Key-point detection with multi-layer center-surround inhibition

Foti Coleca[1], Sabrina Zîrnovean[1,2], Thomas Käster[1,3], Thomas Martinetz[1],
and Erhardt Barth[1]

[1]*Institute for Neuro- and Bioinformatics, University of Lübeck, Ratzeburger Allee 160, Lübeck 23562, Germany*

[2]*University "POLITEHNICA" of Bucureşti, Splaiul Independenţei 313, 060042, Bucureşti, Romania*

[3]*Pattern Recognition Company GmbH, Innovations Campus Lübeck, Maria-Goeppert-Straße 3, 23562 Lübeck, Germany*

{*coleca, sabrina, kaester, martinetz, barth*}@*inb.uni-luebeck.de*

Abstract: We present a biologically inspired algorithm for key-point detection based on multi-layer and nonlinear center-surround inhibition. A Bag-of-Visual-Words framework is used to evaluate the performance of the detector on the Oxford III-T Pet Dataset for pet recognition. The results demonstrate an increased performance of our algorithm compared to the SIFT key-point detector. We further improve the recognition rate by separately training codebooks for the ON- and OFF-type key points. The results show that our key-point detection algorithms outperform the SIFT detector by having a lower recognition-error rate over a whole range of different key-point densities. Randomly selected key-points are also outperformed.

1 INTRODUCTION

Various object-recognition and tracking approaches rely on the detection of object key-points (Lowe, 1999; Csurka et al., 2004). Regions around such key-points are expected to contain the information needed for matching and recognition. Although some have argued that a dense sampling of randomly selected points can replace or even outperform recognition algorithms based on key-points (Nowak et al., 2006), the benefits of appropriate key-points and related saliency measures are obvious since they reduce the amount of information that needs to be processed and often, as in many cases including ours, lead to better recognition performance, see (Vig et al., 2012a) for a recent example. The standard for key-point detection is the SIFT algorithm, which selects key-points as the local maxima of the Difference-of-Gaussians (DoG) operator (Lowe, 1999).

We here show how SIFT key-point selection can be improved by introducing a few biologically inspired extensions and new features. The DoG operator has been often used to model lateral inhibition, a key feature of early visual processing. However, some essential properties of biologically realistic lateral in-

hibition are usually ignored and it turns out that three of them fit our purpose of improving key-point selection. The first property is that of nonlinearity, which is here modeled as a simple one-way rectification (clipping) of the DoG. The second property is that of ON/OFF separation, i.e. the distinct representation of bright-on-dark versus dark-on-bright features. The third property is that of multiple layers, i.e., the fact that lateral inhibition occurs repeatedly in successive layers of early visual processing. All these features have been considered in (Barth and Zetzsche, 1998), where the authors have shown that iterated and nonlinear lateral inhibition generates representations of increasing sparseness and converges to end-stopped representations, i.e., representations of only 2D features like corners and junctions. In other words: the linear DoG operator eliminates 0D regions, which are uniform, and the iteration of the clipped DoG then eliminates 1D regions, i.e., straight edges and lines. This approach, however, has not been used for key-point selection in the context of object recognition.

Operators that extract 2D features, called end-stopped operators in vision, are thought to provide an efficient representation by eliminating redundancies in images and videos (Zetzsche and Barth, 1990; Vig et al., 2012b). This works well because 0D and 1D regions are redundant in a geometrical sense (Mota and Barth, 2000). Statistical approaches consider a

representation to be efficient if statistical dependencies in the signal are reduced. Linear operators such as the DoG, however, can only decorrelate the signal and nonlinear operations such as those advocated in (Zetzsche and Barth, 1990) are needed to deal with higher-order dependencies. However, isotropic operators such as the DoG, even when combined with simple nonlinearities, are limited in their ability to deal with higher-order dependencies.

Although the principle of iterating the nonlinear DoG has been shown to provide 2D representations, this insight was only supported by simulations. However, the computational power of linear-non-linear (LNL) structures has been further analyzed in (Zetzsche and Nuding, 2007) and it has been shown how LNL networks can reduce the statistical dependencies in natural images. The idea is to learn the linear part of the LNL sandwich, e.g., by PCA for decorrelation, then apply a simple nonlinearity that would introduce new correlations (since the nonlinearity can map higher-order dependencies to second-order dependencies), which can then be removed in the next linear stage. These results can explain why the iteration of the nonlinear DoG can progress towards removing more and more redundancies.

More recently, deep learning algorithms have performed very well in object-recognition tasks (Cireşan et al., 2010; Bengio, 2009; Hinton, 2007). The first layers of these deep multi-layer networks are meant to provide efficient representations and are trained by unsupervised learning. When applied to natural images, these algorithms often provide representations that exhibit an increasing degree of sparseness as one proceeds from the first to the subsequent layers. Based on the above considerations one can thus interpret the multiple layers of the deep networks as providing an increasing degree of higher-order decorrelation, which reduce the statistical dependencies between the components of a representation vector. In this context, one can interpret the iterated DoG proposed here as an hard-wired implementation of a deep multi-layer network where the linear decorrelating operations are not learned but simply implemented as a DoG.

Given the above considerations, we evaluate the performance of our key-point detectors for different numbers of layers and different numbers of selected key-points. We use a Bag-of-Words approach on a pet-recognition benchmark and the popular SIFT algorithm as a base line.

2 SALIENCY OPERATORS

Key-points are usually selected randomly or according to some saliency measure. We base our key-point detection algorithm on the iterated nonlinear inhibition operator presented in (Barth and Zetzsche, 1998). There it is shown how iterations of a simple nonlinear operator can exhibit an increasing end-stopping behavior, making it an obvious candidate for key-point detection. The balance between 1D and 2D features that may lead to key points depends on the number of iterations.

2.1 MULTI-LAYER NONLINEAR INHIBITION

We define the nonlinear inhibition of a by b $(a, b \in \mathbb{R})$ as

$$N[N[a] - N[b]] \tag{1}$$

where $N[\ ]$ is a nonlinearity in the form of half-wave rectification

$$N[x] = Max(x, 0). \tag{2}$$

The multi-layer (iterated) nonlinear Difference-of-Gaussian (INDoG) operator is defined by taking a and b to be the outputs of Gaussian low-pass filters of different spatial extent:

$$f_{i+1} = N[N[g_1 \otimes f_i] - N[g_2 \otimes f_i]]; \ i = 0, \dots, M \tag{3}$$

where i is the layer index, $\otimes$ denotes convolution, and g_1, g_2 are Gaussian convolution kernels with variances $\sigma_1 < \sigma_2$, the inhibitory filter being larger. Initially f_0 is equal to the image intensity.

2.2 ON/OFF SEPARATION

As the INDoG operator exhibits responses only for certain features (responds strongest at bright corners and blobs, i.e. like an ON-type operator), one needs to define a pair of operators in order to obtain ON/OFF separation, i.e., distinct responses to bright and dark features.

Accordingly, the CON operator is defined as

$$f_{i+1} = N[N[g_1 \otimes f_i] - N[g_2 \otimes f_i]]; \ i = 0, \dots, M \tag{4}$$

while the COFF type operator is obtained by

$$\begin{aligned} f_1 &= N[N[g_2 \otimes f_0] - N[g_1 \otimes f_0]] \\ f_{i+1} &= N[N[g_1 \otimes f_i] - N[g_2 \otimes f_i]]; \ i = 0, \dots, M \end{aligned} \tag{5}$$

i.e. by reversing the inhibition at the first iteration.

The responses of these operators on synthetic and natural images over $M = 8$ iterations are shown in Fig. 1. For the two squares, it can be seen how the edge response gets progressively weaker and disappears at the end, leaving only the corners to have distinct peaks. In the natural image one can observe the features becoming more separated over the iterations. Larger features can be detected by running the iterations in a DoG scale space, i.e., a collection of different-scale responses are shown in Fig. 3.

3 PERFORMANCE EVALUATION

Evaluations of features are often done via a descriptor repeatability or matching score (Mikolajczyk et al., 2005). However, this is more suitable for key-point detector evaluations used in image matching applications. In our case, we want to evaluate how much the key-points contribute to object recognition and therefore use a object-recognition benchmark.

3.1 The Oxford III-T Pet Dataset

The dataset consists of 2371 pictures of cats and 4978 pictures of dogs. The images are further split into specific breeds, 25 of which are dogs and 12 are cats. The images have been downloaded from various websites on the Internet and therefore exhibit a high degree of variation, the pets being photographed in various poses, at different illumination levels, and the pictures themselves being at different resolutions. Adding to this is a high amount of intra-class variation due to the different breed types being represented.

The dataset was used in (Parkhi et al., 2012) resulting in recognition rates from 83% to 96% for cat-versus-dog classification. We do not use these as a baseline, as their methods are not comparable with ours (multiple regions are defined which are densely sampled, yelding a 20,000 dimensional feature vector in the simplest approach). Also, although annotations in the form of bounding boxes and pet silhouettes are available, we do not use the additional data since we only wish to asses the contribution of the different types of key-points in general, and for a difficult tasks.

3.2 Key-point Detectors

For evaluation, we compare our algorithm to Lowe's (Lowe, 1999) SIFT key-point detector as implemented in the VLFeat (Vedaldi and Fulkerson, 2010) library. The SIFT detector is also based on a DoG scale-space analysis of the image, along with some
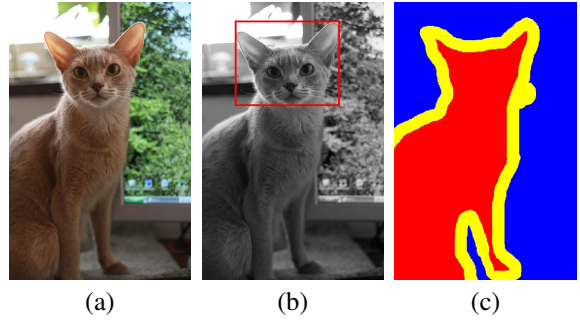

(a)   (b)   (c)

Figure 2: Sample image from the Oxford III-T Pet Dataset, along with face (b) and foreground/background/contour (c) annotations.

enhancements that suppress low contrast and straight-edge responses. The SIFT keypoints that were used in the evaluation were outputted by the unmodified VLFeat package with the default parameters.

In order to make the comparison as close as possible, the INDoG detector was implemented as a modification of the open-source VLFeat package by replacing the usual DoG pyramid used in the SIFT detector with our own INDoG CON/COFF frames and skipping the principal curvature analysis for edge suppression. The INDoG algorithm was run 4 times, for 1, 2, 4 and 8 iterations. As shown in Equations (4) and (5), the first iteration is directly equivalent to the SIFT algorithm, but without the edge suppression enhancements. All parameters, e.g., for the Gaussian kernels, the numbers of scales etc. are the same as the defaults used by the SIFT detector in all cases. This way, the main differences between the SIFT key-points and ours are the iterations (multiple layers) and the ON/OFF separation, and we can study the effect of the number of layers on the recognition performance.

As a control, we included randomly generated key-points at strides of 4, 6, 8 and 10 pixels. The randomly chosen key-points are used to extract SIFT features of random scales ranging from 12 to 30 pixels width per bin with the help of the VL Feat framework. The orientation of the features is also calculated as in the VL Feat framework implementation of the SIFT features.

3.3 The Bag-of-Visual-Words Framework

We evaluate the performance of our detector using a Bag-of-Visual-Words (BoVW) framework. First, the few images that are of excessive size (more than 1000 pixels horizontally or vertically) are discarded. The positives (cats) and negatives (dogs) are randomized and split into 80% training and 20% testing groups.
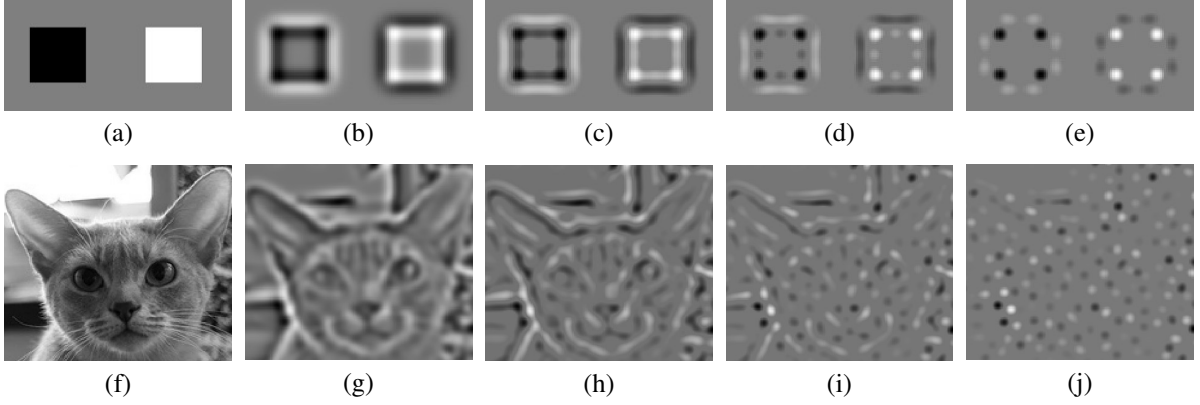
Figure 1: Synthetic (a) and natural (f) images, and their ON/OFF responses (b)-(e), (g)-(j) after 1, 2, 4 and 8 iterations of the INDoG algorithm on a single scale of the frame. Top row: squares are 100×100 pixels in size, taken from octave 2, scale 3 of the DoG scale space ($\sigma_1 = 12.8$, $\sigma_2 = 16.08$ pixels). Bottom row: picture size is 180×160 pixels, taken from octave 0, scale 4 of the DoG scale space ($\sigma_1 = 4.02$, $\sigma_2 = 5.05$ pixels). The ON responses are shaded in white, and the OFF responses in black.
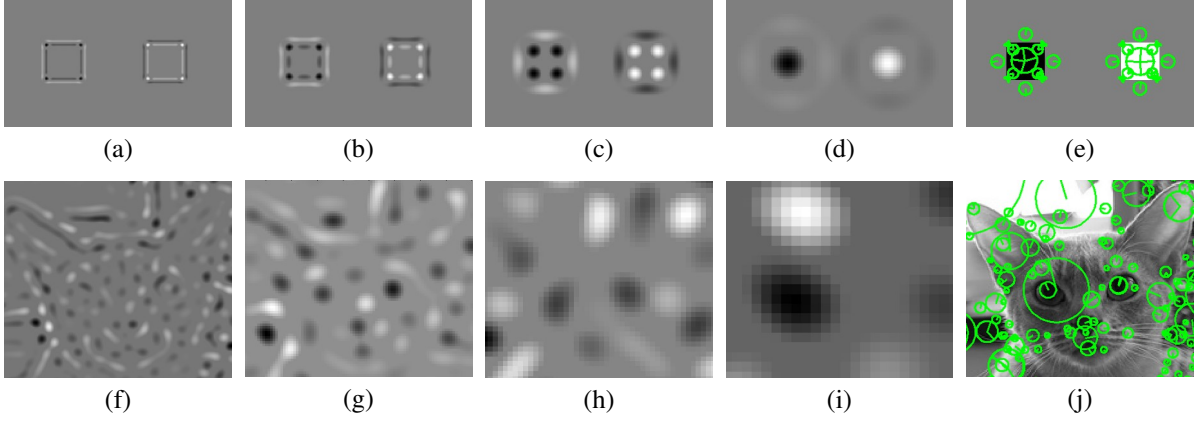

Figure 3: Different scales of the ON/OFF responses on synthetic (a)-(d) and natural (f)-(i) images, and the key-points found by the INDoG algorithm (the first 100 as ordered by their magnitude). Both rows: frames are taken from iteration 4 of the INDoG algorithm, at octaves 0, 1, 2 and 3, scale 4 of the DoG scale space ($\sigma_{01} = 4.03, \sigma_{02} = 5.07$ pixels, $\sigma_{s+1} = 2\sigma_s$).

The detector is run on the images and the key-points that have been found are ordered by their magnitude. The total number of key-points is then limited at 0.25% of the picture size for all detectors, the rest of the key-points being discarded. Note that key-points that return multiple orientation for the same location are only counted as one towards the total number of key-points allowed (i.e. only key-points at differing locations are counted).

SIFT features are extracted at the key-point locations, using the rotation and scaling provided by the detector for each key-point. In order to evaluate the discriminative power of sparse key-points, a number of subsets are created by discarding a portion of the features, creating a total of 9 training/test sets containing the top 100%, 50%, 30%, 20%, 10%, 5%, 3%, 2% and 1% key-points (as ordered by their magni-

tude) of the maximum number of allowed key-points per frame.

The BoVW codebook is created using an online K-means algorithm on all the extracted features in each feature subset separately. The number of clusters in each codebook was empirically set as the square root of the number of key-points in that particular subset (i.e. for 1 million key-points, which was approximately the allowed size of the full positive training set, we would set $k = 1000$). This codebook is used to generate keyword histograms for each image in the current set. The histograms are then normalized with the l^1 norm and used in a support vector machine (SVM) for training and classification. We use a linear kernel SVM with the softness parameter C set to 128 for all tests. This value has been chosen based on several test runs, and it did not influence the relative

performance of the different methods tested here.

The above steps are repeated 5 times with a different subset of the positive/negative images for a 5-fold cross validation scheme, the resulting error rates being averaged at the end. Note that the codebooks are generated from scratch for every set and subset separately, no codebooks being reused at any point during the performance evaluation.

3.4 Further Enhancements

There are many ways of enhancing the VBoW approach, e.g. by improving the codebook creation step. Here we propose a very simple extension, which is to separate the ON- and OFF-type key-points, see Section 2.2. This is implemented by separately training two codebooks for the CON and COFF key-points. We expect the recognition rate to improve due to the additional information pertaining to the scene structure that was extracted at that particular key-point. A further motivation for ON/OFF separation is that ON/OFF streams are segregated in biological vision systems.

As mentioned in section 3.1, the dataset contains additional annotations such as bounding boxes and pet silhouettes. An obvious improvement therefore would be to restrict the codebook generation and learning process only on the relevant portions of the images, which has been shown to boost the recognition rate on this dataset (Parkhi et al., 2012). However, as we are interested in evaluating the overall performance of the detectors in a general sense (i.e. without a priori information about the object location), we do not use these annotations.

4 RESULTS

The results are detailed in Table 1, together with the variance (1σ) obtained by performing a five-way cross validation (for clarity, error bars were not included in Figure 4). Note that the average performance of the INDoG algorithm surpasses the SIFT key point detector as well as the random sampling when using the maximum allowed number of key points, and is more apparent at lower key point densities.

From Figure 4 it can be seen that, in general, the recognition performance increases with the number of key-points. When comparing the different curves, one finds a significant improvement due to the use of several layers instead of just one layer of lateral inhibition, SIFT key points and randomly selected key

points are outperformed by multi-layer key-point detectors. The right plot in Figure 4 demonstrates a further improvement, which is obtained by separating the ON/OFF responses as describe in Sections 2.2 and 3.4. This improvement is also greater and more consistent at lower key point densities, as seen in Table 1.

In Figure 5 the spatial distribution of the found key points is shown, within a typical example of an uncluttered picture from the dataset. It is interesting to note that the key points found by the SIFT detector as well as the first iterations of the INDoG algorithm are more spread throughout the picture, even in the uniform background area. In an uncluttered frame such as the one shown, one could expect to see results similar to the higher iterations of the INDoG algorithm, i.e. key points concentrated on the object of interest, as it exhibits the highest variations.

We can speculate that one of the reasons for the increased performance would be the distribution of key points on only the subjects in uncluttered frames, i.e., the detector avoiding detection of key points in uniform areas. Another reason might be an optimal balance, controlled by the number of iterations, in the allocation of key points to 1D and 2D features. Both factors would lead to better codewords and an overall improved performance of the BoVW framework.

5 DISCUSSION

Although the improvements in performance relative to SIFT key-points are not huge, our results show that a very simple, biologically plausible, lateral-inhibition scheme can be used successfully for key-point detection. The computations have low complexity and can be implemented as a parallel network. Moreover, they can be implemented as neuromorphic hardware on retina-like image sensors (Indiveri et al., 2011), and it seems an interesting results that looping the image through such a hardware will lead to key points.

Unfortunately, as with other deep-networks, an thorough analytical treatment and a deeper understanding of the nonlinear iterative computations is currently missing and we therefore had to rely on some heuristics, such as the notion of higher-order decorrelation, to explain our results.

Various extensions and improvements are possible. As already suggested in (Barth and Zetzsche, 1998), one can replace the DoG with a RoG, i.e., a ratio of Gaussians. The resulting INRoG is sensitive to image contrast as opposed to difference in image intensity. Furthermore, one can optimize the ratio be-
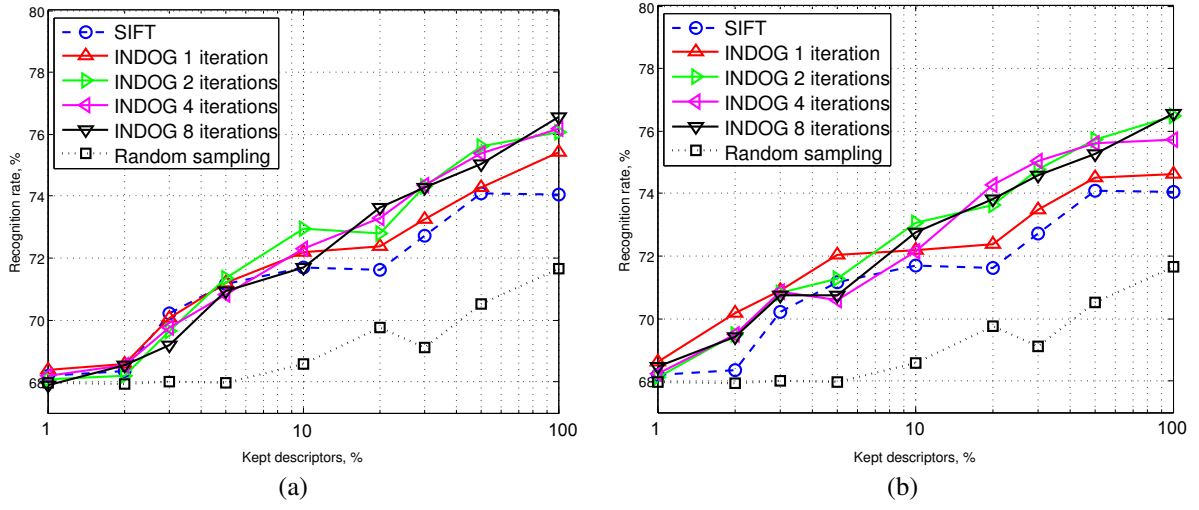
Figure 4: (a) Results for INDoG, SIFT and random key-point sampling at different key-point densities; (b) Results for the INDoG algorithm with separate CON/COFF codebooks versus SIFT and random sampling (same as in (a)). The number of key-points on the right (100%) is equal to 0.25% of the total number of pixels in the image.

Descriptors kept	1%	2%	3%	5%	10%	20%	30%	50%	100%
SIFT	68.2%	68.3%	70.2%	71.1%	71.7%	71.6%	72.7%	74.1%	74.0%
±1σ	0.42	0.33	0.69	0.73	0.90	0.31	0.70	1.71	1.56
INDoG, 1 iteration	68.3%	68.6%	70.0%	71.2%	72.2%	72.3%	73.2%	74.3%	75.4%
±1σ	0.42	0.36	0.72	1.18	0.77	0.82	0.40	1.01	0.59
ON/OFF separation	**68.6**%	**70.1**%	**70.9**%	**72.1**%	72.2%	72.4%	73.5%	74.5%	74.6%
±1σ	0.32	0.89	0.46	0.87	0.88	1.80	1.17	0.81	1.30
INDoG, 2 iterations	68.1%	68.2%	69.7%	71.4%	72.9%	72.8%	74.3%	75.6%	76.0%
±1σ	0.22	0.10	0.44	0.75	1.17	1.13	0.81	1.16	1.17
ON/OFF separation	68.1%	69.5%	70.8%	71.3%	**73.0**%	73.6%	74.7%	**75.7**%	76.5%
±1σ	0.54	0.35	0.30	0.46	1.05	0.67	1.80	0.63	1.02
INDoG, 4 iterations	68.2%	68.5%	69.7%	70.8%	72.3%	73.2%	74.4%	75.4%	76.2%
±1σ	0.22	0.47	0.38	0.62	0.62	0.32	0.75	1.38	0.83
ON/OFF separation	68.2%	69.5%	**70.9**%	70.6%	72.1%	**74.3**%	**75.0**%	75.6%	75.7%
±1σ	0.33	0.58	0.70	0.54	0.47	0.45	0.84	0.41	1.10
INDoG, 8 iterations	67.9%	68.6%	69.2%	70.9%	71.7%	73.6%	74.3%	75.0%	**76.6**%
±1σ	0.53	0.65	0.55	0.60	1.02	0.72	0.87	1.35	1.01
ON/OFF separation	68.5%	69.4%	70.7%	70.7%	72.7%	73.8%	74.6%	75.3%	76.5%
±1σ	0.42	0.60	0.60	0.38	1.59	1.09	0.64	1.14	1.47
Random sampling	67.9%	67.9%	68.0%	68.0%	68.4%	70.1%	69.6%	70.7%	71.8%
±1σ	0.08	0.06	0.15	0.10	0.71	1.59	0.91	1.45	0.66

Table 1: Results for SIFT, INDoG 1, 2, 4, and 8 iterations (with and without CON/COFF separation) and random sampling. The best results in each column are printed in bold type.
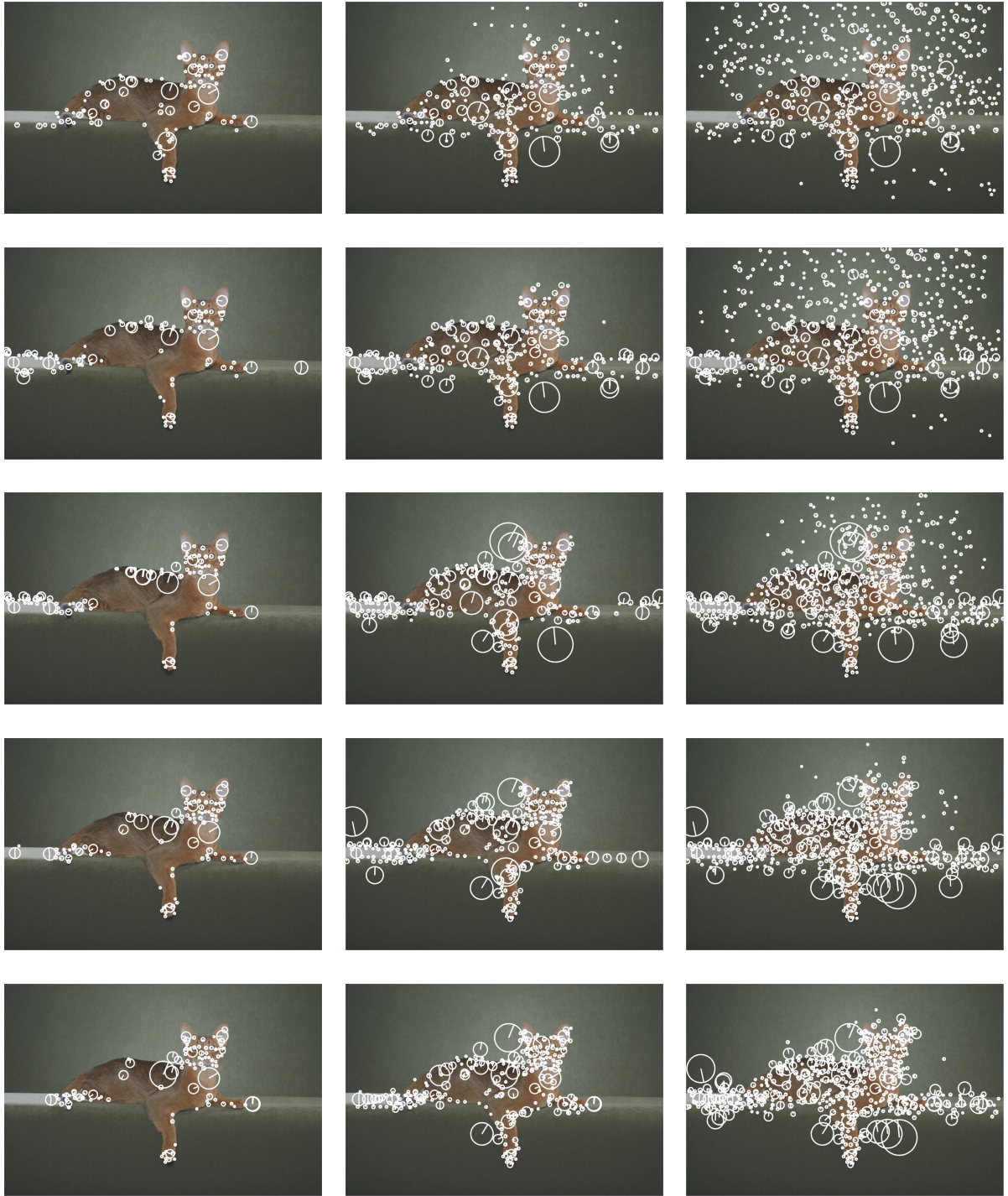
Figure 5: Comparison between keypoints generated by SIFT (first row) and the INDoG algorithm (next rows, from top to bottom: 1, 2, 4 and 8 iterations). From left to right, the first 20%, 50% and 100% of the maximum number of keypoints allowed per image (defined as 0.25% of the total number of pixels in the image, i.e. the same number of keypoints column-wise), sorted by magnitude. The background picture has had its brightness and contrast lowered to facilite the viewing of the keypoints.

tween the two Gaussians, which has been here taken, for better comparison, from the SIFT implementation, although more stable results are obtained with a larger surrounds (Barth and Zetzsche, 1998). Finally, extensions to spatio-temporal DoGs, i.e., to moving images are possible.

6 CONCLUSION

We have shown that the introduction of multiple layers provides a saliency measure and derived key-points that enhance recognition performance relative to just one layer of lateral inhibition. The enhancement seems to be due to the fact that straight edges are progressively suppressed as the number of layers increases.

Surprisingly, performance is enhanced relative to the SIFT key-points, which use second order derivatives and the determinant of the Hessian to suppress straight edges in the saliency measure.

By using the arguments outlined in the Introduction, the role of the multiple layers can be understood as a form of higher-order decorrelation of the input, which leads to more efficient representations with increased sparseness and more representative key-points.

ACKNOWLEDGEMENTS

REFERENCES

Barth, E. and Zetzsche, C. (1998). Endstopped operators based on iterated nonlinear center-surround inhibition. In *Human Vision and Electronic Imaging III*, volume 3299 of *Proc. SPIE*, pages 67–78, Bellingham, WA.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Cireşan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of key-points. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22.

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434.

Indiveri, G., Linares-Barranco, B., Hamilton, T., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.-C., Dudek, P., Häfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., Saighi, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., and Boahen, K. (2011). Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5:1–23.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72.

Mota, C. and Barth, E. (2000). On the uniqueness of curvature features. In *Dynamische Perzeption*, volume 9 of *Proceedings in Artificial Intelligence*, pages 175–178, Köln. Infix Verlag.

Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. (2012). Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3498–3505. IEEE.

Vedaldi, A. and Fulkerson, B. (2010). Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, pages 1469–1472. ACM.

Vig, E., Dorr, M., and Cox, D. (2012a). Saliency-based selection of sparse descriptors for action recognition. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1405 – 1408.

Vig, E., Dorr, M., Martinetz, T., and Barth, E. (2012b). Intrinsic dimensionality predicts the saliency of natural dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1080–1091.

Zetzsche, C. and Barth, E. (1990). Fundamental limits of linear filters in the visual processing of two-dimensional signals. *Vision Research*, 30:1111–1117.

Zetzsche, C. and Nuding, U. (2007). Nonlinear encoding in multilayer LNL systems optimized for the representation of natural images. In *Human Vision and Electronic Imaging XII*, volume 6492 of *Proc. SPIE*, pages 649204–649204–22.