Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

# Color-blob-based COSFIRE filters for object recognition☆

Baris Gecer[a,*], George Azzopardi[b,c], Nicolai Petkov[c]

[a]Imperial Computer Vision and Learning Lab (ICVL), Imperial College London, UK
[b]Intelligent Computer Systems, University of Malta, Malta
[c]Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, The Netherlands

ARTICLE INFO

*Article history:*
Received 25 September 2015
Received in revised form 9 September 2016
Accepted 4 October 2016
Available online 15 November 2016

*Keywords:*
Object recognition
Object representation
Color
Feature extraction
Trainable filters

ABSTRACT

Most object recognition methods rely on contour-defined features obtained by edge detection or region segmentation. They are not robust to diffuse region boundaries. Furthermore, such methods do not exploit region color information. We propose color-blob-based COSFIRE (Combination of Shifted Filter Responses) filters to be selective for combinations of diffuse circular regions (blobs) in specific mutual spatial arrangements. Such a filter combines the responses of a certain selection of Difference-of-Gaussians filters, essentially blob detectors, of different scales, in certain channels of a color space, and at certain relative positions to each other. Its parameters are determined/learned in an automatic configuration process that analyzes the properties of a given prototype object of interest. We use these filters to compute features that are effective for the recognition of the prototype objects. We form feature vectors that we use with an SVM classifier. We evaluate the proposed method on a traffic sign (GTSRB) and a butterfly data sets. For the GTSRB data set we achieve a recognition rate of 98.94%, which is slightly higher than human performance and for the butterfly data set we achieve 89.02%. The proposed color-blob-based COSFIRE filters are very effective and outperform the contour-based COSFIRE filters. A COSFIRE filter is trainable, it can be configured with a single prototype pattern and it does not require domain knowledge.

## 1. Introduction

Color discrimination is a powerful vision faculty of primates, who process color information in several areas of their visual system: from specialized cells in the retina [1] and lateral geniculate nucleus (LGN) [2] to cortical visual areas V1, V2 and V4 [3]. While the evolutionary explanation of the uniqueness of primates in exploiting color is still a debatable topic, it has been demonstrated that color facilitates the recognition of objects by man and computer [4–6].

Most frequently, automatic recognition of objects has been addressed by considering shape information, characterized by the spatial arrangement of contour parts [7,8]. If color is used at all, it is deployed for region segmentation and the contours of the regions are subsequently used for shape recognition. Such methods, however, are not able to distinguish objects which have very similar shapes and can only be distinguished when color is also taken into account, Fig. 1.

In the retina and the LGN there are neurons with center-surround receptive fields that respond to changes in color contrast. While there is neurophysiological evidence that some orientation-selective neurons in area V1 process color information [9,10], it is not clear how color is processed further to contribute to effective object recognition in visual cortex.

Yet, in order to illustrate the importance of color in human visual perception we refer to two different artistic techniques: pencil drawing vs. watercolor. In the former one, an artist creates shapes by drawing the contours of objects, while in the latter one shapes are formed by a set of color blobs with diffuse boundaries. This is in a way similar to the duality between contour- and region-based segmentation in computer vision. Fig. 2 shows examples of a line drawing and an image with diffuse boundaries both derived from the same painting.

A highly effective contour-based approach to object recognition in computer vision called COSFIRE (Combination of Shifted Filter Responses) has been introduced in [11] and has been found effective in various computer vision applications [12–16]. It uses information about the spatial arrangement of edges/contours of a given pattern of interest in gray-scale. A COSFIRE filter response is computed as the weighted geometric mean of the responses of certain orientation-selective (Gabor) filters at specific locations. While COSFIRE filters

☆ This paper has been recommended for acceptance by Seong-Whan Lee.
* Corresponding author.
*E-mail addresses:* b.gecer@imperial.ac.uk (B. Gecer), george.azzopardi@um.edu.mt (G. Azzopardi), n.petkov@rug.nl (N. Petkov).

http://dx.doi.org/10.1016/j.imavis.2016.10.006

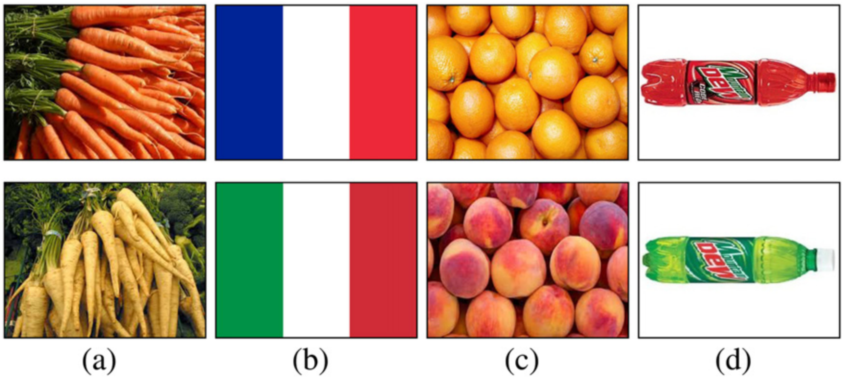


**Fig. 1.** Examples of objects that are difficult and in some cases even impossible to distinguish without color information: (a) carrot vs. parsley root, (b) the flags of France and Italy, (c) oranges vs. peaches, (d) red vs. green bottles of the same brand. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are very effective to detect and recognize objects that are characterized by contour-defined shape information, they lack the ability to distinguish objects that have similar shapes but different colors such as those shown in Fig. 1. This is because, they take into account only contour information and do not use color at all.

In this study we propose trainable COSFIRE filters that are selective for the geometrical arrangement of blobs in given patterns of interest. We use the responses of blob detectors to extract luminance and color information that characterize the interior of regions. The core idea of the proposed filters is to represent a pattern by the spatial arrangement of a set of luminance and color blobs. The mutual spatial arrangement of blobs describes the shape of an object, while the channels of given color space at which the blob detectors give strong responses carry information about the luminance and color distribution in the concerned pattern. Specifications about which DoGs to take, in which channels of color space to apply them, and in which positions to consider their responses are determined in an automatic configuration process applied on a prototype pattern.

This paper is organized as follows: in Section 2 we provide an overview of related work. In Section 3 we describe the color-blob-based COSFIRE filters that we propose. In Section 4 we provide experimental results on the traffic signs recognition benchmark (GTSRB) and the butterfly data sets. In Section 5 we discuss certain aspects of the proposed method and we draw conclusions in Section 6.

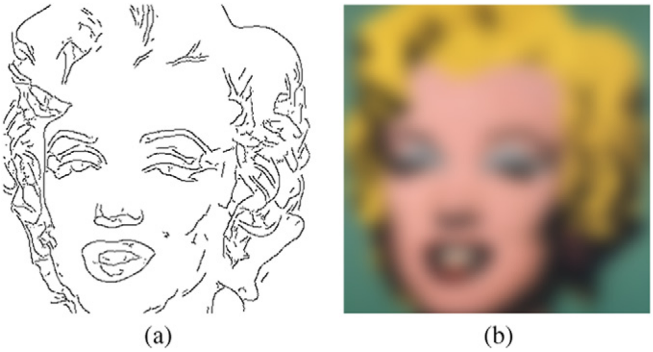


**Fig. 2.** (a) Line drawing obtained with the push-pull CORF algorithm [17] and (b) image with diffuse boundaries obtained from the same painting using Gaussian blurring. It is much easier and quicker to recognize the celebrity in the latter image as the yellow of the hair, the pale pink of the skin and the blue make-up are defining features of this emblematic image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 2. Related work

The robustness, efficiency and stability of color in human visual perception have motivated researchers to deploy color for object recognition [18]. There is a large body of work about object recognition based on combination of shape and color such as SEEMORE [19], C-SIFT [20], OpponentSIFT, RGB histogram, opponent histogram, transformed color histogram [21], color moments, color moment invariants [22] as well as various image retrieval applications [23–25]. Some invariance properties of color (related to viewing direction, highlights and illumination direction) have also been proposed [6,21,26].

Keypoint identification, description and matching are techniques which have been extensively used in the last decade for object recognition and image classification and retrieval [27]. In the computer vision literature, the term ‘keypoint’ is used to refer to the center of a (relatively small) image area with some characteristic distribution of gray level or color values. A certain keypoint descriptor (e.g. histogram of gradient values) is associated with such an area to represent this distribution. A keypoint may have some perceptual saliency but in general its selection as a keypoint is the result of the application of a computer algorithm rather than of the perceptual preference of a human observer. There is a great deal of work about various keypoint descriptors, such as HOG [28,29], SIFT [30], SURF [31], LBP [32], and BILD [33]. Descriptors that use color information include C-SIFT [20], OpponentSIFT, RGB histogram, opponent histogram, transformed color histogram [21], color moments and color moment invariants [22].

The use of existing keypoint descriptors or the development of new ones requires expert knowledge about the field in which they are applied. Moreover, it is usually common that a collection of such descriptors is required to index images [34]. As an alternative to such features, unsupervised feature learning approaches have been proposed [35–37]. They learn feature descriptors from training data without the need of having expert knowledge. In various benchmark data sets, pattern classification methods that use learned features have been demonstrated to outperform those that use algorithm-defined or ad hoc ones [38,39]. Feature learning approaches require lots of training data, which may not always be available or may be prohibitively expensive to obtain. This requirement also contrasts with the remarkable ability of the visual system of the brain that learns from few examples [40].

An object can be represented by a collection of keypoint descriptors extracted from it. The visual bag of words (BOW) [41,42] is one such approach which has gained particular popularity. It considers a histogram of keypoint occurrences in an image or a given bounding box, without using further information about their spatial arrangement. One way to consider spatial arrangement of keypoints is to apply spatial tiling followed by a spatial pyramid kernel [43]. In [44,45], the authors propose probabilistic approaches to model the shape of an object.

The method that we propose is a trainable filter approach, in that the selectivity of a filter (for shape and color) is learned from a prototype pattern that can either be specified by a user or automatically discovered by a system. Unlike the existing feature learning methods, a COSFIRE filter can be configured with one training example and is also tolerant to translation, rotation, scale and reflection transformations. The COSFIRE method is a filtering approach and besides recognition it possesses localization abilities up to a pixel precision. This is in contrast to other methods, such as BOW [41,42] that rely on sliding window techniques and can only indicate if a certain pattern is present or not anywhere in given bounding boxes.

## 3. Method

Fig. 3 (a) shows an image with round and oval (elliptical) diffuse blobs of different sizes and colors. We indicate with a dashed black circle an arrangement of three disks and an ellipse, which we show separately in Fig. 3 (b). We consider this arrangement as a prototype pattern of interest, and use it to automatically configure a trainable filter which will be selective for the same and similar patterns.

The filter that we propose is based on the COSFIRE approach that was introduced in [11]. In that study, the authors demonstrated how a COSFIRE filter can be configured to be selective for an arrangement of contour parts, essential elements of shape, and how it can be applied to grayscale images. Here, we propose a COSFIRE filter that is selective for a preferred arrangement of luminance and color blobs. Note that the original contour-based COSFIRE approach will fail on the patterns shown in Fig. 3 (a) because they do not exhibit clear contours.

The COSFIRE filter that we propose uses as input the responses of certain Difference-of-Gaussians (DoG)[1] filters applied to the luminance and color-opponent channels at certain positions with respect to its center. A DoG filter achieves strong responses to blobs that have (roughly) the same size and shape as the inner region of the DoG function [46].

We compute the response of a COSFIRE filter as the weighted geometric mean of the concerned responses of its constituent DoG filters. Consequently it responds strongly to a specific combination of color blobs. The arrangement of blobs is determined in automatic configuration step, which we explain below.

### 3.1. Configuration of a color-blob-based COSFIRE filter

The configuration phase consists of detecting blobs in the luminance (L*) channel and the color (a*, b*) channels using DoG filters and extracting the properties of the detected blobs together with their mutual spatial arrangement.

We transform a given prototype image to L*a*b* color opponent space as it approximates the human perception of color [47]. For the a* and b* channels a pixel value of 0 means that there is only luminance in that location; i.e. no color (hue) information. We denote by $I_\alpha(x,y)$ the value at location $(x,y)$ in the channel $\alpha$ ($\alpha \in \{L^*,a^*,b^*\}$).

#### 3.1.1. Detection of blobs

We denote by $DoG_{\sigma,\delta=+}(x,y)$ a center-on DoG function, with an excitatory (positive) central region and an inhibitory (negative) surround. The parameter $\sigma$ is the standard deviation of the outer Gaussian function:

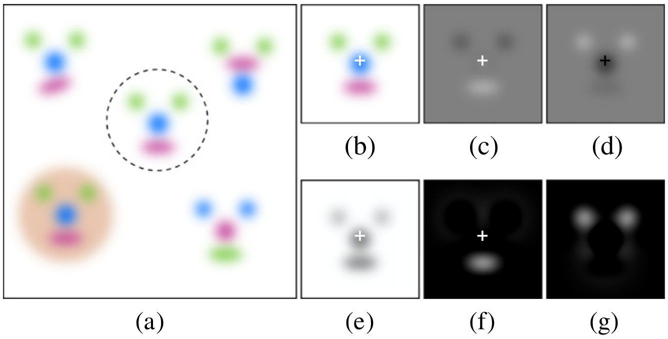


**Fig. 3.** (a) Synthetic image of color blobs with diffuse boundaries. The dashed black circle indicates a prototype pattern of interest (of size 800 × 800 pixels) that is shown separately in (b). The white cross marker indicates the center of the chosen prototype. (c–e) a*, b* and L* color channels of (b), respectively. (f) The response image of a center-on DoG filter ($\sigma = 84$) applied to the image in (c). (g) The response image of a center-on DoG filter ($\sigma = 81$) applied to the image in (d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$DoG_{\sigma,\delta=+}(x,y) \overset{\text{def}}{=} \frac{\exp\left(-\frac{x^2+y^2}{2(0.5\sigma)^2}\right)}{2\pi(0.5\sigma)^2} - \frac{\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^2} \qquad (1)$$

A center-off DoG function is denoted by $DoG_{\sigma,\delta=-}(x,y)$ and is defined as the negative of $DoG_{\sigma,\delta=+}(x,y)$. For more technical details about blob detection with DoG filters we refer the reader to [46,48].

This type of function is an accepted computational model of some cells in the lateral geniculate nucleus (LGN) of the brain of primates [49,50]. We set the standard deviation of the inner Gaussian function to $0.5\sigma$, a decision that is supported by the studies reported in [46,51].

For a given location $(x,y)$ and a channel intensity distribution $I_\alpha(x',y')$ the response $d_{\alpha,\sigma,\delta}(x,y)$ of a DoG filter with a kernel function $DoG_{\sigma,\delta}(x-x',y-y')$ is computed by convolution:

$$d_{\alpha,\sigma,\delta}(x,y) \overset{\text{def}}{=} |I_\alpha * DoG_{\sigma,\delta}|_+ \qquad (2)$$

where $|.|_+$ denotes half-wave rectification. A DoG filter achieves a strong response to a circular pattern with an area that fits the central region of the DoG function, and it does not respond to homogeneous areas. Fig. 3 (f–g) show two such examples. In [48], it was shown that the relationship between the parameter value of $\sigma$ of the DoG function and the radius $r$ of the optimal disk that elicits the maximum response is defined as $r \approx 2\gamma\sigma\sqrt{(-\log\gamma)/(1-\gamma^2)}$, where $\gamma$ stands for the ratio between the standard deviations of the inner and outer Gaussian functions. For $\gamma = 0.5$, the above equation is simplified to $r \approx 0.96\sigma$.

#### 3.1.2. Determining parameter values

The COSFIRE filter that we propose takes as input the responses of a set of DoG filters in different locations. The preferred polarity, standard deviations, locations and color channels at which we take their responses are automatically determined from a prototype example with the procedure explained below.

For each of the L*, a* and b* channels we apply a bank of DoG filters. For the image (800 × 800 pixels) in Fig. 3 (b), which we use as an example, we use 15 values of the parameter $\sigma$ ($\sigma \in \{60+3i \mid i = 0...14\}$) for both centre-on and centre-off DoGs so that they respond to blobs with radii roughly between 60 and 100. Then we take the responses that are greater than a fraction $t_1$ (in this example we use

[1] A Laplacian of Gaussian can also be used instead of a DoG.

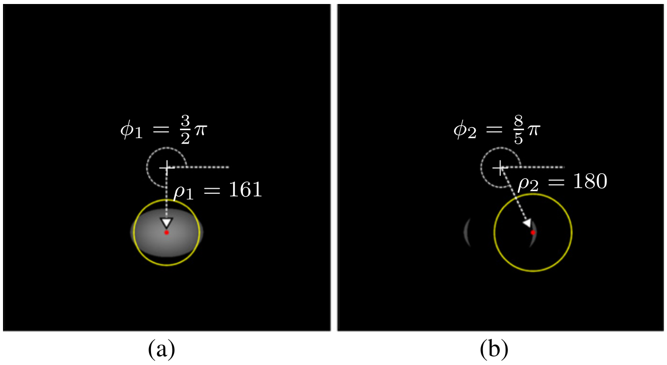
$t_1 = 0.8$ for L* channel and $t_1 = 0.5$ for a* and b* channels) of the maximum response across all coordinates $(x, y)$ and all $\sigma$ values for any combination of $\alpha$ and $\delta$. The elliptical shape in grayscale in Fig. 4 (a) consists of the maximum superposition of the thresholded responses of this bank of center-on DoG filters to the channel a* of the image in Fig. 3 (b).

Next, we apply an iterative procedure to determine which DoG filters and at which locations achieve strong responses in the L*, a* and b* channels. In the first iteration we take the location of the maximum DoG response across all values of $\sigma$ used. For the a* or b* channels, if the intensity value in that location is within a range of threshold values around 0 then we ignore it and consider the location of the next maximum DoG response. This is because a value of 0 represents no color information and values very close to 0 represent very low color information. The lower $l_t$ and upper $u_t$ bounds of this range of threshold values are determined adaptively for each image $I$: $l_t = -\max(5, 0.4|\min(I_\alpha)|)$ and $u_t = \max(5, 0.4|\max(I_\alpha)|)$, where the constant values 5 and 0.4 were determined experimentally on some training data. This restriction allows us to consider only regions where there is significant color information. For the L* channel, we use this criterion in the opposite way. If both values in the a* and b* channels at a specific location are outside the range specified by $l_t$ and $u_t$ then we ignore it and consider the location with the next maximum value. This means that we consider luminance information only at locations where there is no (or very low) color information.

For a selected location $(x', y')$, an example of which is indicated by the red spot in Fig. 4 (a), we form a tuple of five parameters $(\alpha_1, \sigma_1, \delta_1, \rho_1, \phi_1)$, where $\alpha_1$ represents the color channel, $\delta_1$ represents the polarity (centre-on or centre-off), $\sigma_1$ represents the standard deviation of the DoG filter that achieves the maximum response at the point $(x', y')$, and $(\rho_1, \phi_1)$ represents the distance and polar angle of that point with respect to the center of the prototype pattern.

The yellow circle (of radius $0.96\sigma_1$) around the red spot indicates the boundary of the inner region of the determined DoG function. We set to zero the responses of all DoG filters within that region, such that they will be disregarded in the subsequent iterations. Fig. 4 (b) shows the remaining DoG responses.

The red spot in Fig. 4 (b) indicates the location of the maximum response in the second iteration. If the preferred disk of the corresponding DoG filter around this location does not overlap more than 60% with the preferred disks of the DoG filters determined in the previous iterations, then we form a new tuple for this location, otherwise we consider the next maximum response. We repeat this procedure until there are no more valid locations to be considered.



(a) (b)

**Fig. 4.** First two iterations of the configuration procedure applied with center-on DoG filters to the a* channel of the image in Fig. 3 (d). The gray level of a pixel represents the maximum value superposition of the thresholded (at a fraction $t_1 = 0.5$ of the maximum response) responses of a bank of centre-on DoG filters with 15 values of $\sigma$ at that position. The yellow circle indicates the boundary of the inner support region of the DoG filter which gives maximum response at its center indicated by the red dot. The polar coordinates $(\rho_i, \phi_i)$ of each such point are determined with respect to the center of the prototype. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We denote by $S_f = \{(\alpha_i, \sigma_i, \delta_i, \rho_i, \phi_i) | i = 1 \ldots n_f\}$ the set of parameter values that describe the characteristics of the given prototype pattern $f$, where $n_f$ stands for the number of determined color and luminance blobs in the concerned pattern. For the pattern of interest shown in Fig. 3 (b), the configuration procedure described above results in eleven blobs with parameter values specified by the tuples in the following set:

$$S_f = \left\{ \begin{array}{c} \vdots \\ (\alpha_4 = a^*,\ \sigma_4 = 69,\ \delta_4 = -,\ \rho_4 = 211,\ \phi_4 = 3\pi/4), \\ \vdots \\ (\alpha_7 = b^*,\ \sigma_7 = 81,\ \delta_7 = +,\ \rho_7 = 204,\ \phi_7 = 3\pi/4), \\ \vdots \end{array} \right\}$$

The image in the bottom of Fig. 5 (a) illustrates the structure of the configured filter $S_f$. Every tuple in $S_f$ is illustrated by a diffuse color blob and a circle. The colors are determined from the polarity and color dimension. For channel L* we use white when $\delta = +$ and black when $\delta = -$, for channel a* we use magenta when $\delta = +$ and cyan when $\delta = -$ and for channel b* we use yellow when $\delta = +$ and blue when $\delta = -$. If two blobs overlap each other the color of the intersection region is the combined color of the two blobs. For instance, the cyan and yellow circles in the north west correspond to tuples 4 and 7, respectively. The green blob that they surround is the combination of yellow and cyan.

### 3.2. Application of color-blob-based COSFIRE filters

In the following we explain how we use the parameter values determined in the configuration stage to compute the response of the proposed color-blob-based COSFIRE filter.

#### 3.2.1. Shifting the responses of the selected DoG filters

For each tuple $i$ in the set $S_f$, we first compute the responses of a DoG filter with function kernel $DoG_{\sigma_i, \delta_i}$ to the corresponding color channel $\alpha_i$ of image $I$. Then, we shift the responses of the DoG filter by a distance $\rho_i$ in the direction opposite to $\phi_i$. Thus the concerned DoG filter responses, which are located at different positions meet at the support center of the COSFIRE filter at hand. We denote by $s_{\alpha_i, \sigma_i, \delta_i, \rho_i, \phi_i}(x, y)$ the shifted response of a DoG filter that is specified by the $i$-th tuple in the set $S_f$:

$$s_{\alpha_i, \sigma_i, \delta_i, \rho_i, \phi_i}(x, y) \overset{\text{def}}{=} d_{\alpha_i, \sigma_i, \delta_i}(x + \rho_i \cos\phi_i, y + \rho_i \sin\phi_i) \tag{3}$$

Fig. 5 illustrates the filtering and shifting operations of this COSFIRE filter applied to the image in Fig. 3 (a). We skip the blurring step of the original contour-based COSFIRE approach as we rely on the intrinsic blurring operations of the DoGs.

#### 3.2.2. Response of a color-blob-based COSFIRE filter

We define the response $r_{S_f}(x, y)$ of a color-blob-based COSFIRE filter as the weighted geometric mean of all the shifted and thresholded DoG filter responses $s_{\alpha_i, \sigma_i, \delta_i, \rho_i, \phi_i}(x, y)$ that correspond to the properties of the blobs described by $S_f$:

$$r_{S_f}(x, y) \overset{\text{def}}{=} \prod_{i=1}^{|S_f|} \left( s_{\alpha_i, \sigma_i, \delta_i, \rho_i, \phi_i}(x, y) + \epsilon \right)^{\omega_i / \sum_{i=1}^{|S_f|} \omega_i} \tag{4}$$

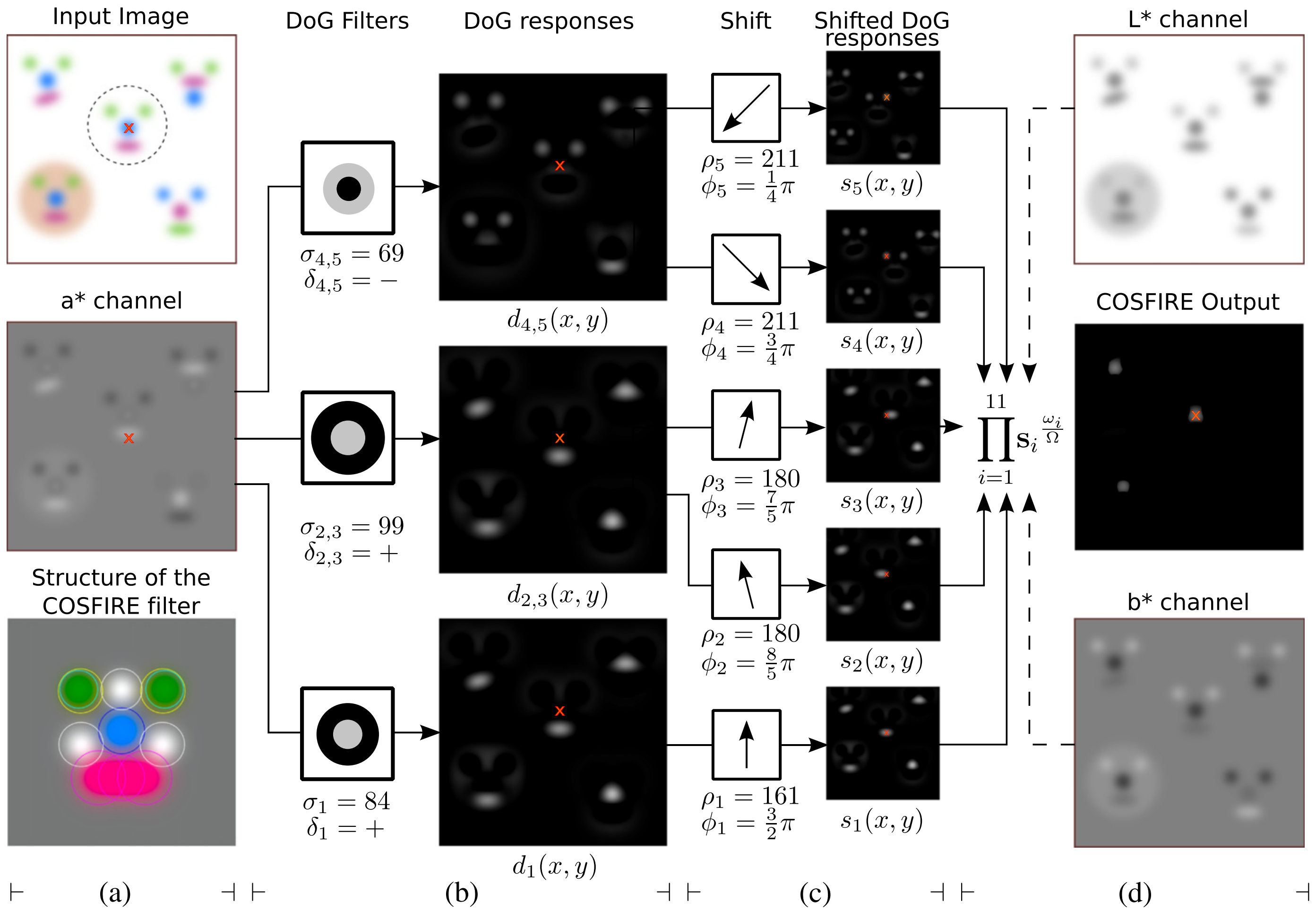


**Fig. 5.** (a) From top to bottom: input image, its a* channel and illustration of the structure of the COSFIRE filter that is configured by the pattern in Fig. 3 (b). The dashed black circle indicates the prototype pattern of interest. Each circle in the bottom image in (a) represents a tuple where its radius corresponds to the preferred standard deviation of a DoG filter and its color corresponds to the preferred polarity and color dimension. For clarity reasons this figure shows only the processing of the first five tuples that are applied to the a* color channel. (b) Tuples 2 and 3 share the responses of the same center-on DoG filter with $\sigma = 99$, and tuples 4 and 5 share the responses of the same center-off DoG filter with $\sigma = 69$. (c) The DoG responses are then shifted accordingly. The shifted DoG responses have the same size as the input image but here they are shown smaller in order to keep the figure concise. (d) Finally, the output of the COSFIRE filter (middle) is achieved by computing the weighted geometric mean of all the shifted DoG filter responses including those coming from the L* and b* channels. For conciseness sake, the processing of the L* and b* channels (shown at the top and bottom, respectively) is represented by the dashed arrows. The red cross marker indicates the location of the specified center point of interest. The three local maxima in the output of the COSFIRE filter correspond to the three similar arrangements in the input image. The symbols $d_i(x,y)$ and $s_i(x,y)$ are short notations for $d_{\alpha_i,\sigma_i,\delta_i}(x,y)$ and $s_{\alpha_i,\sigma_i,\delta_i,\rho_i,\phi_i}(x,y)$, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\omega_i = \exp\left(-\frac{\rho_i^2}{2\theta^2}\right)$ and the parameter $\epsilon$ refers to a very small value in order to avoid having zero $s$ terms.[2] In our experiments we use a value of the standard deviation $\theta$ that is computed as a function of the maximum distance $\rho$ value: $\theta = (-\rho_{\max}^2/(2\ln\kappa))^{1/2}$, where $\rho_{\max} = \max_{i\in\{1\ldots|S_f|\}}\{\rho_i\}$. We make this choice in order to achieve a maximum value $\omega = 1$ of the weights in the center (for $\rho = 0$), and a minimum value $\omega = \kappa$ in the periphery (for $\rho = \rho_{\max}$). For the example in Fig. 3 we use $\kappa = 0.5$.

Fig. 5 shows the output of the configured color-blob-based COSFIRE filter which is defined as the weighted geometric mean of eleven shifted images from the responses of seven DoG filters. Note that this filter responds at points where a pattern is present which is identical or similar to the prototype pattern $f$ at and around the selected point of interest, which was used in the configuration of the filter. In this example, the COSFIRE filter reacts strongly in a given point to a local pattern that contains a blue blob at the center, a pink horizontal ellipse at the bottom and two green blobs - one at the north west and the other at the north east - with the regions between these blobs having higher luminance than their surroundings.

### 3.3. COSFIRE-based descriptors

The use of one COSFIRE filter is suitable for the localization and recognition of one pattern of interest as shown in the example illustrated in Fig. 5. Various applications in computer vision involve the recognition of many different patterns of interest. For such applications we configure a collection of such filters from a given training set and use them to form descriptors. Below we propose two such methods and use them according to the type of the application at hand. We distinguish between two types of application: one with training examples of the patterns of interest isolated (with given bounding boxes) from the background and having the same orientation, and the other one where no bounding boxes are given and the patterns of interest may have different orientations.

[2] In practice we use $\epsilon = 10^{-8}$.

#### *3.3.1. Applications with isolated patterns of interest of same orientation*

The idea is to configure COSFIRE filters with a set of (parts of) prototypical examples from every class in the training set. Below we explain how we determine such prototypical examples.

First, we compute the mean training images $\bar{I}_{i \in \{1 \ldots k\}}$ for each of the $k$ classes in the training set and use them to configure $k$ COSFIRE filters that we denote by $C_{i \in \{1 \ldots k\}}$. Then, we consider every COSFIRE filter $C_i$ and compute the responses of the involved DoG filters for each training image in the corresponding class $i$. A training image in class $i$ is processed by the COSFIRE filter $C_i$ that has $n_f$ tuples, which correspond to $n_f$ contributing DoG filters. We use the responses of these DoG filters in the positions that are defined in the corresponding tuples by the polar coordinates $(\rho_j, \phi_j)$ with respect to the center of the image to form a feature vector of $n_f$ elements for each image. Finally, we apply $K$-means clustering to the feature vectors generated from all training images within the same category. The number of clusters $K$ is set to be the number of eigenvalues that account for 95% of the variability in the covariance matrix of the concerned vectors.

For each resulting cluster from a class $i$ we take the corresponding training images and configure a COSFIRE filter by using their mean image as a prototype. We call such filters whole-pattern-selective. Such filters are selective for the entire pattern of interest and they are effective when there are no occlusions. In order to account for occlusion we configure more filters from the same mean image but this time using only small parts. In practice, from each mean image we randomly select parts of radius that is one fourth of the radius of the full pattern and use them to configure COSFIRE filters to be selective for each part. We call such filters part-selective and we only consider those that involve more than ten tuples.

The above procedure results in a number of whole-pattern- and part-selective COSFIRE filters configured by a representative subset of training images in a given application.

The automatic selection of prototypical examples described above and the configuration of whole-pattern-selective COSFIRE filters is suitable in applications where all patterns of interest in the training set have the same orientation and they can be isolated from the background by given bounding boxes. We use this method for the GTSRB data set of traffic signs described in Section 4.1.

We apply all whole-pattern- and part-selective COSFIRE filters to every image. For the whole-pattern-selective filters, we weight their response maps by a 2D Gaussian function (with a standard deviation that is one sixth of the width and height of the input image) centered in the middle of the input image (as we expect the maximum response to be roughly at the center of the image), and use the maximum weighted response as a feature value. For the part-selective COSFIRE filters we use a $2 \times 2$ spatial tiling and take the maximum value of each filter in each tile. For $W$ number of whole-pattern-selective and $P$ number of part-selective COSFIRE filters this procedure results in a feature vector of $W + 4P$ elements.

#### *3.3.2. Applications with no bounding boxes and patterns of interest in different orientations*

On the other hand, in applications where the patterns of interest do not come with bounding boxes and/or have different orientations (e.g the butterfly data set) we cannot compute the mean class images and therefore we cannot use the procedure described above to determine a representative subset of prototypes. In such applications, depending on how large the training set is, we propose to use either a random subset or the full set of training images as our prototypes. Since in such applications we have no information about the location of the patterns of interest it is not possible to configure whole-pattern-selective COSFIRE filters. We only configure a number of part-selective COSFIRE filters with local patterns selected randomly from training images. If a COSFIRE filter does not result in more than ten tuples then we ignore it as its selectivity would be very low. The number of filters and the size of the local patterns are parameters given by the user or determined empirically on the training set.

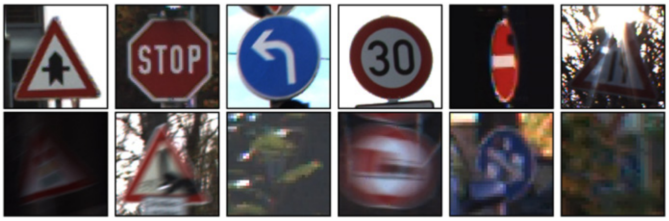


**Fig. 6.** Some examples taken from the GTSRB data set.

As to the descriptor, since we have no information about the orientations of the patterns of interest, spatial tiling is not suitable. We only take the maximum response of each part-selective filter irrespective of its position. For $P$ number of part-selective COSFIRE filters an image is represented by a $P$-element feature vector.

## 4. Experiments

### *4.1. Traffic sign recognition*

Many traffic related applications such as Driver Assistance Systems (DAS) [52], and self-driving cars [53] require a traffic sign recognition system which provides valuable information to drivers or to the automated driving systems.

We use the GTSRB[3] [54] benchmark data set of German traffic signs to evaluate the proposed approach. The data set consists of 39209 training and 12630 test images of complex scenes along with the ground truth bounding boxes that surround the involved traffic signs as well as the ground truth labels. There are 43 categories of traffic signs and they are represented in an unbalanced way. The sizes of the images vary from 15 × 15 to 250 × 250 pixels. Fig. 6 illustrates some examples from this data set. The pictures are taken by a camera that is mounted on a car while driving around. The captured traffic sign images contain various challenging problems, including blurring effects due to the motion of the car, tilting, changes in brightness due to different weather conditions, very small size of a traffic sign in an image, and occlusion.

Similar to other methods [55,56], we use the bounding boxes, which are given with the data set, to crop the traffic signs and resize them to 50 × 50 pixels. Then we convert all these cropped images to the L*a*b* color space and apply gamma correction to the luminance channel with $\gamma = 0.4$ in order to enhance dark images[4].

#### *4.1.1. Results*

We start by configuring COSFIRE filters[5] using the procedure described in Section 3.3.1. Since for this application the traffic signs come with the ground truth bounding boxes and they are resized to the same scale and given in the same upright orientation, we configure whole-selective COSFIRE filters and use spatial tiling for part-selective filters. The configuration results in 292 whole-pattern-selective and 1110 part-selective COSFIRE filters. We form feature vectors of length $(292 + 4 \times 1110 \;=)$ 4732 elements for the 39209 training images. Due to the selectivity of the filters, the resulting distributions of each of the 4732 features are skewed toward low values. We therefore log transform the feature vectors in order to make the distributions more

[3] http://benchmark.ini.rub.de/?section=gtsrb.

[4] The value of $\gamma$ is experimentally chosen by using the training data set.

[5] We use $\sigma \in \left\{ \frac{2^{5+0.3i}}{50} \quad | \quad i = 0 \ldots 16 \right\}$.

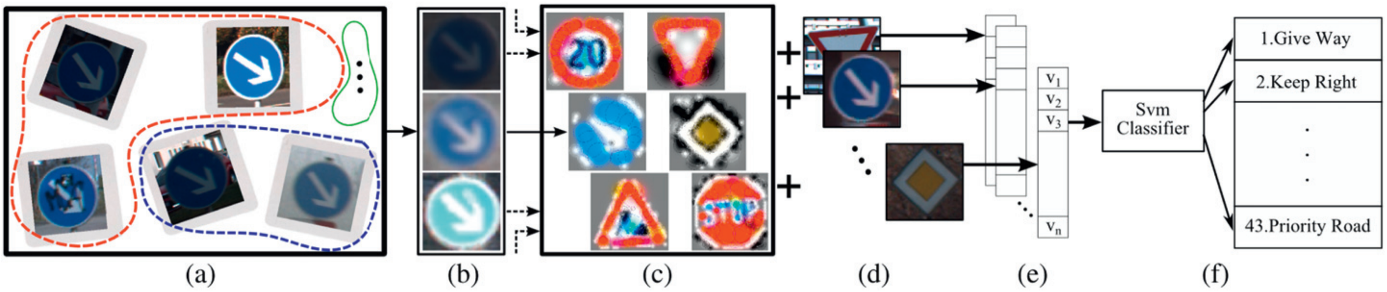


**Fig. 7.** Illustration of the configuration of COSFIRE filters and how they are used to form feature vectors to train an SVM classifier. (a) Clusters of images within one class of traffic signs. (b) Mean images of the identified sub groups of one class of traffic signs. (c) The reconstruction of a set of $m$ COSFIRE filters that are configured by the mean images determined from all groups of traffic signs. Every mean image is used to configure at most six COSFIRE filters; one that is selective for the whole pattern and at most five that are selective for randomly chosen parts. (d) Every image is processed (indicated by the '+' markers) by all $m$ COSFIRE filters. The response images of part-selective filters are divided into four (2 × 2) spatial tiles and the maximum response in each tile is used as a feature value $v_i$. For the whole-pattern-selective filters the weighted (by a Gaussian function) maximum is used as a feature value. (e) Feature vectors of training images are used to train (f) a 43-class SVM classifier.

**Table 1**
Comparison of our results on the GTSRB data set to those of human observers and to those of other methods. Results are reported for the full test set and also for each of the six groups as defined in other studies. For the computational methods, the top two results for each category are marked in bold face.

| Method | <br>All signs | <br>Speed limits | <br>Other prohibitions | <br>Derestriction | <br>Mandatory | <br>Danger | <br>Unique |
|---|---|---|---|---|---|---|---|
| Multi-column deep NN [55] | **99.46** | **99.47** | **99.93** | **99.72** | **99.89** | **99.07** | **99.22** |
| Proposed method | **98.94** | **99.09** | **99.93** | 93.33 | **99.72** | 97.78 | **99.80** |
| Human performance [58] | 98.84 | 97.63 | 99.93 | 98.89 | 99.72 | 98.67 | 100 |
| Multi-scale CNNs [59] | 98.31 | 98.61 | 99.87 | **94.44** | 97.18 | **98.03** | 98.63 |
| Random forests [56] | 96.14 | 95.95 | 99.13 | 87.50 | 99.27 | 92.08 | 98.73 |
| LDA on HOG 2 [58] | 95.68 | 95.37 | 96.80 | 85.83 | 97.18 | 93.73 | 98.63 |

symmetric. Finally, we use the feature vectors of all the 39209 training images to train a 43-class SVM[6] (one-against-one) with a linear kernel.

Similarly, we form the log-transformed feature vectors for the 12630 test images and present them to the SVM classifier. Fig. 7 illustrates the steps of this experiment. We achieve a recognition rate of 98.63%. Table 1 reports the recognition rates that we achieve for different groups of traffic signs as defined by other studies.

In order to understand better the effect of luminance and color we perform two more experiments. First, we consider only the luminance-based (L*) tuples and achieve a recognition rate of 98.66%. Second, we consider only the color-based (a* and b*) tuples and achieve a recognition rate of 77.66%. Finally, we perform another experiment by concatenating the feature vectors resulting from the luminance-based (L*) tuples to those resulting from all the (L*, a* and b*) tuples of the (292 + 1110 =) 1402 COSFIRE filters, and achieve a recognition rate of 98.94%.

The results show that while luminance based features are more effective than color based features for the application at hand, the addition of color information improves the results substantially. In the last experiment, the error rate is decreased by $\left(\frac{0.9894-0.9863}{1-0.9863} = \right)$ 22.63%. In Section 6 we discuss the design decision of this combination of features.

### 4.1.2. Experiment using contour-based COSFIRE

Here we compare the proposed color-blob-based COSFIRE filters with the original contour-based ones. We use the same prototype images and parts of images that were used in the above mentioned experiments, but this time converted to gray-scale, and configure 1402 contour-based COSFIRE filters[7]. With this approach we achieve a recognition rate of 90.68%, which is significantly lower than that achieved by the proposed color-blob-based COSFIRE filters.

## 4.2. Butterfly recognition

We perform an experiment on the butterfly data set that is composed of seven categories with a total of 182 training and 437 test images in color [60]. Every image contains one butterfly of varying size and orientation in a natural environment. This data set does not come with bounding boxes that localize the butterflies in the images. For this reason, here we use the procedure described in Section 3.3.2. We configure[8] eight part-selective COSFIRE filters with randomly selected local patterns (of 6 pixels radius) by using the L*a*b* color space of each training image and end up with 895 part-selective filters.

Similar to the original COSFIRE approach, the proposed filters can achieve tolerance to rotation, scale and reflection by the automatic manipulation of certain parameter values. We apply the configured COSFIRE filters using rotation and scale tolerance[9]. We represent

[6] We use the publicly available libsvm implementation [57].

[7] The best parameters were tuned experimentally on the training images.

[8] We use $\sigma \in \left\{\frac{2^{5+0.3i}}{50} \;\middle|\; i = 0\ldots10\right\}$.

[9] For rotation we use tolerance to eight orientations between −90° and 90° (in intervals of 22.5°) and for scale we use tolerance to six sizes that vary from a factor $2^{-1/2}$ to a multiple $2^{1/2}$ (in steps of 0.6) with respect to the orientation and size of the local prototype pattern that was used to configure the concerned filter.

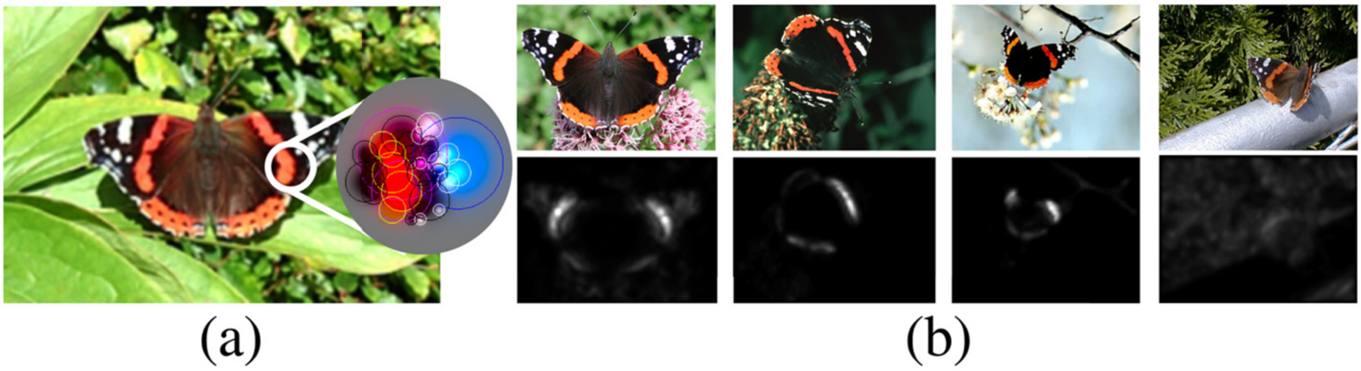


**Fig. 8.** (a) Configuration of a part-selective COSFIRE filter with the encircled pattern from a training butterfly image. The inset shows the structure of the resulting filter. (b) Butterfly test images (top) and the corresponding response maps (bottom) achieved by the configured filter with rotation and scale tolerance.

an image with a 895-element feature vector where the elements are the log-transformed maximum responses of the 895 filters to the given image. Fig. 8 shows an example of the configuration and application of one COSFIRE filter with rotation and scale tolerance.

We achieve a recognition rate of 89.02%. When we apply the filters with only the luminance-based tuples we achieve a recognition rate of 82.83%, and by using only the color-based tuples (a* and b*) we achieve 66.36%. By concatenating the feature vectors generated by the filters using all tuples to those of the same filters but using only the luminance-based tuples we achieve a recognition rate of 88.78%. Table 2 reports the recognition rates that we achieve for different categories of butterflies and Table 3 shows the confusion matrix of our method on the seven categories for the test set.

## 5. Discussion

The results that we achieve on the GTSRB data set are slightly better than those of human observers and rank second among all methods applied to this data set. The method that achieves better results is a multi-column deep neural network (NN) [55], which uses extensive preprocessing and sophisticated learning and classification techniques. We only use gamma correction as a preprocessing step and a basic SVM with a linear kernel. Our contribution is not in the sophistication of the preprocessing and/or classifier, but in the simplicity, yet highly effective, feature detectors, which can be configured by single training examples.

The fact that several traffic sign categories have the same color arrangement (e.g. speed limit signs have the same red circular boundary), color information by itself is not sufficiently effective to discriminate such traffic signs. Hence, it is no surprise that with only color information we achieve lower performance than when we consider only luminance information. This is also in line with the findings of [59]. In order to give more importance to luminance we concatenate vectors coming from L*-tuples only with vectors coming from all tuples. This type of description led to the best results for the GTSRB data set.

The trainable character of a COSFIRE filter makes the proposed approach applicable to any other object classification problem. In fact, the experiments on the butterfly data set demonstrate this versatility. The result of 89.02% that we obtain for the butterfly data set is comparable to the recognition rate of 90.61% reported in [61]. In practice, our method recognizes seven butterflies less (out of 437) than the referred method. To our knowledge, the latter result is the best ever reported in the literature and it was obtained by a method that relies on a learning step to determine the class probabilities of the involved features. Notable is the fact that our result is achieved without such a learning step and with no fine tuning. For instance, we do not analyze the randomly selected local patterns that we use to configure filters. Some patterns may be selected from the background (rather than from the butterflies) and

**Table 2**
Comparison of our results to other methods for the butterfly data set. Results are reported for the full test set and also for each of the seven groups as defined in other studies. The top two results for each category are marked in bold face. Number of correctly classified samples for each category are marked in bold face.

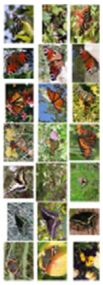
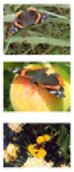
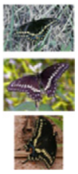
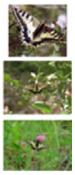
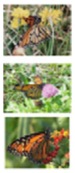
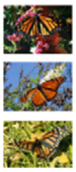
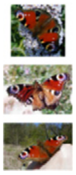
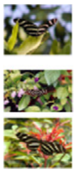
| |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| Method | All butterflies | Admiral | Swallowtail | Machaon | Monarch 1 | Monarch 2 | Peacock | Zebra |
| [61] | **90.61** | **92.9** | **100** | 91.2 | **85.4** | 81.0 | 95.4 | 89.2 |
| [60] | **90.4** | 87.1 | 75.0 | **96.5** | 72.9 | **91.4** | **100** | **89.2** |
| [62] | 89.4 | 91 | 81 | **95** | 67 | 84 | **98** | **92** |
| Ours | 89.02 | **95.29** | **93.75** | 85.96 | **77.08** | **84.48** | 93.51 | 87.69 |

**Table 3**
Confusion matrix of the proposed method obtained from the butterfly test set. Number of correctly classified samples for each category are marked in bold face.

| | | Predictions | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Classes | Admiral | Swallowtail | Machaon | Monarch 1 | Monarch 2 | Peacock | Zebra | Recall (%) |
| Ground truth | Admiral | **81** | 0 | 1 | 2 | 1 | 0 | 0 | 95.3 |
| | Swallowtail | 0 | **15** | 1 | 0 | 0 | 0 | 0 | 93.8 |
| | Machaon | 0 | 5 | **49** | 0 | 2 | 1 | 0 | 86.0 |
| | Monarch 1 | 0 | 0 | 5 | **37** | 1 | 5 | 0 | 77.1 |
| | Monarch 2 | 2 | 0 | 2 | 2 | **49** | 3 | 0 | 84.5 |
| | Peacock | 1 | 1 | 2 | 1 | 1 | **101** | 1 | 93.5 |
| | Zebra | 1 | 2 | 3 | 1 | 1 | 0 | **57** | 87.7 |
| Precision (%) | | 92.3 | 65.2 | 77.8 | 86.0 | 89.1 | 91.8 | 98.3 | |

therefore they do not carry any discriminative power. In future, we will investigate learning techniques, such as GMLVQ [63], to determine the most effective COSFIRE filters for a given application and to determine the corresponding class probabilities of their relevances.

In the experiments we chose parameter values empirically on a small validation set and used the same values for both applications (except those related with scale). Therefore, fine tuning of those parameter values is not necessary. The value of the threshold parameter $t_1$ determines the minimum accepted strength of the DoG filter responses to the detected blobs. The set of $\sigma$ values used during configuration stage is important for selectivity of blob sizes, which are defined on a logarithmic scale in the experiments. A COSFIRE is sensitive to the general surface structure of an object when large values of $\sigma$ are used. It becomes sensitive to details with smaller $\sigma$ values. The $\sigma$ values of DoGs can be adjusted according to the application.

The discrimination ability of a COSFIRE filter depends on the number of tuples used. The more tuples such a filter has the more selective it becomes. In order to avoid filters with limited discrimination ability, we discard any filters that are configured with less than 10 tuples, a value which was found empirically on a validation set.

The configuration of a COSFIRE filter requires only one prototype example. This is very useful especially in applications where there are few training examples. Moreover, the selectivity of a COSFIRE filter can be easily interpretable. For instance, in Fig. 9 we illustrate the structure of a color-blob-based COSFIRE filter that was configured with the iconic painting of Marilyn Monroe. Such a reconstruction gives a clear indication of what the selectivity of the filter is. This filter gives high responses to both the original painting and to the one with diffuse region boundaries shown in Fig. 2b.

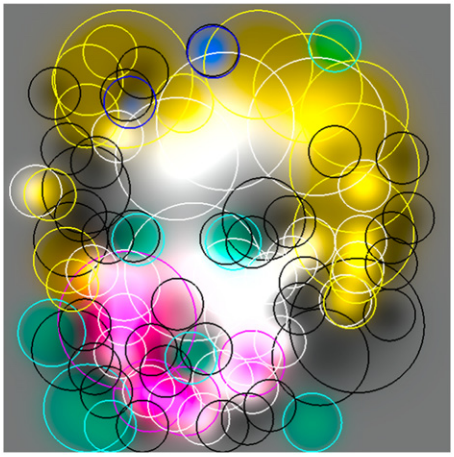


**Fig. 9.** The structure of a color-blob-based COSFIRE filter that is configured by the image in Fig. 2b.

The application of the proposed COSFIRE filters is highly parallelizable as the computations of the DoG filter responses that are used as input are independent of each other. In our experiments, however, we used a sequential Matlab implementation on a notebook with an Intel i7-2630QM 2.00 GHz CPU. For a given application the configured set of color-blob-based COSFIRE filters take input from the same DoG filter bank for each color channel. For instance, for the GTSRB data set we used a bank of 34 DoG filters (17 center-on and 17 center-off) and 3 color channels. Therefore, irrespective of the number of color-blob-based COSFIRE filters the worst case scenario was to convolve a 50 × 50 pixel image with $(34 \times 3 =)$ 102 DoG filters. In practice, the application of an average color-blob-based COSFIRE filter with 33 tuples to a traffic sign image takes 0.07 s, while the computation of the descriptor with $(292 + 1110 =)$ 1402 filters takes 17.64 s. The computational time, therefore, does not increase linearly with the number of filters. In future, we will implement the proposed method in a parallel algorithm, which we expect to become suitable for real-time applications.

The COSFIRE filters that we propose differ from the original COSFIRE approach in three aspects. First, besides shape it includes color information by analyzing all channels in a given color space. Second, we use an iterative configuration procedure, which ensures that all regions with significant color or luminance information are included in the filter. This is in contrast to the discretized configuration step of the original COSFIRE approach, which is based on a given number of concentric circles. Third, it uses blob detectors as afferent inputs, rather than orientation-selective filters. Blob detectors are more suitable for objects built from homogeneous parts and for objects with diffuse boundaries.

The proposed COSFIRE filters can also be used to locate (detect) objects of interest in an image. This is demonstrated in Fig. 5 (d). In future, we will perform quantitative analysis of this aspect on the same GTSRB data set and other suitable ones.

## 6. Conclusions

We present a novel approach that combines luminance, color and spatial arrangement information in trainable COSFIRE filters. We use their responses to form feature vectors whose effectiveness for object recognition is demonstrated on the challenging GTSRB traffic sign and butterfly data sets. For the GTSRB data set we achieve a recognition rate of 98.94%, which is slightly higher than human performance and for the butterfly data set we achieve 89.02%.

In contrast to most object recognition methods that rely on contour-defined features obtained by edge detection or segmentation our method is robust to diffuse region boundaries and exploits also region color information.

COSFIRE filters are trainable, in that they do not require domain knowledge, but they are automatically configured by any given single prototype image. Thus, they are suitable to various object recognition problems.

## References

[1] B.R. Conway, S. Chatterjee, G.D. Field, G.D. Horwitz, E.N. Johnson, K. Koida, K. Mancuso, Advances in color science: from retina to behavior, J. Neurosci. 30 (45) (2010) 14955–14963.

[2] M.S. Livingstone, D.H. Hubel, Psychophysical evidence for separate channels for the perception of form, color, movement, and depth, J. Neurosci. 7 (11) (1987) 3416–3468.

[3] S. Zeki, J. Watson, C. Lueck, K.J. Friston, C. Kennard, R. Frackowiak, A direct demonstration of functional specialization in human visual cortex, J. Neurosci. 11 (3) (1991) 641–649.

[4] J.W. Tanaka, L.M. Presnell, Color diagnosticity in object recognition, Percept. Psychophys. 61 (6) (1999) 1140–1153.

[5] D. Mapelli, M. Behrmann, The role of color in object recognition: evidence from visual agnosia, Neurocase 3 (4) (1997) 237–247.

[6] T. Gevers, A.W. Smeulders, Color-based object recognition, Pattern Recogn. 32 (3) (1999) 453–464.

[7] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.

[8] C. Grigorescu, N. Petkov, Distance sets for shape filters and shape recognition, IEEE Trans. Image Process. 12 (10) (2003) 1274–1286.

[9] S.G. Solomon, P. Lennie, The machinery of colour vision, Nat. Rev. Neurosci. 8 (4) (2007) 276–286.

[10] K. Yang, S. Gao, C. Li, Y. Li, Efficient color boundary detection with color-opponent mechanisms, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013. pp. 2810–2817.

[11] G. Azzopardi, N. Petkov, Trainable COSFIRE filters for keypoint detection and pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 490–503.

[12] J. Guo, C. Shi, G. Azzopardi, N. Petkov, Inhibition-augmented trainable COSFIRE filters for keypoint detection and object recognition, Mach. Vis. Appl. (2016) 1–15.

[13] G. Azzopardi, N. Petkov, Ventral-stream-like shape representation: from pixel intensity values to trainable object-selective COSFIRE models, Front. Comput. Neurosci. 8 (2014).

[14] G. Azzopardi, N. Strisciuglio, M. Vento, N. Petkov, Trainable COSFIRE filters for vessel delineation with application to retinal images, Med. Image Anal. 19 (1) (2015) 46–57.

[15] G. Azzopardi, L. Fernandez Robles, E. Alegre, N. Petkov, Increased Generalization Capability of Trainable COSFIRE Filters with Application to Machine Vision, IEEE, 2016.

[16] G. Azzopardi, A. Greco, M. Vento, Gender Recognition from Face Images with Trainable COSFIRE Filters, IEEE, 2016.

[17] G. Azzopardi, A. Rodríguez-Sánchez, J. Piater, N. Petkov, A push-pull CORF model of a simple cell with antiphase inhibition improves SNR and contour detection, PLoS ONE 9 (7) (2014) 1–13.

[18] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vis. 7 (1) (1991) 11–32.

[19] B.W. Mel, SEEMORE: combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition, Neural Comput. 9 (4) (1997) 777–804.

[20] A.E. Abdel-Hakim, A.A. Farag, CSIFT: a SIFT descriptor with color invariant characteristics, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, IEEE, 2006. pp. 1978–1983.

[21] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.

[22] F. Mindru, T. Tuytelaars, L.V. Gool, T. Moons, Moment invariants for recognition under changing viewpoint and illumination, Comput. Vis. Image Underst. 94 (1) (2004) 3–27.

[23] A.K. Jain, A. Vailaya, Image retrieval using color and shape, Pattern Recogn. 29 (8) (1996) 1233–1244.

[24] H. Yu, M. Li, H.-J. Zhang, J. Feng, Color texture moments for content-based image retrieval, Image Processing. 2002. Proceedings. 2002 International Conference on, Vol. 3, IEEE, 2002. pp. 929–932.

[25] K. Bunte, M. Biehl, M.F. Jonkman, N. Petkov, Learning effective color features for content based image retrieval in dermatology, Pattern Recogn. 44 (9) (2011) 1892–1902.

[26] A. Diplaros, T. Gevers, I. Patras, Combining color and shape information for illumination-viewpoint invariant object recognition, IEEE Trans. Image Process. 15 (1) (2006) 1–11.

[27] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.

[28] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005. pp. 886–893.

[29] A. Bosch, A. Zisserman, X. Munoz, Representing shape with a spatial pyramid kernel, Proceedings of the 6th ACM International Conference on Image and Video Retrieval, ACM, 2007. pp. 401–408.

[30] D.G. Lowe, Object recognition from local scale-invariant features, Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Vol. 2, IEEE, 1999. pp. 1150–1157.

[31] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comput. Vis. Image Underst. 110 (3) (2008) 346–359.

[32] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recogn. 29 (1) (1996) 51–59.

[33] Y. Zhang, T. Tian, J. Tian, J. Gong, D. Ming, A novel biologically inspired local feature descriptor, Biol. Cybern. 108 (3) (2014) 275–290.

[34] V. Mondéjar-Guerra, R. Muñoz-Salinas, M.J. Marín-Jiménez, A. Carmona-Poyato, R. Medina-Carnicer, Keypoint descriptor fusion with Dempster-Shafer theory, Int. J. Approx. Reason. 60 (2015) 57–70.

[35] G. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.

[36] M. Ranzato, F.J. Huang, Y.-L. Boureau, Y. LeCun, Unsupervised learning of invariant feature hierarchies with applications to object recognition, Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007. pp. 1–8.

[37] Q.V. Le, Building high-level features using large scale unsupervised learning, Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013. pp. 8595–8598.

[38] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 2012. pp. 1097–1105.

[39] J. Schmidhuber, Deep Learning in Neural Networks: An Overview, arXiv preprint arXiv:1404.7828, 2014.

[40] S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system, nature 381 (6582) (1996) 520–522.

[41] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 2, Ieee, 2005. pp. 524–531.

[42] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, Int. J. Comput. Vis. 79 (3) (2008) 299–318.

[43] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, IEEE, 2006. pp. 2169–2178.

[44] B. Leibe, A. Leonardis, B. Schiele, Robust object detection with interleaved categorization and segmentation, Int. J. Comput. Vis. 77 (1-3) (2008) 259–289.

[45] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 594–611.

[46] P. Kruizinga, N. Petkov, Computational model of dot-pattern selective cells, Biol. Cybern. 83 (4) (2000) 313–325.

[47] M. Tkalcic, J.F. Tasic, Colour Spaces: Perceptual, Historical and Applicational Background, IEEE, 2003.

[48] N. Petkov, W.T. Visser, Modifications of center-surround, spot detection and dot-pattern selective operators, Institute of Mathematics and Computing Science, University of Groningen, The Netherlands, 2005.

[49] G.E. Irvin, V.A. Casagrande, T.T. Norton, Center/surround relationships of magnocellular, parvocellular, and koniocellular relay cells in primate lateral geniculate nucleus, Vis. Neurosci. 10 (02) (1993) 363–373.

[50] X. Xu, A. Bonds, V.A. Casagrande, Modeling receptive-field structure of koniocellular, magnocellular, and parvocellular LGN cells in the owl monkey (*Aotus trivigatus*), Vis. Neurosci. 19 (2002) 703–711.

[51] F. Kingdom, M. McCourt, B. Blakeslee, In defence of "lateral inhibition" as the underlying cause of induced brightness phenomena: a reply to Spehar, Gilchrist and Arend, Vis. Res. 37 (8) (1997) 1039–1044.

[52] A. Laika, W. Stechele, A review of different object recognition methods for the application in driver assistance systems, Image Analysis for Multimedia Interactive Services, 2007. WIAMIS'07. Eighth International Workshop on, IEEE, 2007. pp. 10.

[53] M. Bertozzi, A. Broggi, A. Fascioli, Vision-based intelligent vehicles: state of the art and perspectives, Robot. Auton. Syst. 32 (1) (2000) 1–16.

[54] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The German traffic sign recognition benchmark: a multi-class classification competition, IEEE International Joint Conference on Neural Networks, 2011. pp. 1453–1460.

[55] D. Cireşan, U. Meier, J. Masci, J. Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Netw. 32 (2012) 333–338.

[56] F. Zaklouta, B. Stanciulescu, O. Hamdoun, Traffic sign classification using K-d trees and random forests, Neural Networks (IJCNN), The 2011 International Joint Conference on, IEEE, 2011. pp. 2151–2155.

[57] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.

[58] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition, Neural Netw. 32 (2012) 323–332.

[59] P. Sermanet, Y. LeCun, Traffic sign recognition with multi-scale convolutional networks, Neural Networks (IJCNN), The 2011 International Joint Conference on, IEEE, 2011. pp. 2809–2813.

[60] S. Lazebnik, C. Schmid, J. Ponce, Semi-local affine parts for object recognition, British Machine Vision Conference (BMVC'04), The British Machine Vision Association (BMVA), 2004. pp. 779–788.

[61] D. Larlus, F. Jurie, Latent mixture vocabularies for object categorization and segmentation, Image Vis. Comput. 27 (5) (2009) 523–534.

[62] F. Scalzo, J.H. Piater, Adaptive patch features for object class recognition with learned hierarchical models, 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007. pp. 1–8.

[63] P. Schneider, M. Biehl, B. Hammer, Adaptive relevance matrices in learning vector quantization, Neural Comput. 21 (12) (2009) 3532–3561.