

# Bayesian Classification of Image Structures

D. Goswami<sup>1</sup>, S. Kalkan<sup>2</sup>, and N. Krüger<sup>3</sup>

<sup>1</sup> Dept. of Computer Science, Indian School of Mines University, India  
dibyendusekharg@gmail.com

<sup>2</sup> BCCN, University of Göttingen, Germany  
sinan@bccn-goettingen.de

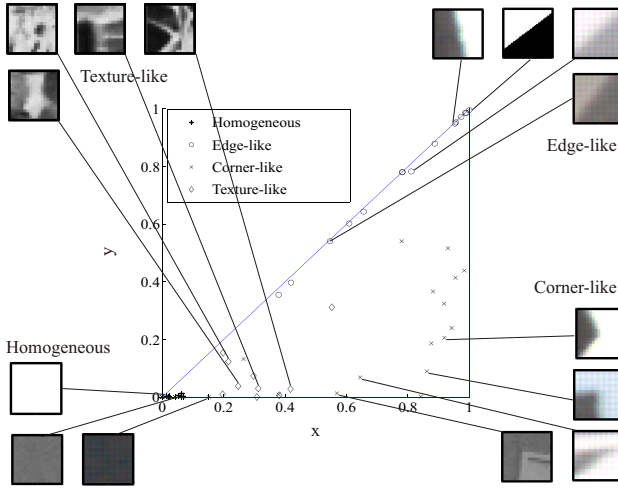
<sup>3</sup> Cognitive Vision Lab, Univ. of Southern Denmark, Denmark  
norbert@mip.sdu.dk

**Abstract.** In this paper, we describe work on Bayesian classifiers for distinguishing between homogeneous structures, textures, edges and junctions. We build semi-local classifiers from hand-labeled images to distinguish between these four different kinds of structures based on the concept of intrinsic dimensionality. The built classifier is tested on standard and non-standard images.

## 1 Introduction

Different kinds of image structures coexist in natural images: homogeneous image patches, edges, junctions, and textures. A large body of work has been devoted to their extraction and parametrization (see, e.g., [1,2,3]). In an artificial vision system, such image structures can have rather different roles due to their implicit properties. For example, processing of local motion at edge-like structures faces the aperture problem [4] while junctions and most texture-like structures give a stronger motion constraint. This has consequences also for the estimation of the global motion. It has turned out (see, e.g., [5]) to be advantageous to use different kinds of constraints (i.e., line constraints for edges and point constraints for junctions and textures) for these different image structures. As another example, in stereo processing, it is known that it is impossible to find correspondences at homogeneous image patches by direct methods (i.e., triangulation based methods based on pixel correspondences) while textures, edges and junctions give good indications for feature correspondences. Also, it has been shown that there is a strong relation between the different 2D image structures and their underlying depth structure [6,7]. Therefore, it is important to classify image patches according to their junction-ness, textured-ness, edge-ness or homogeneous-ness.

In many hierarchical artificial vision systems, later stages of visual processing are discrete and sparse, which requires a transition from signal-level, continuous, pixel-wise image information to sparse information to which often a higher semantic can be associated. During this transition, the continuous signal becomes discretized; *i.e.*, it is given discrete labels. For example, an image pixel whose contrast is above a given threshold is labeled as edge. Similarly, a pixel is classified as junction if, for example, the orientation variance in the neighborhood is high enough.



**Fig. 1.** How a set of 54 patches map to the different areas of the intrinsic dimensionality triangle. Some examples from these patches are also shown. The horizontal and vertical axes of the triangle denote the contrast and the orientation variances of the image patches, respectively.

The parameters of this discretization process are mostly set by its designer to perform best on a set of standard test images. However, it is neither trivial nor ideal to manually assign discrete labels to image structures since the domain is continuous. Hence, one benefits from *building* classifiers to give discrete labels to continuous signals. In this paper, we use hand-labeled image regions to *learn* the probability distributions of the features for different image structures and use this distribution to determine the type of image structure at a pixel. The local 2D structures that we aim to classify are listed below (examples of each structure is given in Fig. 1):

- Homogeneous image structures, which are signals of uniform intensities.
- Edge-like image structures, which are low-level structures that constitute the boundaries between homogeneous or texture-like signals.
- Junction-like structures, which are image patches where two or more edge-like structures with significantly different orientations intersect.
- Texture-like structures, which are often defined as signals which consist of repetitive, random or directional structures. In this paper, we define texture as 2D structures which have low spectral energy and high variance in local orientation (see Fig. 1 and Sect. 2).

Classification of image structures has been extensively studied in the literature, leading to several well-known feature detectors such as Harris [1], SUSAN [2] and

intrinsic dimensionality ( $iD$ )<sup>1</sup> [8]. The Harris operator extracts image features by shifting the image patch in a set of directions and measuring the correlation between the original image patch and the shifted image patch. Using this measurement, the Harris operator can distinguish between homogeneous, edge-like and corner-like structures. The SUSAN operator is based on placing a circular mask at each pixel and evaluating the distribution of intensities in the mask. The intrinsic dimensionality [8] uses the local amplitude and orientation variance in the neighborhood of a pixel to compute three confidences according to its being homogeneous, edge-like and corner-like (see Sect. 2). Similar to the Harris operator, SUSAN and intrinsic dimensionality can distinguish between homogeneous, edge-like and corner-like structures.

Up to the authors' knowledge, a method for simultaneous classification of texture-like structures together with homogeneous, edge-like and corner-like structures does not exist. The aim of this paper is to create such a classifier based on an extension of the concept of intrinsic dimensionality in which semi-local information is included in addition to purely local processing. Namely, from a set of hand-labeled images<sup>2</sup>, we learn local as well as semi-local classifiers to distinguish between homogeneous, edge-like, corner-like as well as texture-like structures. We present results of the built classifier on standard as well as non-standard images.

The paper is structured as following: In Sect. 2, we describe the concept of intrinsic dimensionality. In Sect. 3, we introduce our method for classifying image structures. Results are given in Sect. 4 with a conclusion in Sect. 5.

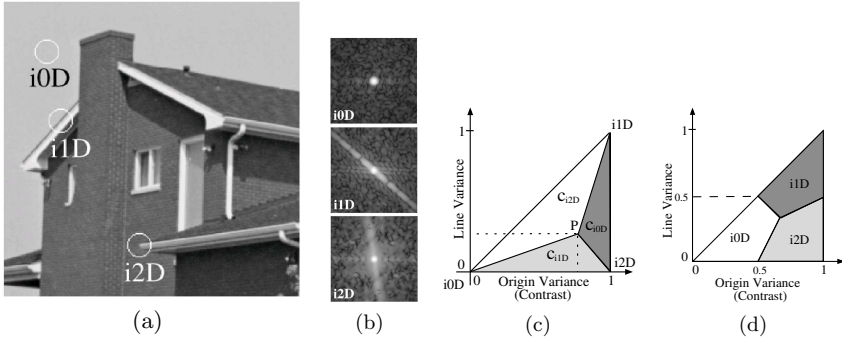
## 2 Intrinsic Dimensionality

When looking at the spectral representation of a local image patch (see Fig. 2(a,b)), we see that the energy of an  $i0D$  signal is concentrated in the origin (Fig. 2(b)-top), the energy of an  $i1D$  signal is concentrated along a line (Fig. 2(b)-middle) while the energy of an  $i2D$  signal varies in more than one dimension (Fig. 2(b)-bottom).

Recently, it has been shown [8] that the structure of the  $iD$  can be understood as a triangle that is spanned by two measures: origin variance and line variance. Origin variance describes the deviation of the energy from a concentration at the origin while line variance describes the deviation from a line structure (see Fig. 2(b) and 2(c)); in other words, origin variance measures non-homogeneity of the signal whereas the line variance measures the junctionness. The corners of the triangle then correspond to the 'ideal' cases of  $iD$ . The surface of the triangle corresponds to signals that carry aspects of the three 'ideal' cases, and the distance from the corners of the triangle indicates the similarity (or dissimilarity) to *ideal*  $i0D$ ,  $i1D$  and  $i2D$  signals.

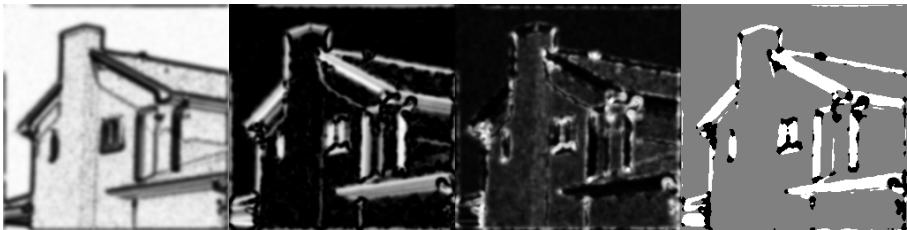
<sup>1</sup>  $iD$  assigns the names *intrinsically zero dimensional* ( $i0D$ ), *intrinsically one dimensional* ( $i1D$ ) and *intrinsically two dimensional* ( $i2D$ ) respectively to homogeneous, edge-like and junction-like structures.

<sup>2</sup> The software to label images is freely available for public use at [http://www.mip.sdu.dk/covig/software/label\\_on\\_web.html](http://www.mip.sdu.dk/covig/software/label_on_web.html)



**Fig. 2.** Illustration of the intrinsic dimensionality (Sub-figures (a,b,c) taken from [8]). (a) Three image patches for three different intrinsic dimensions. (b) The 2D spatial frequency spectra of the local patches in (a), from top to bottom:  $i0D$ ,  $i1D$ ,  $i2D$ . (c) The topology of  $iD$ . Origin variance is variance from a point, i.e., the origin. Line variance is variance from a line, measuring the junctionness of the signal.  $c_{iND}$  for  $N = 0, 1, 2$  stands for confidence for being  $i0D$ ,  $i1D$  and  $i2D$ , respectively. Confidences for an arbitrary point  $P$  is shown in the figure which reflect the areas of the sub-triangles defined by  $P$  and the corners of the triangle. (d) The decision areas for local image structures.

As shown in [8], this triangular interpretation allows for a *continuous formulation* of  $iD$  in terms of 3 confidences assigned to each discrete case. This is achieved by first computing two measurements of origin and line variance which define a point in the triangle (see Fig. 2(c)). The bary-centric coordinates (see, e.g., [9]) of this point in the triangle directly lead to a definition of three confidences that add up to one. These three confidences reflect the volume of the areas of the three sub-triangles which are defined by the point in the triangle and the corners of the triangle (see Fig. 2(c)). For example, for an arbitrary point  $P$  in the triangle, the area of the sub-triangle  $i0D$ - $P$ - $i1D$  denotes the confidence for  $i2D$  as shown in Fig. 2(c). That leads to the decision areas for  $i0D$ ,  $i1D$  and  $i2D$  as seen in Fig. 2(d). For the example image in Fig. 2, computed  $iD$  is shown in Fig. 3.



**Fig. 3.** Computed  $iD$  for the image in Fig. 2, black means zero and white means one. From left to right:  $c_{i0D}$ ,  $c_{i1D}$ ,  $c_{i2D}$  and highest confidence marked in gray, white and black for  $i0D$ ,  $i1D$  and  $i2D$ , respectively.

### 3 Methods

In this section, we describe the labeling of the images that we have used for learning and testing (Sect. 3.1), the basic theory for Bayesian classification (Sect. 3.2), the features we have used for classification (Sect. 3.3), as well as the three classifiers that we have designed (see Sect. 3.4).

#### 3.1 Labeling Images

As outlined in Sect. 1, we are interested in the classification of four image structures (*i.e.*, classes). To be able to compute the prior probabilities, we labeled a large set of images using a software that we developed. The software allows for the labeling arbitrary regions in an image, which are saved and then used for computing the prior probabilities (as well as evaluating the performance of learned classifiers that will be introduced in 3.4) for classifying image structures. Fig. 4 shows a few examples of labeled images patches.

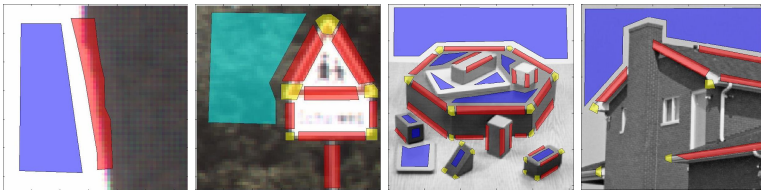
We labeled only image patches that were close to be the 'ideal' cases of their class because we did not want to make decisions about the class of an image patch which might be carrying aspects of different kinds of image structures. We would like a Bayesian classifier to make manifestations about the type of 'non-ideal' image patches based on what it has learned about the 'ideal' image structures.

#### 3.2 Bayesian Classification

If  $C_i$ , for  $i = 1, \dots, 4$ , represents on the the four classes, and  $X$  is the feature vector extracted for the pixel whose class has to be found, then the probability that the pixel belongs to a particular class  $C_i$  is given by the posterior probability  $P(C_i|X)$  of that class  $C_i$  given the feature vector  $X$  (using Bayes' Theorem):

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}, \quad (1)$$

where  $P(C_i)$  is the prior probability of the class  $C_i$ ;  $P(X|C_i)$  is the probability of feature vector  $X$ , given the pixel belongs to the class  $C_i$ ; and,  $P(X)$  is the total probability of the feature vector  $X$  (*i.e.*,  $\sum_i P(X|C_i)P(C_i)$ ).



**Fig. 4.** Images with various classes labeled. The colors blue, red, yellow and green correspond to homogeneous, edge-like, junction-like and texture-like structures, respectively.

A Bayesian classifier first computes  $P(C_i|X)$  using equation 1. Then, the classifier gives the label  $C_m$  to a given feature vector  $X_0$  if  $P(C_m|X_0)$  is maximal, *i.e.*,  $C_m = \arg \max_i \{ P(C_i|X) \}$ . The prior probabilities  $P(C_i)$ ,  $P(X)$  and the conditional probability  $P(X|C_i)$  are computed from the labeled images. The prior probabilities  $P(C_i)$  are 0.5, 0.3, 0.02 and 0.18 respectively for homogeneous, texture-like, corner-like and edge-like structures. An immediate conclusion from these probabilities is that corners are the least frequent image structures whereas homogeneous structures are abundant.

### 3.3 Features for Classification

As can be seen from Fig. 1, image structures have different neighborhood patterns. The type of an image structure at a pixel can be *estimated* from the signal information in the neighborhood. For this reason, we utilize the neighborhood of a given pixel for computing features that will be used for estimating the class of the pixel.

Now we define three features for each pixel  $P$  in the image. For two of these we define a neighborhood which is a ring of radius  $r^3$ :

- **Central feature** ( $x_{central}, y_{central}$ ): The co-ordinates of pixel  $p = (p_x, p_y)$  in the  $iD$  triangle (see Sect. 2):  $x_{central} = 1 - i0D_p$ ,  $y_{central} = i1D_p$ . The central feature has been used in [8] to distinguish between edges, corners and homogeneous image patches based on the barycentric co-ordinates. As we show in this work, it can also be used in a Bayesian classifier to characterize also texture, however not surprisingly with a large degree of misclassification in particular between texture and junctions.
- **Neighborhood mean feature** ( $x_{nmean}, y_{nmean}$ ): The mean value of the co-ordinates  $(x, y)$  in the  $iD$  triangle of all the pixels in the circular neighborhood of the pixel  $P$ . More formally,  $x_{nmean} = \frac{1}{N} \sum_{i=1}^N 1 - i0D_i$ ,  $y_{nmean} = \frac{1}{N} \sum_{i=1}^N i1D_i$ .
- **Neighborhood variance feature** ( $x_{nvar}, y_{nvar}$ ): The variance value of the co-ordinates  $(x, y)$  in the  $iD$  triangle of all the pixels in the neighborhood of pixel  $P$ . So,  $x_{nvar} = i0D_{nvar}$ ,  $y_{nvar} = i1D_{nvar}$ , where  $i0D_{nvar}$  and  $i1D_{nvar}$  are respectively the variance in the values of  $i0D$  and  $i1D$  in the neighborhood of pixel  $P$ .

The motivation behind using these three features is the following. The central feature represents the classical  $iD$  concept as outlined in [8] and has already been used for classification (however, not in a Bayesian sense). The neighborhood mean represent the mean  $iD$  value in the ring neighborhood. For edge-like structures it can be assumed that there will be  $iD$  values representing edges (at the

---

<sup>3</sup> The radius  $r$  has to be chosen depending on the frequency the signal is investigated at. In our case, we chose a radius of 3 pixels which reflects that the spatial features at that distance, although still sufficiently local, give new information in comparison to the  $iD$  values at the center pixel.

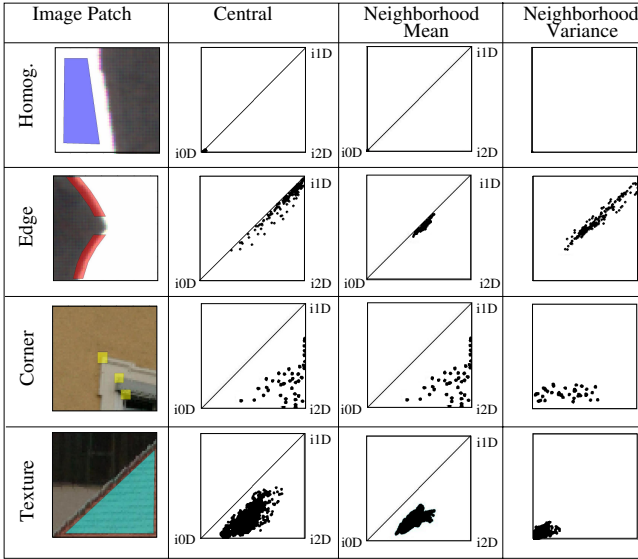
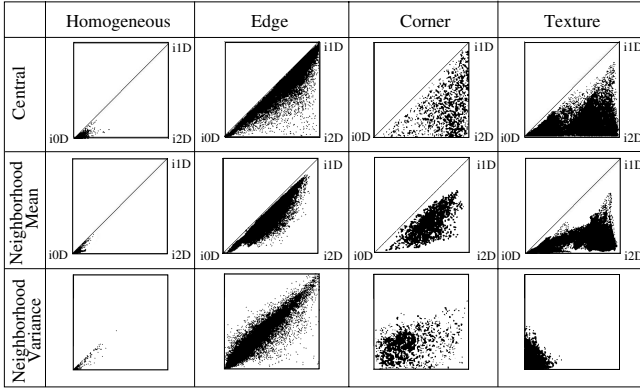


Fig. 5. The distributions of the features for each of the individual classes

prolongation of the edge at the center) as well as homogeneous image patches orthogonal to the edge. For junctions, there will be a more distributed pattern at the i2D corner while for textures, we will expect rather similar iD values on the ring due to the repetitive nature of texture. These thoughts will also be reflected in the neighborhood variance feature. Hence the two last features should give complementary information to the central feature. This is becoming clear looking at the distribution of these features over example structures as outlined in the next paragraph.

Fig. 5 shows the distribution of the features for selected regions in different images, and the total distribution of the features for each type of image structure is given in Fig. 6 (computed from a set of 65 images). The labeling process led to 91.500 labeled pixels which included 45.000 homogeneous, 20.000 edge-like, 1.500 corner-like and 25.000 texture-like pixels.

By observing the central feature distributions in Fig. 6, we see that many points labeled as corners have overlapping regions with textures and edges. However, we see from Fig. 6 that the neighborhood mean as well as the neighborhood variance can further help to distinguish between the four classes. Another important observation from Fig. 6 is that the neighborhood variance divides the points into two distinct divisions: the high variance classes (edges and corners) and the low variance classes (homogeneous and texture). This is due to the fact that edges and corners have, by definition, more variance in their neighborhood.



**Fig. 6.** The cumulative distribution of the features collected from a set of 65 images. There are 91,500 labeled pixels in total, which includes 45,000 homogeneous, 20,000 edge-like, 1,500 corner-like and 25,000 texture-like pixels.

### 3.4 The Classifiers

We design five classifiers:

- **Naive classifier (NaivC):** Classifier just using the  $iD$  based on barycentric co-ordinates, which is only able to distinguish junctions, homogeneous image patches and edges.
- **Central Bayesian Classifier (CentC):** The first and elementary Bayesian Classifier that we built is based on  $(x, y)$  co-ordinates of the pixel in the  $iD$  triangle, where  $x = 1 - i0DP$  and  $y = i1DP$ . Our experiments with this classifier showed that though it is good at detecting edges and the other classes, its detection of corners is poor: It could only detect only about 35% of the corners in the training set of images and only 20% in the test set. With the intention of building a better classifier, therefore, we decided to enhance the performance of the classifier by taking into account the features of the neighborhood of a pixel.
- **Classifier using neighborhood mean (NmeanC):** Our next classifier (NmeanC) is based on the central and neighborhood mean features of a pixel; *i.e.*, classifier NmeanC has the following feature vector:  $(x_{central}, y_{central}, x_{nmean}, y_{nmean})$ .
- **Classifier using neighborhood variance (NvarC):** Though classifier NmeanC is much better than the CentC, it made many errors in the detection of corners. We can observe from figure 6 that there is some overlap between the neighborhood mean distributions of corners and edges, and also corners and textures. With this observation, we build a classifier taking into account the central and neighborhood variance features of a pixel; *i.e.*, classifier NvarC has the following feature vector:  $(x_{central}, y_{central}, x_{nvar}, y_{nvar})$ .
- **Classifier using all features (CombC):** CombC consists of all three features: central, neighborhood mean and neighborhood variance; *i.e.*, classifier



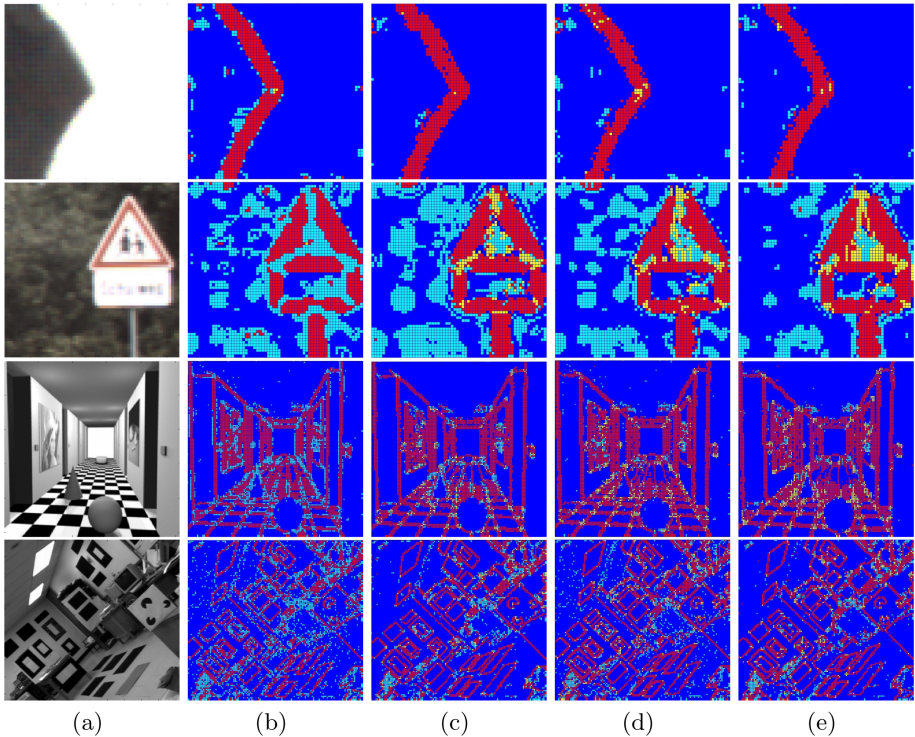
CombC has the following feature vector:  $(x_{central}, y_{central}, x_{nmean}, y_{nmean}, x_{nvar}, y_{nvar})$ .

## 4 Results

We used 85 hand-labeled images for training the classifiers. The performance of the classifiers on the training as well as the test set is given in table 1. Due to computational reasons, we were unable to test the CombC classifier.

**Table 1.** Accuracy (%) of the classifiers on the training set (in parentheses) and the non-training set. Since there is no training involved for the NaivC classifier, it is tested on all the images.

Class	NaivC	CentC	NmeanC	NvarC
Homogeneous	95	85 (88)	98 (99)	95 (99)
Edge	70	80 (85)	90 (95)	89 (97)
Corner	70	20 (35)	70 (97)	86 (98)
Texture	–	75 (83)	77 (96)	73 (90)



**Fig. 7.** Responses of the classifiers on a subset of the non-training set. Colors blue, red, light blue and yellow respectively encode homogeneous, edge-like, texture-like and corner-like structures.

We observe that the classifiers NmeanC, NvarC and CombC are good edge as well as corner detectors. Comparing NmeanC, NvarC and CombC against CentC, we can see that inclusion of neighborhood in the features improves the detection of corners drastically, and other image structures quite significantly (both on the training and non-training sets). Fig. 7 provide the responses of the classifiers on the non-training set. A surprising results is that combination of neighborhood variance and neighborhood mean features (CombC) performs worse than neighborhood variance feature (NvarC).

## 5 Conclusion

In this paper, we have introduced simultaneous classification of homogeneous, edge-like, corner-like and texture-like structures. This approach goes beyond current feature detectors (like Harris [1], SUSAN [2] or intrinsic dimensionality [8]) that distinguish only between up to three different kinds of image structures. The current paper has proposed and demonstrated a probabilistic extension to one of such approaches, namely the intrinsic dimensionality.

**Acknowledgements.** This work is supported by the EU Drivscop project (FP6-IST-FET-016276-2).

## References

1. Harris, C.G., Stephens, M.J.: A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conference, Manchester, pp. 147–151 (1988)
2. Smith, S., Brady, J.: SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision* 23(1), 45–78 (1997)
3. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(10), 1615–1630 (2005)
4. Kalkan, S., Calow, D., Wörgötter, F., Lappe, M., Krüger, N.: Local image structures and optic flow estimation. *Network: Computation in Neural Systems* 16(4), 341–356 (2005)
5. Rosenhahn, B., Sommer, G.: Adaptive pose estimation for different corresponding entities. In: Van Gool, L. (ed.) DAGM 2002. LNCS, vol. 2449, pp. 265–273. Springer, Heidelberg (2002)
6. Grimson, W.: Surface consistency constraints in vision. *CVGIP* 24(1), 28–51 (1983)
7. Kalkan, S., Wörgötter, F., Krüger, N.: Statistical analysis of local 3D structure in 2D images. In: *IEEE Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 1114–1121 (2006)
8. Felsberg, M., Kalkan, S., Krüger, N.: Continuous dimensionality characterization of image structures. *Image and Vision Computing* (2008) (in press)
9. Coxeter, H.: *Introduction to Geometry*, 2nd edn. Wiley & Sons, Chichester (1969)