces is roughly the same. On average the YCbCr color space, however, yields the best results among the examined color spaces. Thus, we conduct all remaining experiments in the YCbCr color space.

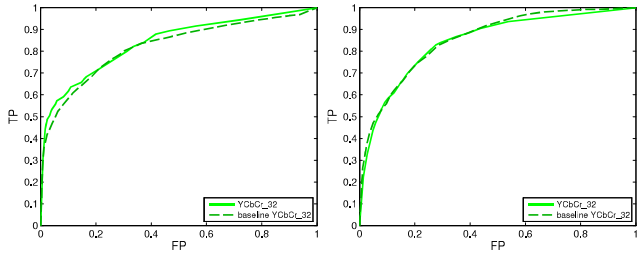
After determining the best color space, we evaluate different numbers of bins per channel. The resulting ROC curves are shown in Figure 4b. We compare results for 8, 16, and 32 bins. According to the ROC curves, the results are similar for all numbers of bins. Using 16 bins however yields a slightly better curve than using 8 bins and the result for 32 bins is arguably superior to the results for 16 bins. using 16 bins yields a slightly better curve than using 8 bins and the result for 32 bins is arguably superior to the results for 16 bins. Thus we use 32 bins per channel for the following experiments.

4.1.3 Comparison to Baseline of Each Class

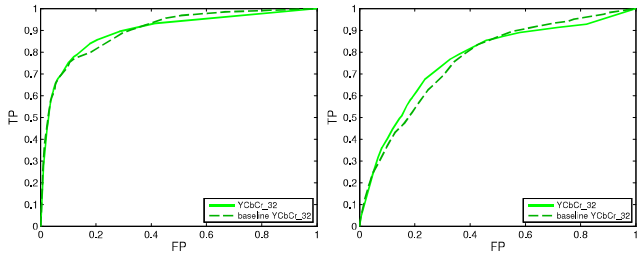
After determining the color space configuration, we examine the performance of our approach on each of our 6 logo classes. As a consequence of the previous experiments we use the YCbCr color space and 32 bins per color channel. We conduct the same experiment as described above. However, instead of averaging the ROC curves, we now determine the ROC curve for each class separately. The results are shown



(a) Class "DHL" (b) Class "Aldi"



(c) Class "Coca Cola" (d) Class "Pepsi"



(e) Class "Shell" (f) Class "Esso"

Figure 5: ROC curves over θ for our approach (solid line) compared to the baseline approach [7] (dashed line) in the YCbCr color space for 32 bins per channel. (a) - (f) show the results for each of our logo classes.

in Figure 5. Again we include the results of the baseline approach in each graph.

Overall, our approach is on par with the baseline for all classes except for "Aldi" despite using significantly less prior knowledge. Note that Aldi is the most difficult logo class since the logo consists of four different main colors and is backlit in some images which produces yet another color scheme. However, since we do not want to use any domain-specific knowledge, we do not adjust our parameters for individual classes

Interestingly, both the baseline and our approach work best for the class "DHL" which is due to the fact that the DHL logo usually contains bright yellow which is quite uncommon in background images. This is also true for the class "Shell" which however has more difficult training images which often include large background areas while the logo is very small. However, the baseline approach which is capable of omitting the background of positive images during training due to pixel-wise annotations still does not work significantly better.

Figure 6 shows two example test images for each logo class and the resulting image after removing all pixels which are classified as negative by the color model. For these examples we chose $p(object) = 0.1$ (and $T = 1$) in equation 4.1.2. Most of the non-logo pixels are removed correctly in the resulting images. However, pixels which have the same color as the logo obviously cannot be excluded by a color model. Since we set our parameters conservatively, we also miss some logo pixels, especially for the "Aldi" class.

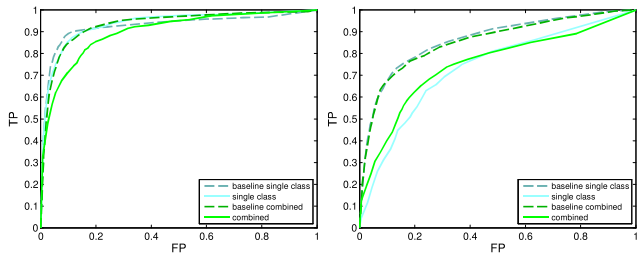
Figure 6: Example results for all 6 logo classes. Negative pixels were replaced by black pixels in the right image of each pair.

4.1.4 *Combined Object Classes*

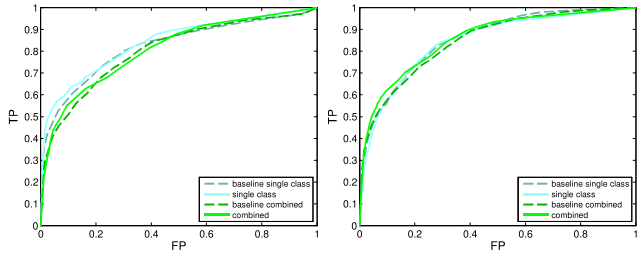
In this section we evaluate the effect of violating the assumption that the object has only one distinct color scheme. In these experiments we thus simulate classes for which this assumption is not true, i.e. we combine two objects into one class which is equivalent to the case of differently colored instances of one object class.

We thus create new positive image sets by combining the positive training and validation images of two different logos and learn a color model for the new set. We then apply this model to test images of each individual logo for evaluation. We only combine logos where at least one color is exclusive to one of the two classes. Figure 7 shows the results for each of our pair wise combinations on each of the individual test sets. For comparison we also plot the performances of the color model which was created only with training images of the tested class (i.e. the light blue curves in Figure 7 are identical to the curves in Figure 5). Thus we can determine the impact of mixing the training images in comparison to using only training images of a single class.

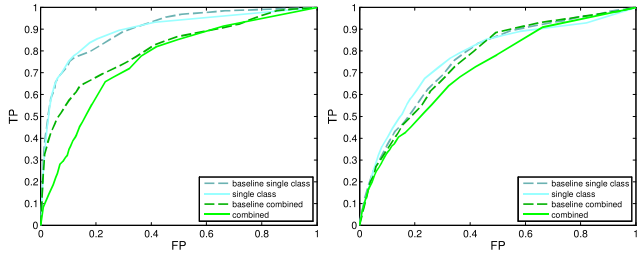
For the combination of "DHL" and "Aldi", the combined model's performance is inferior to the "DHL" model's performance on the "DHL" test set (see Figure 7a). On the "Aldi" test set however, the combined model yields a similar result as the individual "Aldi" model (see Figure 7b). Intuitively this is not surprising since the "DHL" logo consists of colors



(a) "DHL & Aldi" on "DHL" (b) "DHL & Aldi" on "Aldi"



(c) "Coca Cola & Pepsi" on "Coca Cola" (d) "Coca Cola & Pepsi" on "Pepsi"



(e) "Shell & Esso" on "Shell" (f) "Shell & Esso" on "Esso"

Figure 7: Results for combined logo classes. All results are for 32 bins in YCbCr color space.

which are similar to colors of the "Aldi" logo. Thus the results on the "Aldi" test set are only marginally affected. The "Aldi" logo, however, features a number of additional colors which are not present in the "DHL" logo and therefore the false positive rate on the "DHL" test images increases. Analogous observations can be made for the combined model of "Coca Cola" and "Pepsi".

In our final pair of classes we combine "Shell" and "Esso" which is the only combination where each logo has an exclusive color which is not part of the other logo. Consequently, for both test sets the combined model's performance is inferior to the respective individual model's performance. This is due to the fact that for each experiment the combined model contains a color which is not part of the respective logo and thus the false alarm ratio is increased.

In general, combining logos apparently results in an increasing false alarm rate when the combined model is used to detect the respective logo, which consists of fewer different colors. This is not surprising since adding images featuring another logo to the training set increases the number of different positive colors, which replace redundant positive colors of the individual logos.

In summary, we can conclude that the wanted object class does not necessarily have to feature the exact same color scheme in each training image, however, the number of differently colored instances should be reasonably small.

4.2 Evaluation on Flowers Dataset

The final dataset we use for evaluation is a subset of the Oxford Flowers dataset[8] which consists of 17 flower classes with 80 images each. As the previous experiment suggests, our approach is capable of creating models for differently colored objects to some extent. However, it is still not designed for objects which appear in a large number of different color schemes, so we only use the 15 classes form the Flower dataset for which the respective flowers does not appear in 4 or more differently colored variants (which is the case for "Iris" and "Crocus"). Note that some of the remaining classes still feature flowers, which occur in varying colors in different images, e.g. "Pansy" or "Fritillary". Since we feel that the flower dataset is similar to the logo dataset with regards to the nature of the wanted objects, we again use the YCbCr color space with 32 bins per channel for our experiments.

Some of the images are annotated pixel-wise, so we can use them to train the baseline approach and evaluate our method as described above. However, for some classes only very few images were annotated. We thus skip two classes which feature no or only very few annotated images. Overall, a total of 13 classes which are usable for our experiments remain.

For creating our model, we again use all annotated training and validation images of the respective class as positive images. We do not use images without annotations, since we cannot use them for creating the baseline. Thus the numbers of training images available for the individual sets vary from 20 to 53. Since the flower dataset does not contain a negative set, we simply use all images from all classes except the positive class as negative training images.

We conduct the same experiments as for the logos (see section 4.1.1 for details) to obtain TP and FP values for each class. However, since the flowers are annotated unambiguously, we can compute the FP on the background of the positive test images. Again, the number of test images varies between 8 and 20 due to the limited number of available annotations.

For clarity we do not show all individual ROC curves but the equal error rate for each flower class. The equal error rate corresponds to the point on the ROC curve for which the false positive rate is equal (or closest) to the false negative rate. Figure 8 shows the error rates compared to the error rates of the baseline approach. Also, one example result for the flower class "Tigerlily" is shown in Figure 1.

For most classes the error rate of our method is comparable to the error rate of the baseline. The most significant exception is class 2 for which our model apparently does not work. This is due to the fact that this class has almost no diversity in the background of the different images, which violates our assumption that the wanted object must be the most prominent common feature of the positive images. If parts of the background, however, are more similar across training images than the actual objects, our model prefers background colors over object colors.

Also for the other classes, the baseline is superior. However, the difference is not as significant, so our results are satisfying considering the fact that the baseline uses manual annotations.

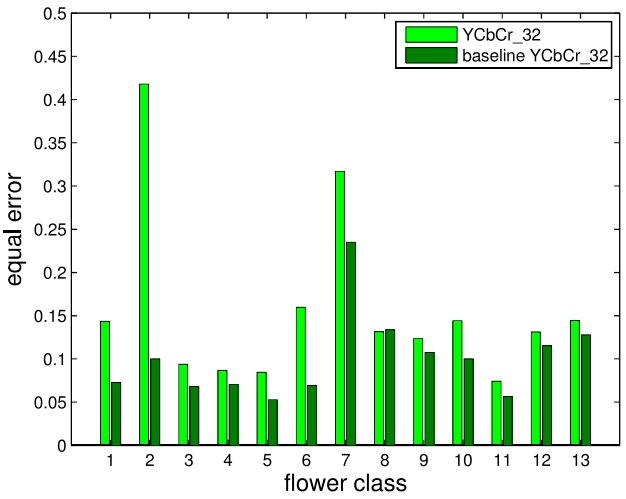


Figure 8: Equal error rates for the 13 classes used from the flower dataset.

Also note that many flower classes are very similar and almost identical regarding the color scheme. Therefore object colors in some cases appear in a considerable number of background images. Yet a little amount of noise among the negative images apparently does not affect the performance of our approach, which further eases providing negative sets in practice.

5. CONCLUSION AND FUTURE WORK

In this paper we have proposed a straightforward method to create a discriminative color model for a given object class from global image labels. The main advantage of our model over comparable approaches is that we do not require manual annotations of positive pixels.

In our experiments we have shown that our approach is roughly on par with or slightly below a baseline approach on the majority of our test classes. Given the fact that the baseline approach is based on pixel-wise annotation, this is a satisfactory result since our approach only requires positive and negative labels for the training images, which can be provided with significantly less human effort. We also showed that our approach still yields acceptable results in the face of two different positive objects simulating the case of objects, which occur in (a few) different colors. Still we found that the object should not appear in a large number of different colors.

Besides reducing human annotation effort, our method is particularly useful for situations where thorough manual annotation is undesirable due to the number or the nature of the positive images.

Currently our approach is a basic way to create a color model, which does not exploit any information except for global image labels, which still leaves room for improvements. Without requiring further previous knowledge, a more sophisticated model could for example be based on discretized n-tuples of colors obtained from connected groups of pixels instead of single color values from single pixels. Another possible extension is the exploitation of shape information we obtain from blobs of positive pixels, which might help identifying certain object classes.

Another interesting direction for future work is adjusting the method for more intra-class variance in order to create models for objects which may appear in a somewhat larger number of differently colored instances. Note that this would implicitly alleviate one of the strong assumptions which are required to hold for our method.

6. REFERENCES

[1] A. Albiol, L. Torres, and E. Delp. Optimum color spaces for skin detection. In *Proceedings of the International Conference on Image Processing, 2001.*, volume 1, pages 122 –124 vol.1, 2001.

[2] M. Benallal and J. Meunier. Real-time color segmentation of road signs. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2003.*, volume 3, pages 1823 – 1826 vol.3, may 2003.

[3] M.-C. Chi, J.-A. Jhu, and M.-J. Chen. H.263+ region-of-interest video coding with efficient skin-color extraction. In *Digest of Technical Papers of the International Conference on Consumer Electronics, 2006.*, pages 381 –382, jan. 2006.

[4] A. de la Escalera, L. Moreno, M. Salichs, and J. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6):848 –859, dec 1997.

[5] S. El Fkihi, M. Daoudi, and D. Aboutajdine. Skin and non-skin probability approximation based on discriminative tree distribution. In *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), 2009.*, pages 2377 –2380, nov. 2009.

[6] B. Jedynak, H. Zheng, M. Daoudi, and D. Barret. Maximum entropy models for skin detection. In *Proceedings of the Third Indian Conference on Computer Vision, Graphics and Image Processing*, pages 276–281, 2002.

[7] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, January 2002.

[8] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

[9] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol. Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 25:1–25:8, New York, NY, USA, 2011. ACM.

[10] H. A. Rowley, Y. Jing, and S. Baluja. Large scale image-based adult-content filtering. In *Proceedings of the 1st International Conference on Computer Vision Theory*, pages 290–296, 2006.

[11] J. Torresen, J. Bakke, and L. Sekanina. Efficient recognition of speed limit signs. In *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, 2004.*, pages 652 – 656, oct. 2004.

[12] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of the GraphiCon-2003*, pages 85–92, 2003.