A three-layer model of natural image statistics

Michael U. Gutmann[a], Aapo Hyvärinen[a]

[a]*Dept of Mathematics and Statistics, Dept of Computer Science and HIIT*
P.O. Box 68 FIN-00014 University of Helsinki, Finland,
{michael.gutmann,aapo.hyvarinen}@helsinki.fi

Abstract

An important property of visual systems is to be simultaneously both selective to specific patterns found in the sensory input and invariant to possible variations. Selectivity and invariance (tolerance) are opposing requirements. It has been suggested that they could be joined by iterating a sequence of elementary selectivity and tolerance computations. It is, however, unknown what should be selected or tolerated at each level of the hierarchy. We approach this issue by learning the computations from natural images. We propose and estimate a probabilistic model of natural images that consists of three processing layers. Two natural image data sets are considered: image patches, and complete visual scenes downsampled to the size of small patches. For both data sets, we find that in the first two layers, simple and complex cell-like computations are performed. In the third layer, we mainly find selectivity to longer contours; for patch data, we further find some selectivity to texture, while for the downsampled complete scenes, some selectivity to curvature is observed.

Keywords: Natural images, probabilistic modeling, visual processing, selectivity, invariance, sparse coding, deep learning

1. Introduction

Our paper belongs to the larger body of work on Bayesian perception. This theory of vision entails that the visual system is adapted to the properties of the world which it senses. In other words, it "knows" about the regularities within the visual stimuli (see, for example, Barlow, 2001; Simoncelli and Olshausen, 2001; Hyvärinen et al., 2009; Freeman and Simoncelli, 2011). Knowledge about the regularities can be mathematically expressed as

knowledge about the probability distribution of the visual stimuli. Our goal here is to model this distribution and relate it to visual processing.

One powerful class of models specifies the distribution in a top-down manner in terms of latent variables which explain the structure in the visual stimuli (Olshausen and Field, 1996; Hyvärinen et al., 2009; Karklin and Lewicki, 2009; Zoran and Weiss, 2009; Ranzato and Hinton, 2010; Cadieu and Olshausen, 2012). Another class of models corresponds to a bottom-up approach where the visual stimuli are processed in multiple layers of computation (Osindero et al., 2006; Köster and Hyvärinen, 2010; Gutmann and Hyvärinen, 2012b). The model in this paper belongs to this latter class.

It has been proposed that the layers should alternate between elementary selectivity and invariance, or tolerance computations (Fukushima, 1980; Riesenhuber and Poggio, 1999). In line with simple models for experimental data (see, for example, Hubel, 1995), the first layer should be selective to localized, oriented bandpass structure, and the second layer should be tolerant to variations in the localization of that structure. The idea is that after several layers of computations, high selectivity to specific structure could be combined with moderate tolerance to its possible variations. The combination of the opposing poles of selectivity and invariance is thought to be essential for reliable object recognition, or for biological and artificial visual processing in general (DiCarlo and Cox, 2007; Serre et al., 2007; Jarrett et al., 2009; Rust and Stocker, 2010).

A fundamental question that arises with the bottom-up approach is to know what should be selected or tolerated at each layer. We approach this issue by learning the selectivity and tolerance computations from natural images. This approach has previously accounted for the computations on the first two layers (Osindero et al., 2006; Köster and Hyvärinen, 2010; Gutmann and Hyvärinen, 2012b). Here, we learn all layers in a three-layer model, and pay particular attention to the computations which emerge in the third layer.[1]

2. Material and methods

In Section 2.1, we present the natural image data used. In Section 2.2, we introduce and explain the parametric model of the processing in the three

[1]Preliminary results were reported at the International Conference on Pattern Recognition 2012 (Gutmann and Hyvärinen, 2012a).

layers. Section 2.3 shows how to learn the parameters by fitting a probability density function to the natural image data.

2.1. Data and preprocessing

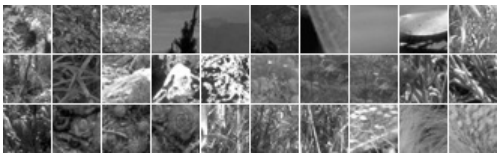
We use two types of natural image data. The first data set consists of image patches that we have extracted from thirteen larger gray-scale images which have been used before to study properties of natural images (Hyvärinen et al., 2009). The patches are of size 32×32 pixels. The second data set is the tiny images data set by Torralba et al. (2008), converted to gray scale. That data set consists of about eighty million images which show complete visual scenes downsampled to 32×32 pixels. Examples from the two data sets are shown in Figures 1a and 1b. When referring to both data sets at the same time, we will call them "natural images".

As preprocessing, we removed the DC component (average value of each tiny image, or image patch) and normalized the norm of the resulting image. The norm used here was computed in the whitened space. Unlike the ordinary norm without whitening, this norm is not dominated by the low-frequency content of an image (Hyvärinen et al., 2009, Chapter 5). This preprocessing is a form of luminance and contrast gain control. Further, the preprocessing makes it easier to model the statistical dependencies between the pixels by normalizing their marginal distributions to some extent. This preprocessing thus is motivated by both neuroscience and data-modeling considerations. After normalization, we reduced the dimensionality by PCA from 1024 to 600, which corresponds to low-pass filtering of the images. After dimension reduction, the images are elements inside a 600 dimensional sphere. The retained dimensions account for a bit more than 98% and 99% of the variance of the image patches and the tiny images, respectively. We denote the resulting, preprocessed images by $\mathbf{x}$.

Figures 1c and 1d show the effect of the preprocessing for the natural image examples in 1a and 1b, respectively. For visualization, we scaled each preprocessed natural image such that the full colormap is used. The examples visualize the luminance and contrast gain control, and they show further that our dimension reduction does not cause a perceptible blurring.

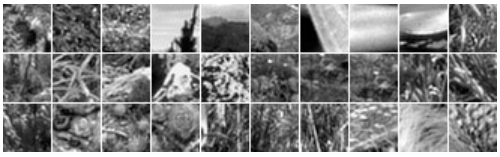
2.2. Parametric model for the three layers of computation

After the initial preprocessing (gain control), an input image is processed in three layers of computation. The outputs of each layer form statistics which we use in Section 2.3 to define the value that a probability density



(a) Raw image patches

(b) Raw tiny images



(c) Preprocessed image patches

(d) Preprocessed tiny images

Figure 1: Natural images before and after preprocessing. (a,b) Examples of extracted patches from larger images and examples from the tiny images data set. Pixel values of zero are shown in black, and white corresponds to pixel values of 255. (c,d) The same images after preprocessing. Preprocessing consists of removing the average value from each image, standardizing its norm, and PCA-based dimension reduction. Each preprocessed image was re-scaled to use the full color map.

function $p_{\mathbf{x}}$ takes at $\mathbf{x}$, that is, at a given image after gain control. The three layers are defined as follows.

First layer. The gain-controlled image $\mathbf{x}$ is projected onto features $\mathbf{w}_i^{(1)}$, followed by half-wave rectification. This gives the outputs $y_i^{(1)}$ of the first-layer units,

$$y_i^{(1)} = \max\left(\mathbf{w}_i^{(1)} \cdot \mathbf{x},\, 0\right), \quad i = 1 \dots n^{(1)}. \tag{1}$$

Here, $\mathbf{w}_i^{(1)} \cdot \mathbf{x}$ denotes the dot-product between the vectors $\mathbf{w}_i^{(1)}$ and $\mathbf{x}$. The features $\mathbf{w}_i^{(1)}$ are parameters of the model which will be learned from the data using the procedure outlined in Section 2.3 below. The number of first-layer units was fixed to $n^{(1)} = 600$. A linear stage followed by rectification is a simple model for the steady-state firing rate of neurons (Dayan and Abbott, 2001, Chapter 7.2). In this model, the features $\mathbf{w}_i^{(1)}$ correspond to the receptive fields of the neurons.

Based on the symmetry of natural images (see, for example, Gutmann and Hyvärinen, 2012b, Section 5.4), we make the simplifying assumption

that for each receptive field $\mathbf{w}_i^{(1)}$, there exists a receptive field $\mathbf{w}_{i'}^{(1)}$ with a sign-inverted spatial pattern, that is $\mathbf{w}_{i'}^{(1)} = -\mathbf{w}_i^{(1)}$. We also assume that the weights in the second layer (see below) are the same for $y_i^{(1)}$ and $y_{i'}^{(1)}$. This assumption reduces the number of free parameters, and we can compute the first layer outputs as $y_i^{(1)} = \mathbf{w}_i^{(1)} \cdot \mathbf{x}$, for $i = 1 \ldots n^{(1)}/2 = 300$.

Second layer. After elementwise squaring, the outputs $\mathbf{y}^{(1)} = (y_1^{(1)}, \ldots, y_{n^{(1)}}^{(1)})$ from the first layer are projected onto second-layer features $\mathbf{w}_i^{(2)}$. The outputs $y_i^{(2)}$ of the second-layer units are obtained as

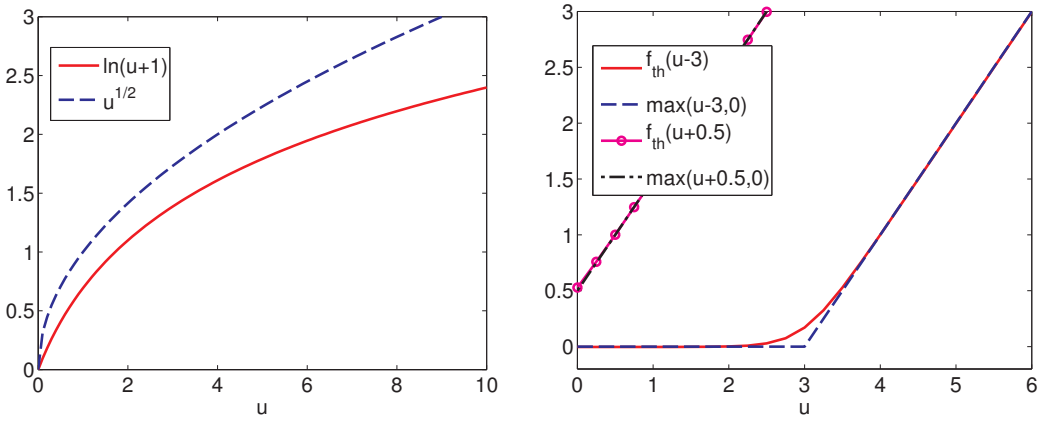
$$y_i^{(2)} = \ln\left(\mathbf{w}_i^{(2)} \cdot (\mathbf{y}^{(1)})^2 + 1\right), \quad i = 1 \ldots n^{(2)}. \tag{2}$$

The number of second-layer units was fixed to $n^{(2)} = 100$. The weight vectors $\mathbf{w}_i^{(2)}$ are, again, parameters that we learn from the data. Each element $w_{ki}^{(2)}$ of a vector $\mathbf{w}_i^{(2)}$ is constrained to be nonnegative. The functional form of (2) corresponds to the energy model for complex cells (Adelson and Bergen, 1985), albeit with receptive fields and pooling patterns that are not yet specified, but to be learned from the data. The nonlinearity $\ln(u+1)$ is concave, which is important for both a practical and a conceptual reason. Practically, a concave nonlinearity keeps the second-layer outputs within a reasonable range (this argument for having such a nonlinearity after the pooling was also given by Adelson and Bergen, 1985). The nonlinearity $\ln(u + 1)$ is shown in Figure 2a. It is qualitatively similar to the square root, but has a smaller slope at zero and grows also more slowly, which makes it more robust. Conceptually, combining a concave nonlinearity with a convex one (the squaring) can be considered to be a mathematical abstraction of the idea of combining a tolerance with a selectivity computation (see Section 4.4 for a discussion of this point).

Intermediate gain control layer. Next, we pass the outputs of the second layer through a gain control stage,

$$\mathbf{z}^{(2)} = \text{gain control}\left(\mathbf{y}^{(2)}\right), \tag{3}$$

which is defined in the same way as on the level of the pixels: The DC value $1/n^{(2)} \sum_i y_i^{(2)}$ is first removed from $\mathbf{y}^{(2)} = (y_1^{(2)}, \ldots, y_{n^{(2)}}^{(2)})$. The resulting centered vector is thereafter whitened and its norm standardized. If some of the $y_i^{(2)}$ are strongly correlated, we reduce the dimension by PCA before



(a) Nonlinearity on the second layer (b) Output nonlinearity in the models

Figure 2: (a) The nonlinearity $\ln(u+1)$ (red solid) used in (2) is qualitatively similar to the square-root (blue dashed). (b) The nonlinearity $f_{\text{th}}(u+b)$ (red solid for $b=-3$ and magenta with circles for $b=1/2$) used in (5) and (6) is a smooth approximation of $\max(u+b,0)$ (blue dashed for $b=-3$ and black dash-dotted for $b=1/2$). We learn the value of b from the data. Negative b result in thresholding.

performing whitening.[2] Just like on the level of the pixels, gain control makes the statistical dependencies between the second-layer outputs more accessible, which is helpful for the learning of the features on the third layer.

Third layer. The third processing layer has the same form as the first one. The outputs $y_i^{(3)}$ of the third-layer units are computed as

$$y_i^{(3)} = \max\left(\mathbf{w}_i^{(3)} \cdot \mathbf{z}^{(2)}, 0\right), \quad i = 1 \ldots n^{(3)}. \tag{4}$$

The third-layer features $\mathbf{w}_i^{(3)}$ are also learned from the data. They are the parameters which we are the most interested in. The number of third-layer units was fixed to $n^{(3)} = 50$. We will call negative elements in a vector $\mathbf{w}_i^{(3)}$ inhibitory weights. If the corresponding element in $\mathbf{z}^{(2)}$ is negative too, disinhibition occurs.

[2]Strong correlations would make the whitening operation nonrobust. As we will see in Section 3.2, dimension reduction was only necessary for patch data.

2.3. Learning the parameters by fitting a probability density function

We learn the parameters which govern the computation of the three layers by fitting a probability density function (pdf) $p_{\mathbf{x}}$ to natural image data. The basic idea is that the overall activity of the feature outputs determines how likely an input image is. Because of the computational complexity of the processing, we first learn the parameters of layer one and two, ignoring layer three. Afterwards, we keep the first two layers fixed, and learn the parameters of layer three.

We have learned the parameters of the first *two* layers for a different kind of natural image data before (Gutmann and Hyvärinen, 2012b, Section 5.3). The data consisted of image patches from the woods. While the patch data here is similar, we can expect different results for the tiny images data. Our previous work contained a detailed description of the general learning principles so here, we will be brief.

Learning the parameters of layer one and two. The pdf is defined as

$$p_{\mathbf{x}}(\mathbf{x}) \propto \prod_{i=1}^{n^{(2)}} \exp\left(f_{\text{th}}\left(y_i^{(2)} + b_i^{(2)}\right)\right), \tag{5}$$

where the coefficient $b_i^{(2)}$ is an an additional parameter that we learn from the data, together with the first- and second-layer features $\mathbf{w}_i^{(1)}$ and $\mathbf{w}_i^{(2)}$. No constraints are imposed on the $b_i^{(2)}$. The function $f_{\text{th}}(u) = 0.25\ln(\cosh(2u)) + 0.5u + 0.17$ is a smooth approximation of the function $\max(u, 0)$, see Figure 2b. The figure also shows that the behavior of $f_{\text{th}}(y_i^{(2)} + b_i^{(2)})$, which takes only non-negative $y_i^{(2)}$ as input, is quite different for negative or positive $b_i^{(2)}$. For negative $b_i^{(2)}$, thresholding occurs, for positive and zero $b_i^{(2)}$, on the other hand, the function resembles an affine transformation.

In case of thresholding, only feature outputs above the threshold effectively contribute to $p_{\mathbf{x}}(\mathbf{x})$. For sufficiently negative $b_i^{(2)}$, an input $\mathbf{x}$ is only assigned a large relative probability $p_{\mathbf{x}}$ if the feature outputs are large, that is, if strong presence of some characteristic image structure is detected. Thresholding is thus related to feature detection (for a more detailed argument, see Gutmann and Hyvärinen, 2012a, Section 5.3).

Note that we do not need to impose any constraints on the feature outputs. The fact that $p_{\mathbf{x}}$ should take large values for likely but small values

for unlikely $\mathbf{x}$ prevents the feature outputs from becoming too large or from staying large all the time.

The probabilistic model in (5) is unnormalized. That is, we do not know the correct normalizing proportionality factor (partition function) for which $p_{\mathbf{x}}$ integrates to one for all parameters. Given the complexity of the model, analytical integration is not possible, and numerical integration is computationally too costly because of the high dimensionality of $\mathbf{x}$. The standard approach of estimating the model by maximizing the likelihood is thus not feasible. We use instead noise-contrastive estimation (Gutmann and Hyvärinen, 2012b) which we recently developed to estimate unnormalized models in a statistically principled, yet computationally feasible way. Its basic idea is to estimate unnormalized models by performing nonlinear logistic regression between the observed data and some artificially generated noise. Here, the observed data are the natural images, and the uniform distribution in the sphere where $\mathbf{x}$ is defined serves as contrastive noise.

Learning the parameters of layer three. Keeping the first two layers fixed, we learn the parameters of the third layer by estimating the pdf

$$p_{\mathbf{x}}(\mathbf{x}) \propto \prod_{i=1}^{n^{(3)}} \exp\left(f_{\text{th}}\left(y_i^{(3)} + b_i^{(3)}\right)\right). \tag{6}$$

Here, f_{th} is the same function as in (5). The coefficients $b_i^{(3)}$ are again learned from the data, together with the third-layer features $\mathbf{w}_i^{(3)}$. As for the first two layers, we use noise-contrastive estimation for the learning of the parameters.

3. Results

Section 3.1 briefly considers the learned parameters which govern the computation in the first two layers. Our main focus is on the learned computation in layer three, which is presented in Section 3.2.

3.1. Computation in layer one and two

The learned parameters $b_i^{(2)}$ are all negative. As pointed out in Section 2.3, the second-layer outputs are thus thresholded in the computation of $p_{\mathbf{x}}$ in (5). If a second-layer output is below the threshold is does not effectively contribute to the value which $p_{\mathbf{x}}$ takes for the input $\mathbf{x}$. For patch data, the individual second-layer units contribute to $p_{\mathbf{x}}$ in $10 \pm 3\%$ of the inputs (average

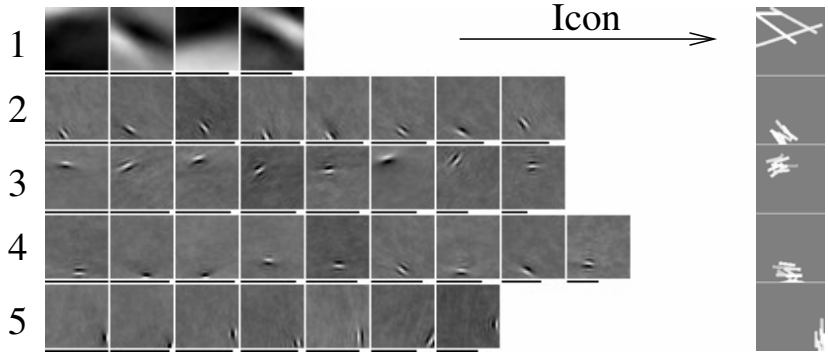
over the units ± standard deviation).[3] For the tiny images, they contribute in $12 \pm 2\%$ of the inputs. These percentages measure the contribution of a single unit for many inputs; they are lifetime percentages. Population percentages are equally interesting; they indicate how many units of the whole population contribute to $p_{\mathbf{x}}$ for a single input. We find that $10 \pm 5\%$ of all units contribute for patch data (average over 10,000 test images ± standard deviation). For the tiny images, it is $12 \pm 5\%$. That is, about 5 to 17 units tend to contribute to the computation of $p_{\mathbf{x}}(\mathbf{x})$. The "active" second-layer units above the threshold signal that some characteristic structure is strongly present in an input image. We next visualize that structure by visualizing the learned features.

Figure 3 shows the learned first and second-layer features in the same way as in previous work of ours (Gutmann and Hyvärinen, 2012b, Figure 12). The results for patch data are shown in Figure 3a; the results for the tiny images in Figure 3b. We only show a small random selection of the learned features. The complete set is shown in Appendix A. After learning, the second-layer weight vectors $\mathbf{w}_i^{(2)}$ are extremely sparse: For patch data 97%, and for tiny images 95% of the elements in the vectors $\mathbf{w}_i^{(2)}$ are smaller than the 10^9-th fraction of their maximal elements. Note that this result was obtained without norm constraints or other measures that impose sparsity on the weight vectors. Because of the high level of sparsity, we visualize a second-layer unit i in Figure 3 by showing the few $\mathbf{w}_k^{(1)}$ for which the corresponding elements $w_{ki}^{(2)}$ of the vector $\mathbf{w}_i^{(2)}$, that is the weights for the $(y_k^{(1)})^2$ in (2), are "nonzero". A first-layer feature $\mathbf{w}_k^{(1)}$ itself is visualized by showing the image which yields the largest value of $y_k^{(1)}$. First-layer features for which the weights $w_{ki}^{(2)}$ are largest are shown first.

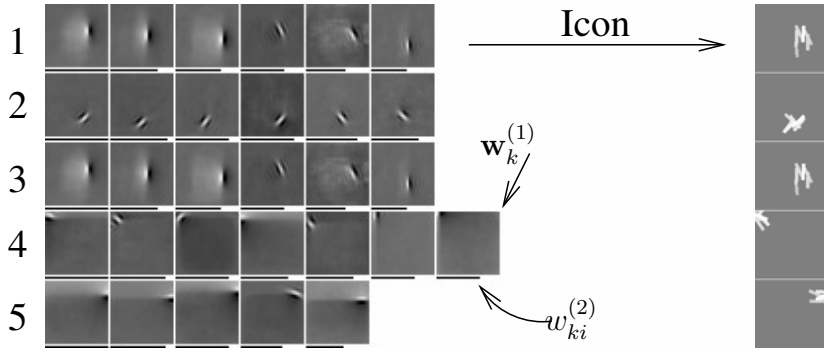
The first processing layer is mostly sensitive to Gabor-like image features, and the second layer pools dominantly over similarly oriented and localized first-layer features. The first layer thus implements a selectivity stage, with the Gabor-like image features being the preferred input. The visualization of all the features in Appendix A shows that the first layer is more often sensitive to low frequency image structure for patch data than for the tiny images.

Figure 3 also shows a condensed visualization of the features by means

[3] We used 10,000 test images to compute the percentages, both for the lifetime and the population measurements.

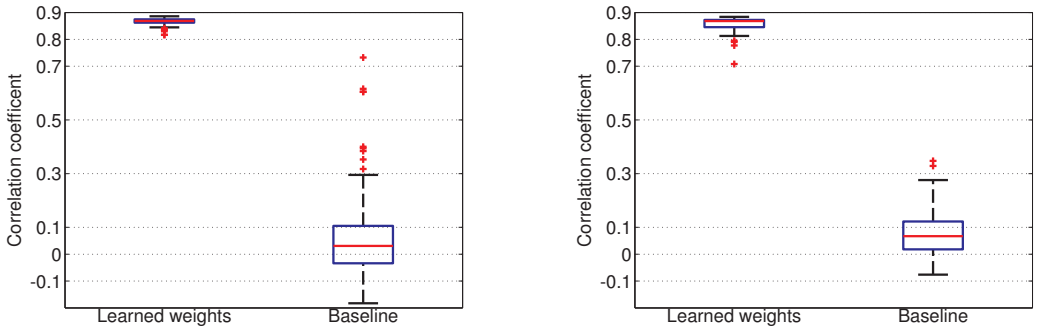


(a) Results for patch data



(b) Results for the tiny images

Figure 3: Visualization of the learned processing on the first two layers. The five rows in (a) and (b) visualize five randomly selected second-layer units learned for patch data and the tiny images, respectively. We visualize second-layer unit i on a certain row by showing the few $\mathbf{w}_k^{(1)}$ for which the elements $w_{ki}^{(2)}$ of $\mathbf{w}_i^{(2)}$, that is the weights for the $(y_k^{(1)})^2$ in (2), are "nonzero". The first-layer features $\mathbf{w}_k^{(1)}$ are visualized by showing the optimal stimulus. The nonzero weights $w_{ki}^{(2)}$ are visualized by the lengths of the black horizontal bars under the optimal stimuli. The first processing layer is mostly sensitive to Gabor-like image features, and the second layer pools dominantly over similarly oriented and localized first-layer features. The icons of the second-layer units will be used for the visualization of the third-layer units.



(a) Results for patch data

(b) Results for the tiny images

Figure 4: The figures show the distribution of the correlation coefficients between second-layer outputs and maximal first-layer outputs, both for learned second-layer weights and random weights to obtain a baseline. Comparison with the baseline shows that the strong correlation for the learned weights is due to the adaptation to the natural images. The learned pooling in the second layer gives rise to a max-like computation over selected first-layer outputs, with the selection being learned from the data.

of icons that we have created by representing each Gabor-like feature $\mathbf{w}_k^{(1)}$ by a bar and superimposing them weighted by $w_{ki}^{(2)}$ (like in Gutmann and Hyvärinen, 2012b, Figure 12). Below, we use the icons to visualize some properties of the third-layer features. Low-frequency features are, however, not well represented by their icons and hence not used in this third-layer visualization.

We next show that the processing on the second layer can be interpreted as a max-like computation over selected first-layer feature outputs $|y_k^{(1)}|$ where the selection is learned from the natural images. For each second-layer weight vector $\mathbf{w}_i^{(2)}$, we computed the maximum value over the $y_k^{(1)}$, limiting the maximum operation to those indices k for which the pooling weight $w_{ki}^{(2)}$ was larger than 0.0001, relative to the maximal element of $\mathbf{w}_i^{(2)}$. We thus considered the learned weights $w_{ki}^{(2)}$ as indices that select over which first-layer outputs to take the max operation. We then computed the correlation coefficient between each second-layer output $y_i^{(2)}$ and the obtained maximum value, using natural images as input. Figure 4 shows the distribution of the correlation coefficients that we have obtained for all $y_i^{(2)}$. For both patch data and the tiny images, the correlation coefficients are close to 0.9. To

establish a baseline, we also used random vectors with positive elements that sum to the same value as the learned second-layer vectors. This yields correlation coefficients with median 0.02 for patch data, and 0.07 for tiny images. Hence, the strong correlation obtained for the learned weights is due to the adaptation of the weights to the properties of the natural images; it is not merely a consequence of the assumed functional form of $y_i^{(2)}$ in (2).

3.2. Computation in layer three

Some of the second-layer outputs are strongly correlated for patch data. We thus reduced the output-dimensionality of the gain control layer as discussed in Section 2.2. Finding the right amount of dimension reduction was straightforward since the eigenvalues of the covariance matrix of $\mathbf{y}^{(2)}$ dropped abruptly for the last two dimensions. For tiny images, no dimension reduction was necessary.

Like the learned $b_i^{(2)}$, the learned $b_i^{(3)}$ are all negative: the third-layer outputs are thresholded in the computation of $p_{\mathbf{x}}$ in (6). We computed how often the third-layer outputs are above the threshold. As for the second-layer results, we performed lifetime and population measurements. Regarding the lifetime measurements, we find that, for patch data, the individual third-layer units are in $23 \pm 8\%$ of the inputs above the threshold. For the tiny images, it happens $32 \pm 12\%$. Regarding the population measurements, we find similar results: $23 \pm 6\%$ for patch data and $32 \pm 5\%$ for the tiny images. Since we had fixed $n^{(3)}$ to 50, about 8 to 15 and 13 to 19 third-layer outputs tend to contribute to $p_{\mathbf{x}}(\mathbf{x})$ for patch data and the tiny images, respectively. Thus, compared to the second layer, the percentages are larger on the third layer, the absolute number of "active" feature outputs, on the other hand, is similar, even though still a bit larger on the third layer.

In Figure 5 and Figure 6, we visualize selected third-layer features that emerged for the patch data and the tiny images, respectively. The complete set of features is visualized in Appendix B in the same way. Each row corresponds to a single third-layer unit. The figures have four columns. The leftmost (panels with black frames) contains space-orientation receptive fields, which visualize the response to local gratings of different orientations (similarly to what has been done by Anzai et al., 2007). These receptive fields show the space-dependent orientation tuning of the third-layer units. The second column contains inhibitory space-orientation receptive fields (panels with red frames). These show the location and orientation of local gratings

which inhibit the units most. The third column visualizes the activity patterns of layer two which lead to the largest outputs $y_i^{(3)}$ in the third layer. The fourth column shows examples of natural images for which the third-layer outputs $y_i^{(3)}$ are large.

Space-orientation receptive fields. The space-orientation receptive fields are shown in the first column of Figure 5 and Figure 6. We constructed them by probing the third-layer units with Gabor stimuli (localized gratings) of different frequency, location, and orientation. The tested spatial frequencies were $0.1, 0.15, 0.2$, and 0.25 cycles per pixel. The Gabor stimuli were fixed to have an aspect ratio of one, and a frequency bandwidth of 1.4 octaves (full width at half response), which is a typical value for simple cells in the cat or macaque monkey (Daugman, 1985). The standard deviation σ of the Gaussian window which underlies a Gabor stimulus thus depends on the fixed frequency bandwidth and the frequency itself. The test locations formed a grid with a spacing of about 2σ. We use circles to indicate the locations of the test stimuli in each receptive field. Their radius is σ, and they contain about 68% of the total mass of the Gaussian window. The spacing was thus frequency-dependent: it is more narrow for high-frequency and wider for low-frequency stimuli. For each third-layer unit, we only show the space-orientation receptive field of the spatial frequency which elicited the largest response.

Figures 5 and 6 show that the space-orientation receptive fields are well structured: areas where the units on the third layer are susceptible to stimulation are mostly contiguous, and often localized. These areas form the classical receptive field. Visual inspection shows that for both kinds of data, homogeneous and inhomogeneous receptive fields have emerged, albeit, for tiny images, inhomogeneous receptive fields seem to occur more often. For inhomogeneous receptive fields, the preferred orientation varies across the receptive field; for homogeneous receptive fields, the preferred orientation does not change with location. Localized homogeneous receptive fields resemble longer straight contours (as for example unit 46 in Figure 5). We next quantify homogeneity and orientation tuning on population level.

Regarding orientation tuning, we computed the preferred orientation for all units, that is, we computed the orientation of the local grating that results in the largest response, for each unit. The corresponding histogram is shown in Figure 7a. For tiny images, horizontally and vertically oriented local gratings often yield the largest response (red, circles). The preferred

orientation for patch data, on the other hand, is more uniformly distributed (blue, asterisks). The horizontal preference is, however, still the dominant one.

We quantified the level of homogeneity by investigating the difference in orientation tuning within the receptive fields. Locations (the circle-centers) in the receptive field where the response was less than a certain fraction r of the maximal response were excluded from the analysis because for small responses, the preferred orientation cannot be computed reliably. Figure 7b shows cumulative distribution functions for the difference in orientation tuning, for $r \in \{0.25, 0.5, 0.75\}$. The difference was computed for all possible pairs of locations, and pooled across the population. Whatever the value of r, the cumulative distribution functions for the tiny images (curves in red) tend to increase more slowly than those for the patch data (curves in blue). This means that, within a receptive field, large differences in preferred orientation are more probable for tiny images than for patch data. The figure illustrates also the influence of r: retaining only locations in the receptive field where the response is large (large value of r), makes the occurrence of large differences in preferred orientation less probable. This applies both to the results for patch and tiny images data.

An alternative analysis of homogeneity consists in showing the distribution of the maximal difference in orientation tuning within a receptive field. We show the results for this kind of analysis in Figure C.14 in Appendix C. If only locations where the response is large are included (case of $r = 0.75$), we find that about 70% of the units have a maximal difference of less than 30°, both for patch and tiny images data. For tiny images, about 20% of the units have a maximal difference of more than 60°; for patch data, the number drops to about 10%. If locations with smaller responses are included in the analysis (smaller values of r), the maximal difference tends to get larger. Nevertheless, whatever measure used, we find that the receptive fields are more inhomogeneous for tiny images than for patch data.

Inhibitory space-orientation receptive fields. The receptive fields shown in the first column of Figures 5 and 6 are based on the third-layer outputs $y_i^{(3)} = \max(\mathbf{w}_i^{(3)} \cdot \mathbf{z}^{(2)}, 0)$, defined in (4). The inhibitory space-orientation receptive fields shown in the second column are, in contrast, based on the outputs $-\min(\mathbf{w}_i^{(3)} \cdot \mathbf{z}^{(2)}, 0)$. They show the orientation and location of local gratings which inhibit the units most. For better comparison, we computed the inhibitory receptive fields for the same frequency band as the receptive

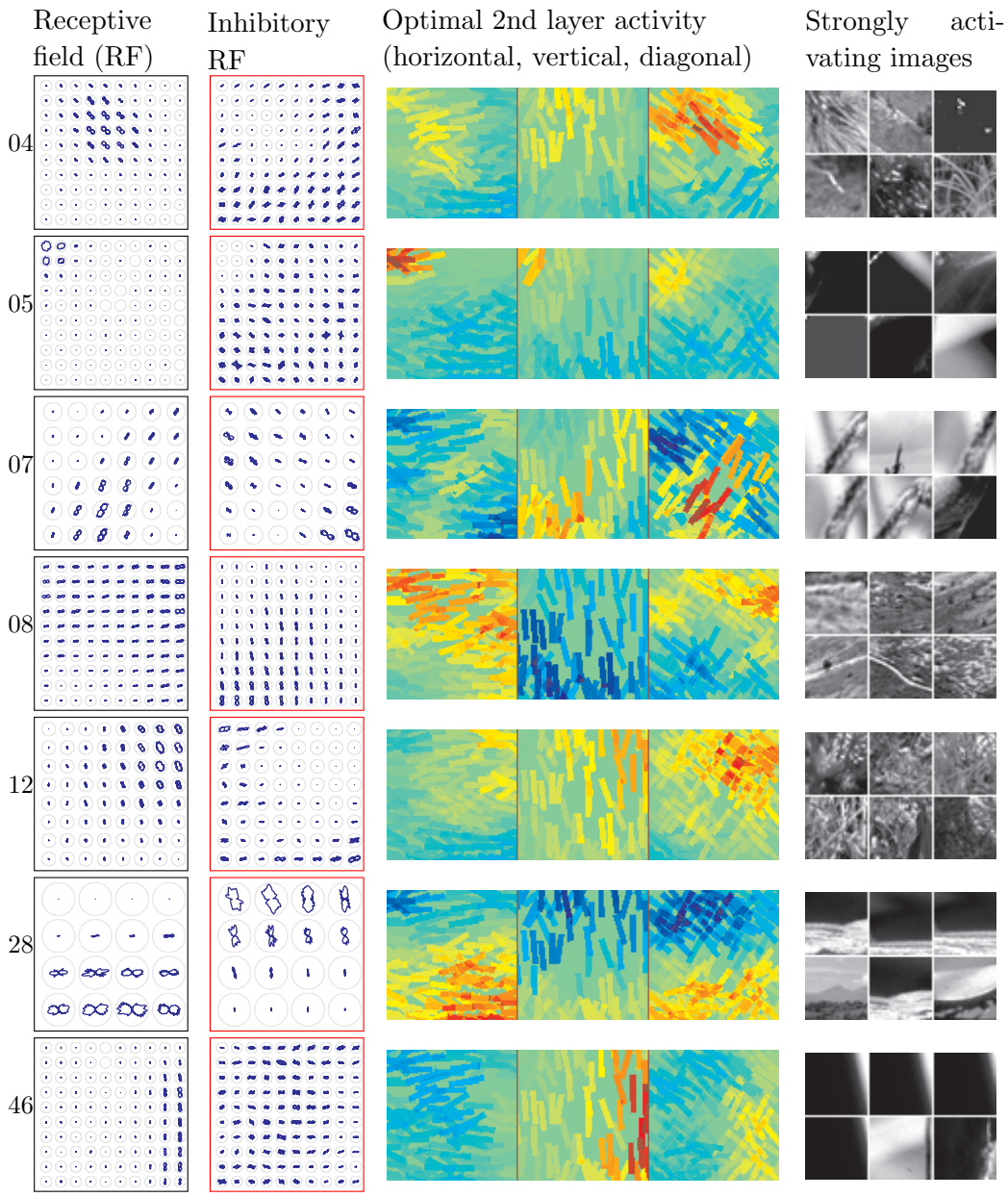


Figure 5: Third-layer features for patch data. The numbers on the left label the features. Each row visualizes one third-layer unit. The complete set of the learned features is shown in Appendix B. See main text for explanation of the visualization used.

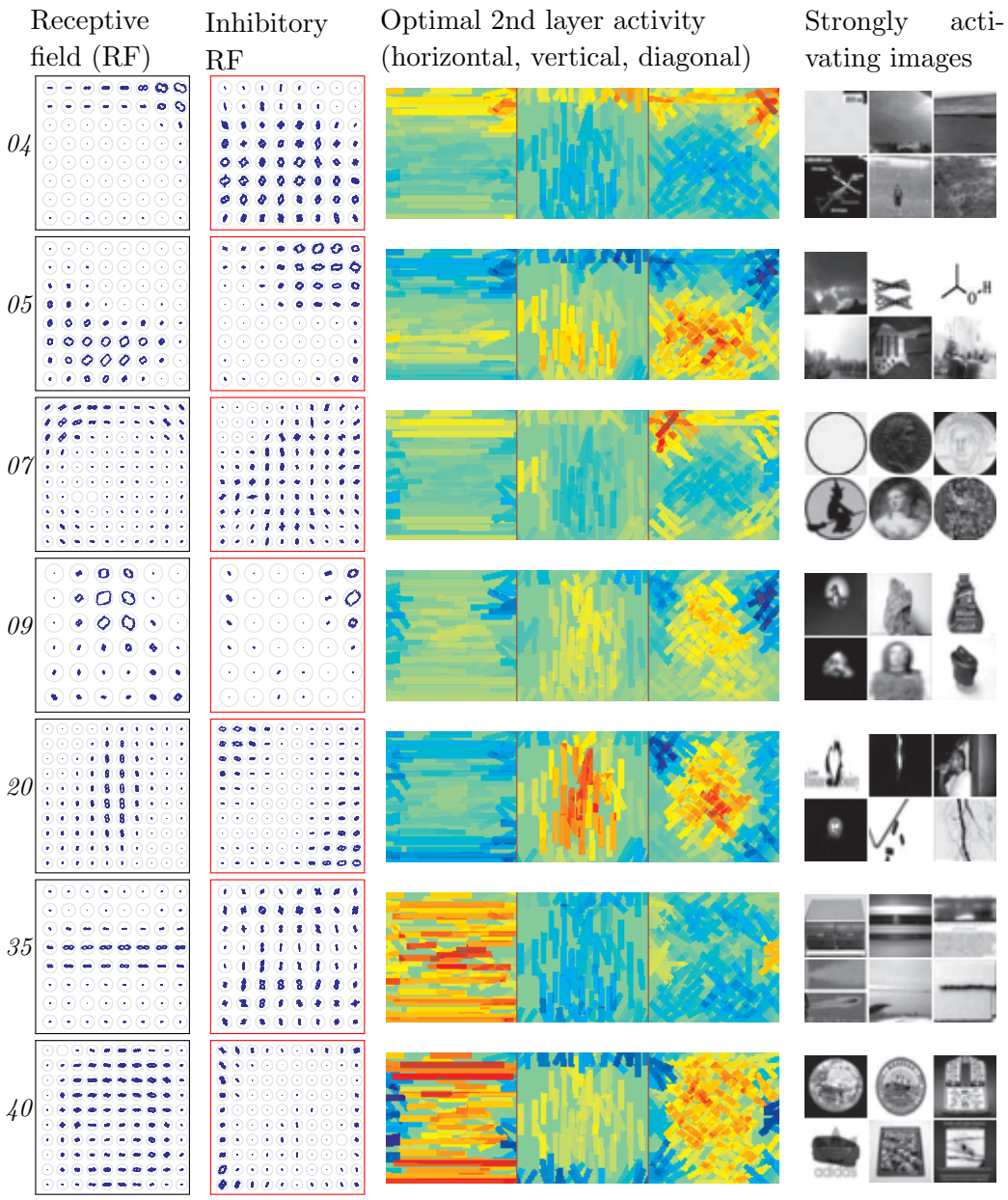
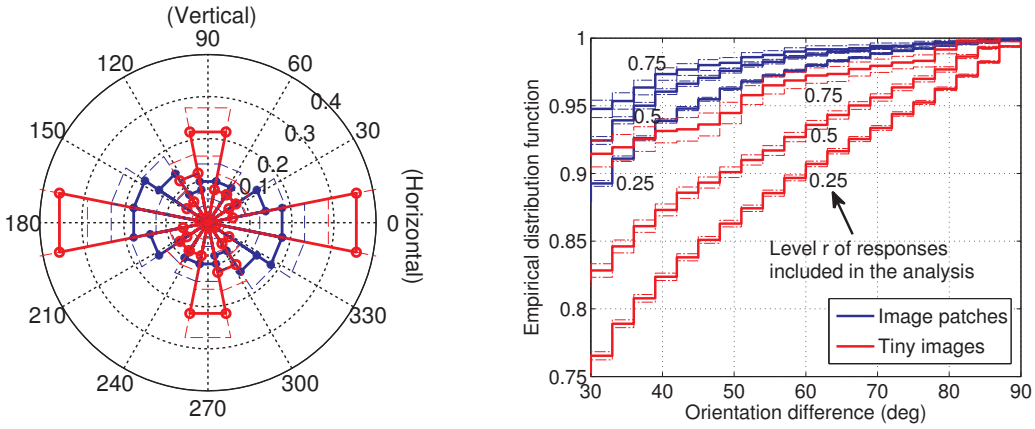


Figure 6: Third-layer features for tiny images. The numbers on the left label the features. Each row visualizes one third-layer unit. The complete set of the learned features is shown in Appendix B. See main text for explanation of the visualization used.

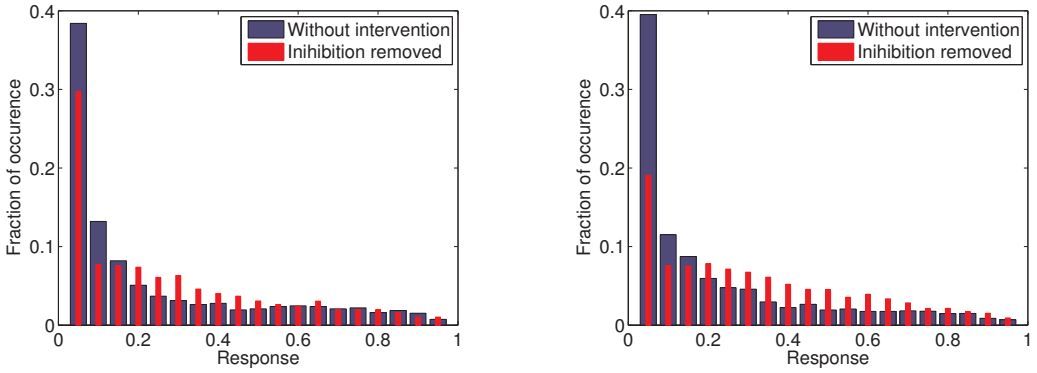


(a) Orientation tuning (b) Homogeneity

Figure 7: Orientation tuning and homogeneity of the population of third-layer units. Curves in blue correspond to results for patch data, red to tiny images. (a) The circular histograms of the preferred orientation of the third-layer units. The angle corresponds to the orientation, the radius to the fraction of occurrence. Horizontal orientation tends to be the preferred one. (b) The empirical distribution functions for the difference in orientation tuning within a receptive field. Tiny images tend to give more often inhomogeneous features. Standard errors are shown as finer dashed lines.

fields in the first column.

The inhibitory receptive fields show that local gratings placed outside the classical receptive field often have an inhibitory effect. In fact, the inhibitory receptive fields appear "spatially complementary" to the actual receptive fields. This structure of the inhibitory receptive fields enhances localization and orientation tuning. We next investigated the role of inhibition on the population level. We computed space-orientation receptive fields after having removed inhibitory connections (negative elements in the weight vectors $\mathbf{w}_i^{(3)}$ were set to zero). With this intervention, the maximal response to the local gratings drops for the tiny image data by $40 \pm 12\%$ (median $\pm$ median absolute deviation from the median). For patch data, the drop is $41 \pm 9\%$. The drop in the maximal response is due to disinhibition effects. Figure 8 shows the distribution of the responses within a receptive field, before (blue) and after (red) the intervention. The responses are computed relative to the maximal response per receptive field. The figure shows that removing inhi-



(a) Results for patch data

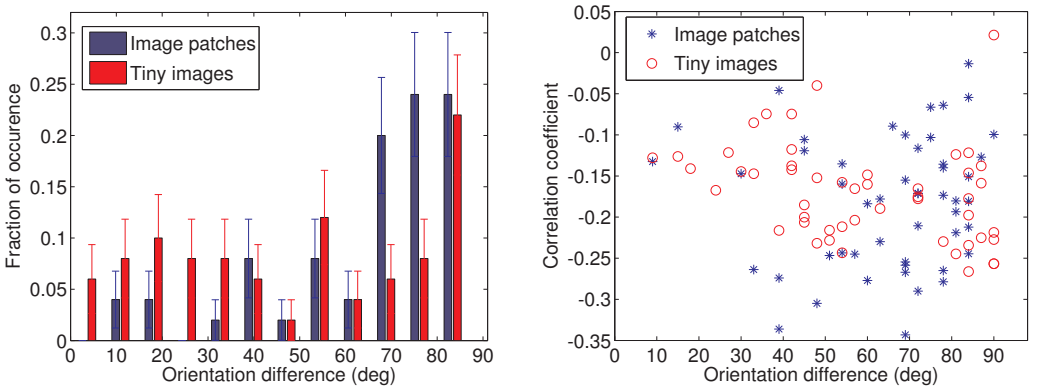
(b) Results for the tiny images

Figure 8: Histogram for the largest response at the test locations within a third-layer receptive field (RF). Blue: initial result with inhibitory connections. Red: result with inhibitory weights set to zero. For each RF and condition ("blue" or "red"), the maximal response was normalized to one. Removing inhibition reduces the fraction of "zero" responses. The distribution becomes more uniform, which indicates a loss of localization within the RFs.

bition reduces the fraction of small responses. This means that the receptive fields become less localized if inhibition is removed.

Visual inspection of the inhibitory receptive fields suggest that strongest inhibition often occurs for local gratings which are orthogonally oriented to the preferred orientation. In order to quantify this observation on population level, we computed the angle between the preferred and the least preferred orientation for all receptive fields. The least preferred orientation was defined as the orientation of the grating yielding the strongest inhibition. Figure 9a shows the resulting histogram for patch data (blue) and tiny images (red). With the largest angle being 90°, the mode of the distribution was for both data sets at $83° \pm 7°$.

Optimal stimuli. The space-orientation receptive fields represent the sensitivity of the third-layer units to local gratings. In order to investigate the sensitivity of each unit to nonlocal stimuli, we computed the outputs $y_i^{(2)}$ of the second layer which together give rise to the largest response on the third layer. Because of the gain control in (3), this optimal activation pattern of



(a) Orientation difference histogram (b) Correlation vs orientation difference

Figure 9: (a) The difference between the most and least preferred orientation was computed using the (inhibitory) receptive fields of the third-layer units. (b) The correlation and angle between the most and least activated second-layer unit, computed for the activation pattern giving rise to the largest third-layer response. We find that the two second-layer units tend to form large angles, that they are negatively correlated for natural image input, and that the third-layer units pool them with weights of opposite signs.

layer two is defined up to a scaling constant and an additive offset. We complemented this analysis by finding natural images which yielded particularly large responses for a given third-layer unit.

The optimal activation patterns of layer two are shown in the third column in Figures 5 and 6 for patch data and tiny images, respectively. We visualize the optimal activation patterns by coloring the second-layer icons in Figure 3 as follows. Second-layer units that are more activated than the DC value are visualized in shades of yellow to red, while units that are less activated than the DC value are colored in light to dark blue. Red corresponds to the largest activation, dark blue to the smallest. Green colored units have an activation around the DC value. In order to make the plots more legible, we separately show the colored icons for second-layer units tuned to horizontally, vertically, and diagonally oriented stimuli.

The six natural images yielding the largest responses among 10,000 randomly selected data points are shown in the fourth column of the figures (in descending order, row-wise from left to right). For each unit, the smallest response, obtained for the sixth image, was still larger than 65% of the max-

imal response. Each image was independently scaled in order to use the full colormap.

Visual inspection of the optimal stimuli suggest that for patch data, the third layer units respond prominently to longer straight contours (for example unit 7 or 46) or to texture (for example unit 8 or 12). The boundaries of the contours can be defined in terms of texture or luminance. For the tiny images, the third-layer units encode longer straight contours, too. In addition, they are sensitive to curvatures (for example unit 5 and 7) or triangular shapes (for example unit 9). The properties of some units likely reflect the fact that in complete visual scenes, center and surround have distinct characteristics (for example unit 40).

We noted above that gratings which are orthogonal to the preferred orientation of the third-layer units often produce strong inhibition. We relate here this orientation inhibition to the statistics of natural images. In particular, we show that there is a connection between the different tuning properties of the second-layer units, the statistical dependencies between them, and the way they are combined by the third-layer units: For each optimal second-layer activation pattern, we identified the most and the least active second-layer unit, and computed both the difference in preferred orientation and the correlation coefficient between the two units (the correlation was computed for natural image input). Figure 9b shows the resulting scatter plot for patch data (blue asterisks) and the tiny images (red circles). In line with Figure 9a, (close to) perpendicular angles are the most probable ones. The scatter plot shows that the two second-layer units are negatively correlated. For the case of tiny images, as the angle between the two units increases, the units become more negatively correlated (p-value of 0.02). For the patch data, the collected data does not give evidence for a linear relation between orientation difference and correlation coefficient.

4. Discussion

We start with Section 4.1 where we discuss our probabilistic approach to learn the computations. In Section 4.2, we discuss then the computations that we have learned for each layer. In Section 4.3, we explore the differences between the results for patch and tiny images data. Section 4.4 is on possible modifications and extensions of our model.

4.1. Learning computations by fitting a probability density function

In this paper, we estimated probabilistic models of natural images. That is, we learned the parameters of our models such that the resulting probability density functions represent the relative probability of natural image structure as correctly as possible. We discuss here this learning scheme and contrast it with other approaches.

The basic reason why we construct and estimate probabilistic models of natural images is the theory of Bayesian perception, where the brain is assumed to use a model of the environment to interpret incoming sensory signals. Since sensory signals are often ambiguous, such an interpretation is useful in order to respond appropriately or to make proper predictions. Bayesian perception, together with Darwinian arguments of evolution, asserts that the visual system is highly adapted to its sensory environment. Due to the adaptation, it has knowledge about what in our environment is likely and what not. Mathematically, this knowledge corresponds to knowledge about the probability density function (pdf) of the natural stimuli. The pdf serves as prior information in Bayesian inference, which is optimal from a normative viewpoint. The fact that the optimal inference depends on the prior is the reason why we focus on modeling and estimating the prior from the data. In other words, the computations learned in this paper are assumed to occur whenever our model is used as a prior in Bayesian inference.

In this paper, learning means fitting a probabilistic model. Other learning approaches have also been used in computational neuroscience or computer vision. Biologically plausible learning rules such as Hebbian learning, together with some stabilizing mechanisms, are often used in computational neuroscience (see, for example, Miller et al., 1989; Turrigiano and Nelson, 2000; Bednar, 2012). This learning has its roots in experimental findings while our learning has its roots in a normative theory (this point was also discussed by Bednar, 2012, Section 4). In computer vision, learning is guided by performance in some target applications instead of theoretical considerations per se (see, for example, LeCun et al., 2006; Bengio, 2009).

A related but different normative approach is sparse coding, which says that strong neural activity should be a relatively rare event (for an introduction, see, for example, Hyvärinen et al., 2009, Chapter 6). Sparse coding is a rather broad concept (see, for example, Willmore et al., 2011, Figure 2). Among other possibilities, it can refer to the situation where a neuron is activated by only a few stimuli (lifetime sparsity), or to the case where only a few neurons inside a larger population are activated by any single

stimulus (population sparsity). Using similarly lifetime and population measurements, we analyzed how often the learned features effectively contribute to the pdf. We found that the features relatively rarely take values above the learned thresholds, meaning that their contribution to the pdf is rather sparse, both in terms of lifetime and population sparsity.[4]

In Appendix D, we analyze the sparsity of the feature outputs before thresholding. In particular, we compare the levels of sparsity in the three different layers. We focus here on lifetime sparsity; the comparison of population sparsity is more intricate because of the smaller sample sizes and because of the fact that the number of units is different in the three layers, see Appendix D for a further discussion of this point. For lifetime sparsity, on average, we found that the feature outputs in the first and third layer are equally sparse, while those in the second layer are less sparse. This finding relates well to the processing in the different layers: As discussed in Section 4.2 below, in layers one and three, selectivity computations are performed, while in layer two, tolerance computations occur. Intuitively, tolerance entails tuning to a wider range of stimuli, which may explain the reduced sparsity in layer two. The finding also relates to recent experimental work by Willmore et al. (2011) where no evidence for an increase in lifetime sparsity across the visual hierarchy was observed. Further, the finding illustrates that modeling structure of natural images, as done in this paper, is not equivalent to maximizing the (lifetime) sparsity of the feature outputs; they are related but different normative approaches.

4.2. Learning of selectivity and invariance computations

In our three layers of computation, presented in Section 2.2, the units in layer one and three have the same functional form. We thus may see them as instantiations of the same canonical computing unit. The units in the second layer belong to a different class of canonical units.

We learned the parameters which govern the behavior of the canonical units by fitting a probability density function (pdf) to natural images. After learning, the units in the first layer are selective to Gabor-like image features ("simple cells"). In the second layer, similarly oriented and localized first-layer outputs are pooled together ("complex cells"). We showed that the

[4]The percentages indicated in Sections 3.1 and 3.2 are for the binary situation of the features being above or below the threshold. In Appendix D, we show that the percentages are in a straightforward way related to the sparsity indices S_2 and S_3.

processing in the second layer can be interpreted as a max-like computation where the maximum is taken over selected first-layer outputs. Importantly, the selection was learned from the data. Max-like computations can be interpreted as tolerance (invariance) computations (Serre et al., 2005). In the third layer, units which are selective to longer contours emerged. These findings hold qualitatively for both patch data and the tiny images (see below for some differences on the quantitative level). On the third layer, units that are specific to the two different data sets emerged as well: For patch data, these units preferred texture input, for the tiny images, they preferred strong curvatures. In addition to grating stimuli, we used mathematically derived optimal activation patterns as well as sample images to perform the analysis. An interesting further analysis would consist in using texture patterns (El-Shamayleh and Movshon, 2011), or complex shapes (Hegdé and Van Essen, 2000) as input.

These results generalize previous results of ours where the first two processing layers were tuned by hand (Hyvärinen et al., 2005). Learning the computations on *all* three layers is also in contrast to work in the computer vision community where selectivity or tolerance computations are often fixed by the researchers and not learned from the data (see, for example, Serre et al., 2007; Jarrett et al., 2009).

Inhibition sharpened the selectivity of the third-layer units to the orientation and localization of the stimulus. Second-layer outputs which tend to be negatively correlated for natural images were weighted with opposite signs. Thus, inhibition increases the sparsity of the response. Willmore et al. (2010) found that neurons in visual area V2 had a stronger suppressive tuning than neurons in the primary visual cortex. The authors speculated that the strong suppression is related to the higher-order statistics of natural scenes. Our modeling study of natural images seems to confirm their conjecture.

Hierarchically organized processing based on canonical selectivity or invariance units has been proposed as a model for cortical computation (see, for example, Fukushima, 1980; Riesenhuber and Poggio, 1999; Kouh and Poggio, 2008). A fundamental problem is, however, to know what the canonical units should be selective or invariant to on each layer of the hierarchy. Our results suggest that it is possible to learn the selectivity or invariance computations from ecologically valid stimuli – using very few assumptions. Only the general form of the layers and the pdf, as well as the non-negativity of the second-layer weights were assumed. The fact that there are differences in the results for natural image patches and tiny images (see below), and that

random weights in the second layer do not induce max-like computations suggest that our assumptions impose only very weak constraints.

In line with the theoretical developments above, it has been empirically shown that visual processing, on population level, becomes gradually more selective and invariant along the hierarchy (Rust and DiCarlo, 2010). But it is currently unknown what kind of single-neuron properties could underlie the observed properties on population level. The computations that we have learned in this paper could yield predictions for possible properties on single-neuron or cell assembly level – with the caveat that our model can only provide functional explanations or predictions of cortical processing. The actual neural implementation is another issue. For the discrepancy between function and implementation in the particular case of simple and complex cells, we refer the reader to the discussion by Ringach (2004).

4.3. Natural image patches versus tiny images

We applied the same statistical model on two different kinds of natural image data: patches extracted from larger images and complete scenes downsampled to patch size ("tiny images"). While comparison of the two data sets was not the primary purpose of this paper, we discuss here some differences that we have observed. We would like to emphasize that despite these differences, qualitatively similar computations emerged for both data sets, in particular on layers one and two (see Section 4.2 above).

For patch data, the first processing layer was more sensitive to low frequency content than for the tiny images. This presumably reflects the fact that an image patch can be an extract of a smooth area in the larger image. Since the tiny images represent complete visual scenes, such data points are more rare in the tiny images data set.

For both patch data and the tiny images, the processing layers are particularly sensitive to horizontal image structure, which is in line with previous findings (Betsch et al., 2004). For tiny images, however, the preference for horizontal structure is much stronger, and vertical structure is dominant too. This can be understood by noting that the tiny images can be considered to be visual scenes seen from a larger distance. Torralba and Oliva (2003) found that, as viewing distance grows, natural structure becomes increasingly biased to horizontal and vertical orientation.

We found that tiny images more often produce inhomogeneous receptive fields on the third layer, compared to image patches. While this may be due

to the limited set of larger images from which we extracted the patches,[5] it makes intuitive sense that complete scenes have stronger variations in orientation than image patches. The degree of inhomogeneity in visual area V2 has been recently investigated for macaque monkeys by Anzai et al. (2007) and Tao et al. (2012). Both studies found similar percentages of homogeneous receptive fields (60-80% had a maximal orientation difference of less than 30° within a receptive field). Concerning strongly inhomogeneous receptive fields with a maximal orientation difference of more than 60°, rather different percentages were found (30% versus 5%). In agreement with the two studies, we found that about 70% of the learned third-layer units have a maximal difference of less than 30° – under the condition that only locations in the receptive field where the response is strong are included. Concerning the receptive fields with a maximal orientation difference of 60° or more, we obtained percentages that lie between the two experimental values, namely 10% for patch data and 20% for the tiny images. However, given the sensitivity of the results to the exact criteria used, such a quantitative comparison should not be taken too seriously. The main thing we can conclude concerning homogeneity is that homogeneous receptive fields emerged more often than strongly inhomogeneous ones, and that inhomogeneous receptive fields emerged more often for tiny images than for patch data.

4.4. Nonlinearities and model definition

Only the squared outputs of the first layer appear in our model, see Section 2.2. We could have subsumed the squaring into the definition of $y_i^{(1)}$ in (1) without changing the model. The processing in the first layer would have been a dot-product between the input and a feature vector followed by a convex nonlinearity (the squaring). In the second layer, we would have had a dot-product followed by a concave nonlinearity (the logarithm). Note that this setup would give exactly the same output on the second layer as the current definitions, only the interpretation is different.

A dot-product followed by a convex nonlinearity can be considered to be a mathematical abstraction of a selectivity unit. A dot-product followed by a concave nonlinearity, on the other hand, may be taken as an abstraction of an invariance unit. While such interpretations may not be entirely new,

[5] In preliminary simulations with patches from images in the woods, some strongly inhomogeneous receptive fields emerged.

we are not aware of work where they are explicitly mentioned or where the feature vectors underlying the dot-products are learned from the data. The assignment of the two classes of nonlinearities to the two classes of computational units relates well to their opposing effects on sparsity: as shown in Figure D.15 in Appendix D, squaring increases sparsity while taking the logarithm reduces it.

Our definition of the computation in the third layer does not include a squaring. In additional simulations using squaring (results not shown), we obtained similar features as with our current definition. The results were, however, less robust. The reason for this is presumably that squaring puts too much emphasis on large values, which makes the learning less stable.

For the learning of the third-layer parameters, the pdf $p_{\mathbf{x}}$ was defined on the level of the third-layer outputs only, see (6). It would have been possible to include the outputs from the second layer to obtain a pdf of the form

$$p_{\mathbf{x}} \propto \prod_{i=1}^{n^{(2)}} \exp\left(f_{\text{th}}\left(y_i^{(2)} + b_i^{(2)}\right)\right) \prod_{i=1}^{n^{(3)}} \exp\left(f_{\text{th}}\left(y_i^{(3)} + b_i^{(3)}\right)\right). \tag{7}$$

Moreover, we could have included terms that depend on the DC values and norms which are removed by gain control. While not necessary to obtain meaningful features, such a pdf might be more appropriate if it is used as a prior in Bayesian inference. Moreover, since the thresholding nonlinearities are not optimal (Gutmann and Hyvärinen, 2012b, Section 5.4), for Bayesian inference tasks, it might be a good idea to further refine the pdf by relearning the nonlinearities, leaving the features $\mathbf{w}_i^{(1)}$ to $\mathbf{w}_i^{(3)}$ fixed.

Part of our model is gain control. We removed the DC value from the vector subject to gain control and normalized its norm after whitening. We assumed this fixed form of gain control for simplicity. It would, however, be possible to learn the parameters which govern it, too. In particular, learning which elements to include in the normalization pool would be interesting. This would presumably lead to an improved model (Schwartz and Simoncelli, 2001).

5. Conclusions

Simultaneous selectivity and invariance is an important property of visual systems. It has been proposed that this property emerges when elementary selectivity and tolerance computations are iterated in a hierarchical fashion.

In this paper, we have addressed the fundamental question what to select or tolerate at each layer of the hierarchy. We sought an answer for the first three layers of the hierarchy by learning the computations from natural images. We are hopeful that this approach can be extended to learn the computations in further layers.

Acknowledgements

This work was supported by the Academy of Finland, Computational Science Program LASTU, the Finnish Centre-of-Excellence in Algorithmic Data Analysis, the Finnish Centre-of-Excellence in Inverse Problems Research, and the Finnish Centre-of-Excellence in Computational Inference Research COIN (251170). M.U.G is also grateful to Jukka Corander for supporting this project.

References

Adelson, E., Bergen, J., 1985. Spatiotemporal energy models for the perception of motion. J Opt Soc Am, A 2 (2), 284–299.

Anzai, A., Peng, X., Van Essen, D. C., 2007. Neurons in monkey visual area V2 encode combinations of orientations. Nat Neurosci 10 (10), 1313–1321.

Barlow, H., 2001. Redundancy reduction revisited. Network: Computation in Neural Systems 12 (3), 241–253.

Bednar, J., 2012. Building a mechanistic model of the development and function of the primary visual cortex. Journal of Physiology-Paris 106 (5-6), 194–211.

Bengio, Y., 2009. Learning deep architectures for AI. Foundations and Trends in Machine Learning 2 (1), 1–127.

Betsch, B. Y., Einhäuser, W., Körding, K. P., König, P., 2004. The world from a cat's perspective – statistics of natural videos. Biol. Cybern. 90 (90), 41–50.

Cadieu, C. F., Olshausen, B. A., 2012. Learning intermediate-level representations of form and motion from natural movies. Neural Computation 24 (4), 827–866.

Daugman, J., 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Am. A 2 (7), 1160–1169.

Dayan, P., Abbott, L., 2001. Theoretical Neuroscience. The MIT Press.

DiCarlo, J. J., Cox, D. D., 2007. Untangling invariant object recognition. Trends in Cognitive Sciences 11 (8), 333–341.

El-Shamayleh, Y., Movshon, J. A., 2011. Neuronal responses to texture-defined form in macaque visual area V2. The Journal of Neuroscience 31 (23), 8543–8555.

Freeman, J., Simoncelli, E. P., 2011. Metamers of the ventral stream. Nat Neurosci 14 (9), 1195–1201.

Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36 (4), 193–202.

Gutmann, M. U., Hyvärinen, A., 2012a. Learning a selectivity–invariance–selectivity feature extraction architecture for images. In: 21st International Conference on Pattern Recognition (ICPR). pp. 918–921.

Gutmann, M. U., Hyvärinen, A., 2012b. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. Journal of Machine Learning Research 13, 307–361.

Hegdé, J., Van Essen, D., 2000. Selectivity for complex shapes in primate visual area V2. The Journal of Neuroscience 20 (5), RC61–RC61.

Hoyer, P., 2004. Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research 5, 1457–1469.

Hubel, D., 1995. Eye, Brain, and Vision.
URL `http://hubel.med.harvard.edu/index.html`

Hurley, N., Rickard, S., 2009. Comparing measures of sparsity. IEEE Transactions on Information Theory 55 (10), 4723–4741.

Hyvärinen, A., Gutmann, M., Hoyer, P., 2005. Statistical model of natural stimuli predicts edge-like pooling of spatial frequency channels in V2. BMC Neuroscience 6:12.

Hyvärinen, A., Hurri, J., Hoyer, P., 2009. Natural Image Statistics. Springer.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y., 2009. What is the best multi-stage architecture for object recognition? In: International Conference on Computer Vision (ICCV). pp. 2146–2153.

Karklin, Y., Lewicki, M. S., 2009. Emergence of complex cell properties by learning to generalize in natural scenes. Nature 457 (7225), 83–86.

Köster, U., Hyvärinen, A., 2010. A two-layer model of natural stimuli estimated with score matching. Neural Computation 22 (9), 2308–2333.

Kouh, M., Poggio, T., 2008. A canonical neural circuit for cortical nonlinear operations. Neural Computation 20 (6), 1427–1451.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., 2006. Predicting Structured Data. MIT Press, Ch. A Tutorial on Energy-Based Learning.

Miller, K., Keller, J., Stryker, M., 1989. Ocular dominance column development: analysis and simulation. Science 245 (4918), 605–615.

Olshausen, B., Field, D., 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381 (6583), 607–609.

Osindero, S., Welling, M., Hinton, G. E., 2006. Topographic product models applied to natural scene statistics. Neural Computation 18 (2), 381–414.

Ranzato, M., Hinton, G., 2010. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In: Proc. of Computer Vision and Pattern Recognition Conference (CVPR). pp. 2551–2558.

Riesenhuber, M., Poggio, T., 1999. Hierarchical models of object recognition in cortex. Nature 2 (11), 1019–1025.

Ringach, D. L., 2004. Mapping receptive fields in primary visual cortex. J Physiol 558 (3), 717–728.

Rust, N. C., DiCarlo, J. J., 2010. Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area V4 to IT. J. Neurosci. 30 (39), 12978–12995.

Rust, N. C., Stocker, A. A., 2010. Ambiguity and invariance: two fundamental challenges for visual processing. Current Opinion in Neurobiology 20 (3), 382–388.

Schwartz, O., Simoncelli, E. P., 2001. Natural signal statistics and sensory gain control. Nat Neurosci 4 (8), 819–825.

Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T., 2005. A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Tech. Rep. AI Memo 2005-036, CSAIL-MIT.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T., 2007. Robust object recognition with cortex-like mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3), 411–426.

Simoncelli, E. P., Olshausen, B. A., 2001. Natural image statistics and neural representation. Annual Review of Neuroscience 24 (1), 1193–1216.

Tao, X., Zhang, B., Smith, E. L., Nishimoto, S., Ohzawa, I., Chino, Y. M., 2012. Local sensitivity to stimulus orientation and spatial frequency within the receptive fields of neurons in visual area 2 of macaque monkeys. Journal of Neurophysiology 107 (4), 1094–1110.

Torralba, A., Fergus, R., Freeman, W. T., 2008. 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (11), 1958–1970.

Torralba, A., Oliva, A., 2003. Statistics of natural image categories. Network: Computation in Neural Systems, 391–412.

Turrigiano, G., Nelson, S., 2000. Hebb and homeostasis in neuronal plasticity. Current Opinion in Neurobiology 10 (3), 358–364.

Willmore, B. D. B., Mazer, J. A., Gallant, J. L., 2011. Sparse coding in striate and extrastriate visual cortex. Journal of Neurophysiology 105 (6), 2907–2919.

Willmore, B. D. B., Prenger, R. J., Gallant, J. L., 2010. Neural representation of natural images in visual area V2. J. Neurosci. 30 (6), 2102–2114.

Zoran, D., Weiss, Y., 2009. The "tree-dependent components" of natural scenes are edge filters. In: Advances in Neural Information Processing Systems 22 (NIPS2009). pp. 2340–2348.

Appendix A. Complete set of second-layer features

For patch data, seven of the one hundred second-layer weight vectors converged with learning to small values. For tiny images, this happened to three weight vectors. These vectors were omitted. We show here the remaining complete set of features for layers one and two, visualized as in Figure 3. Figures A.10 and A.11 show the results for patch data; Figures A.12 and A.13 the results for tiny images.

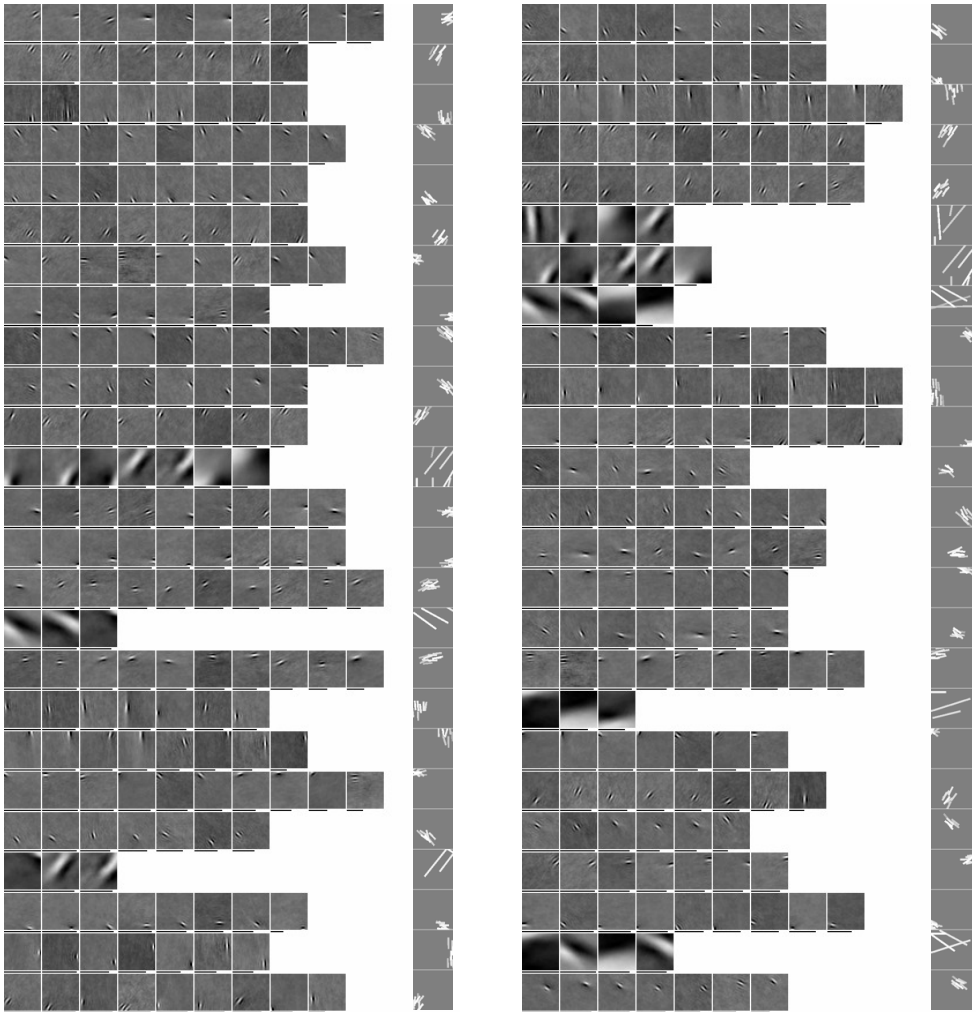


Figure A.10: Patch data: complete set of learned features for layer one and two (part 1), visualized as in Figure 3.

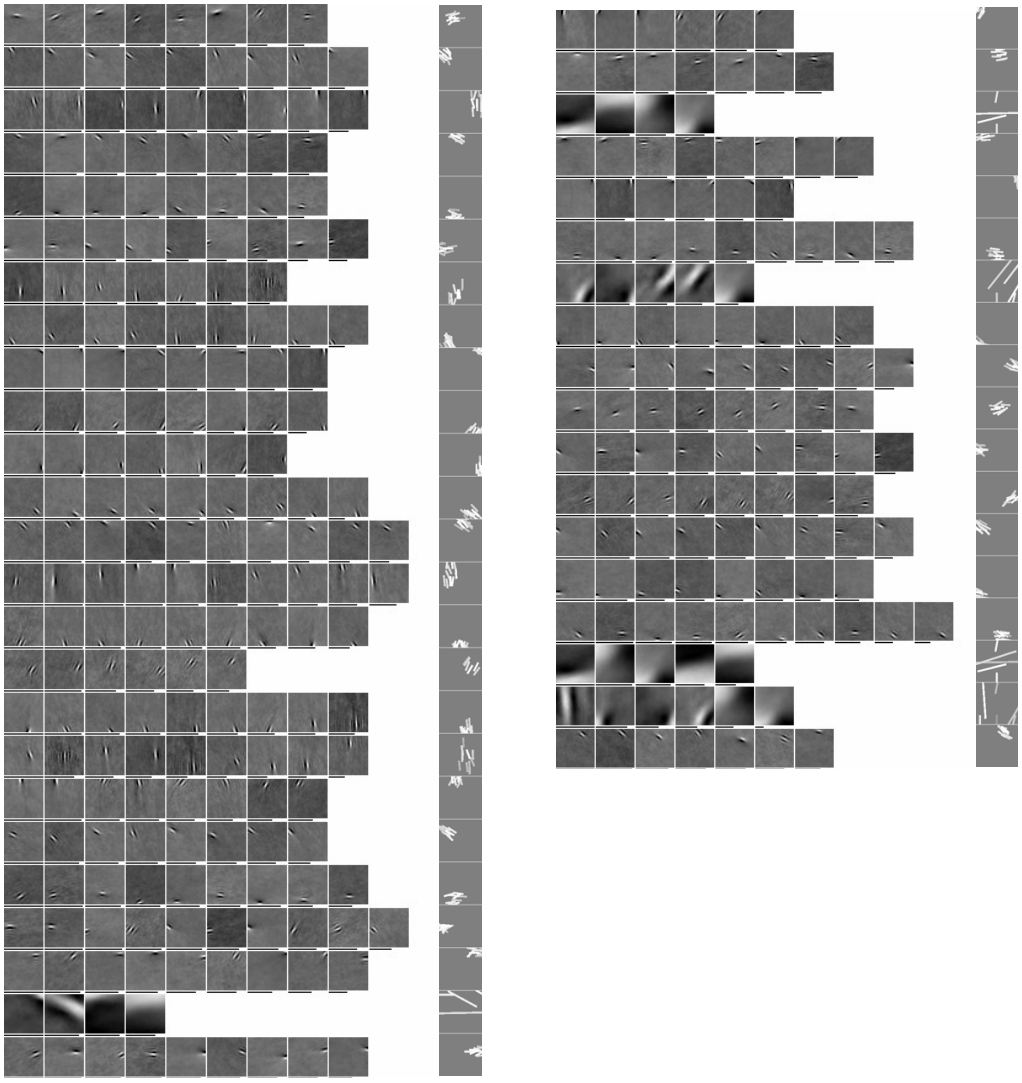


Figure A.11: Patch data: complete set of learned features for layer one and two (part 2), visualized as in Figure 3.

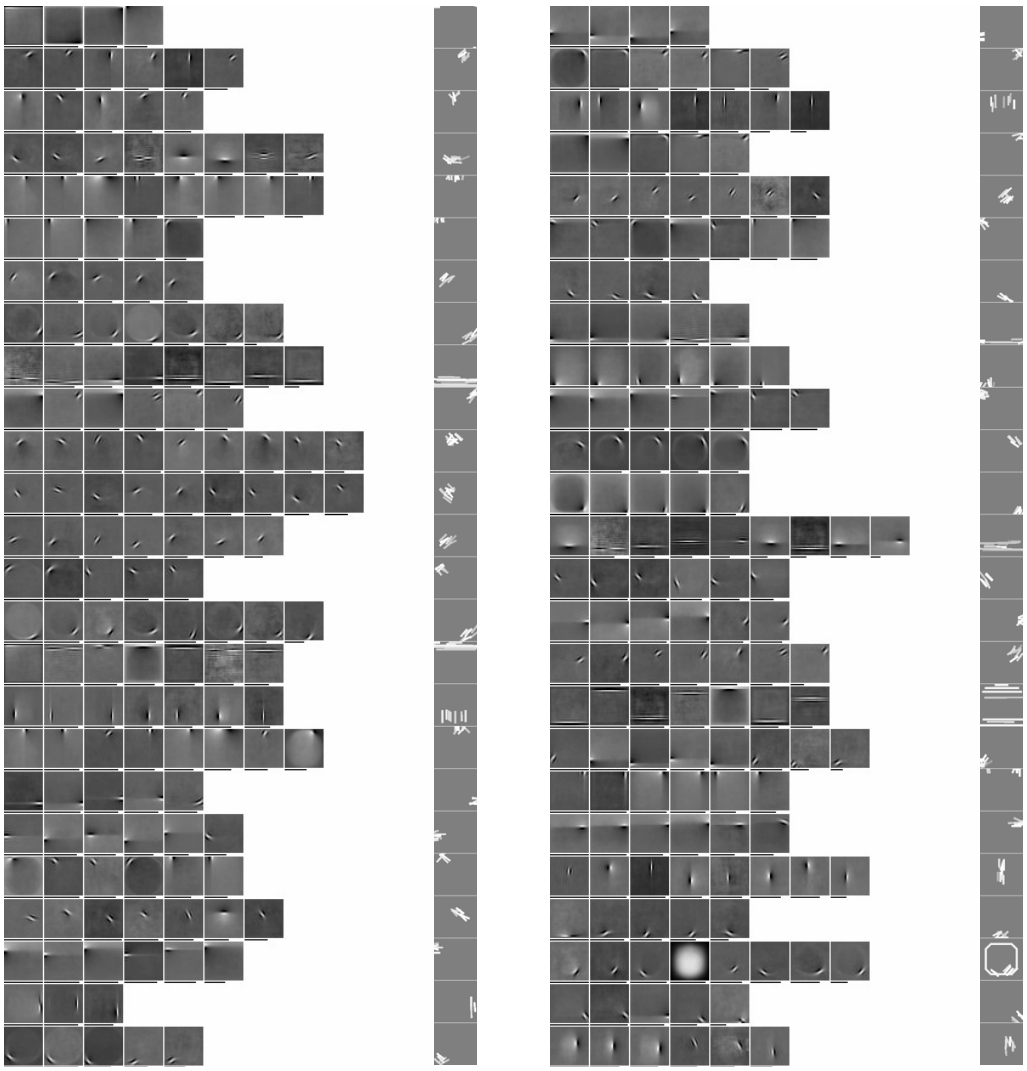


Figure A.12: Tiny images: complete set of learned features for layer one and two (part 1), visualized as in Figure 3.

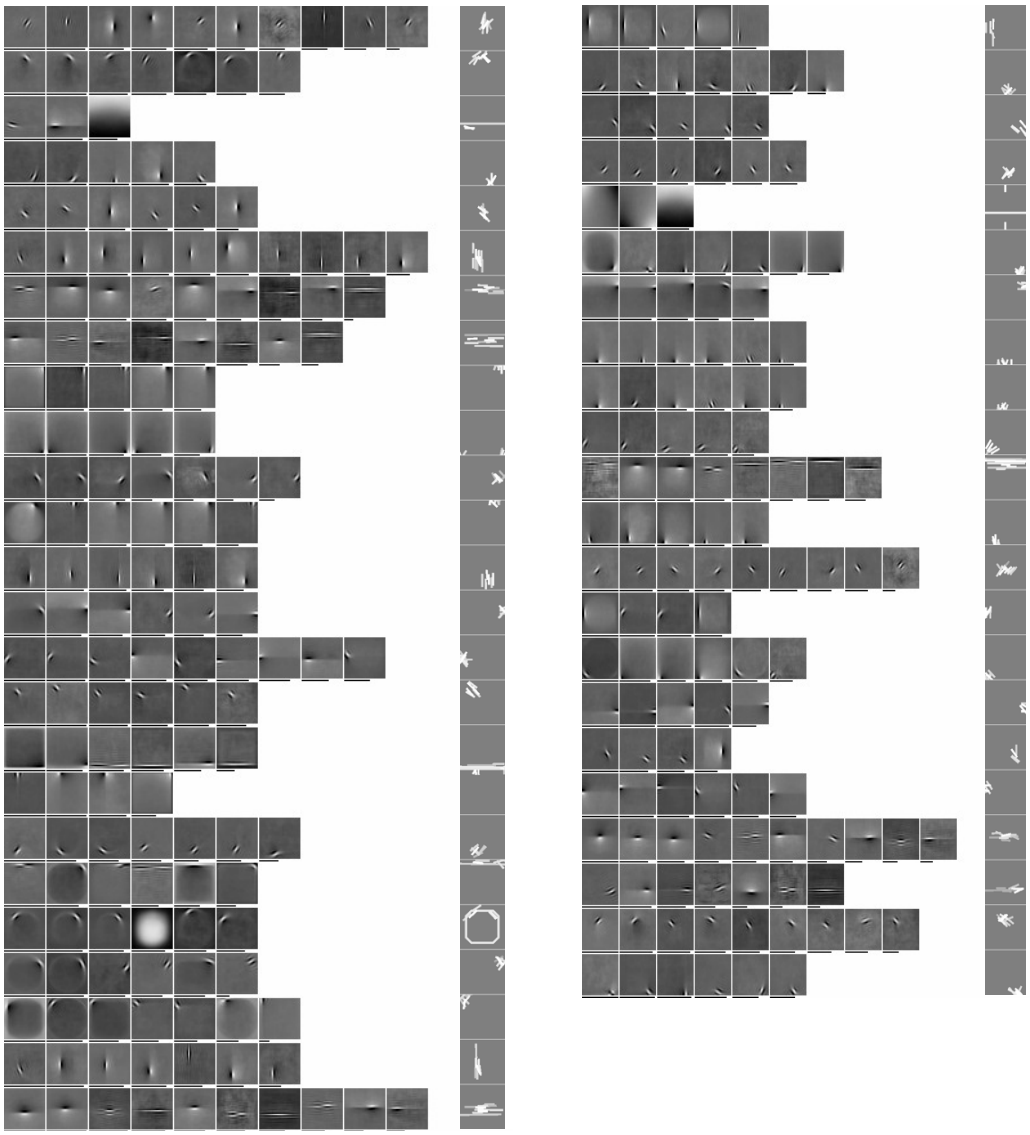
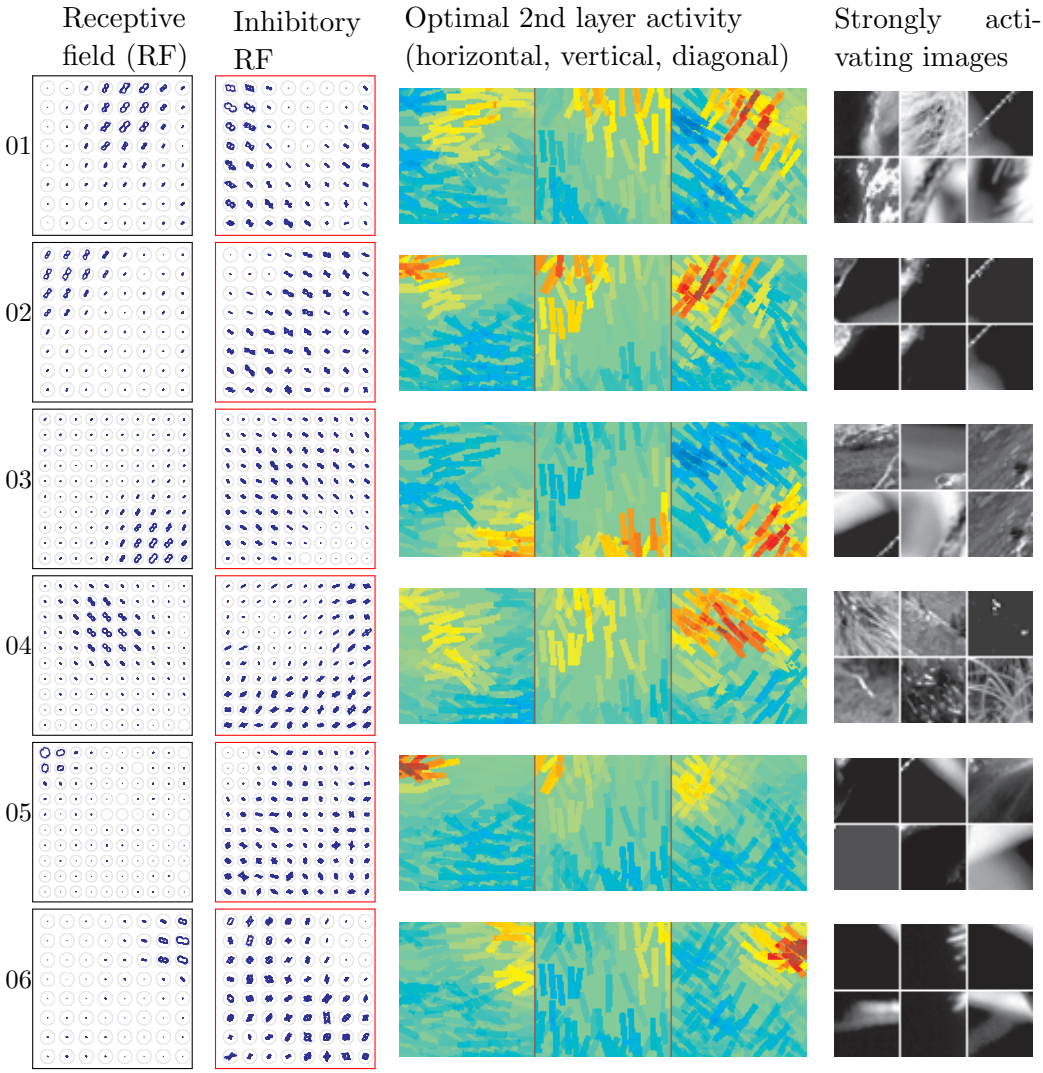


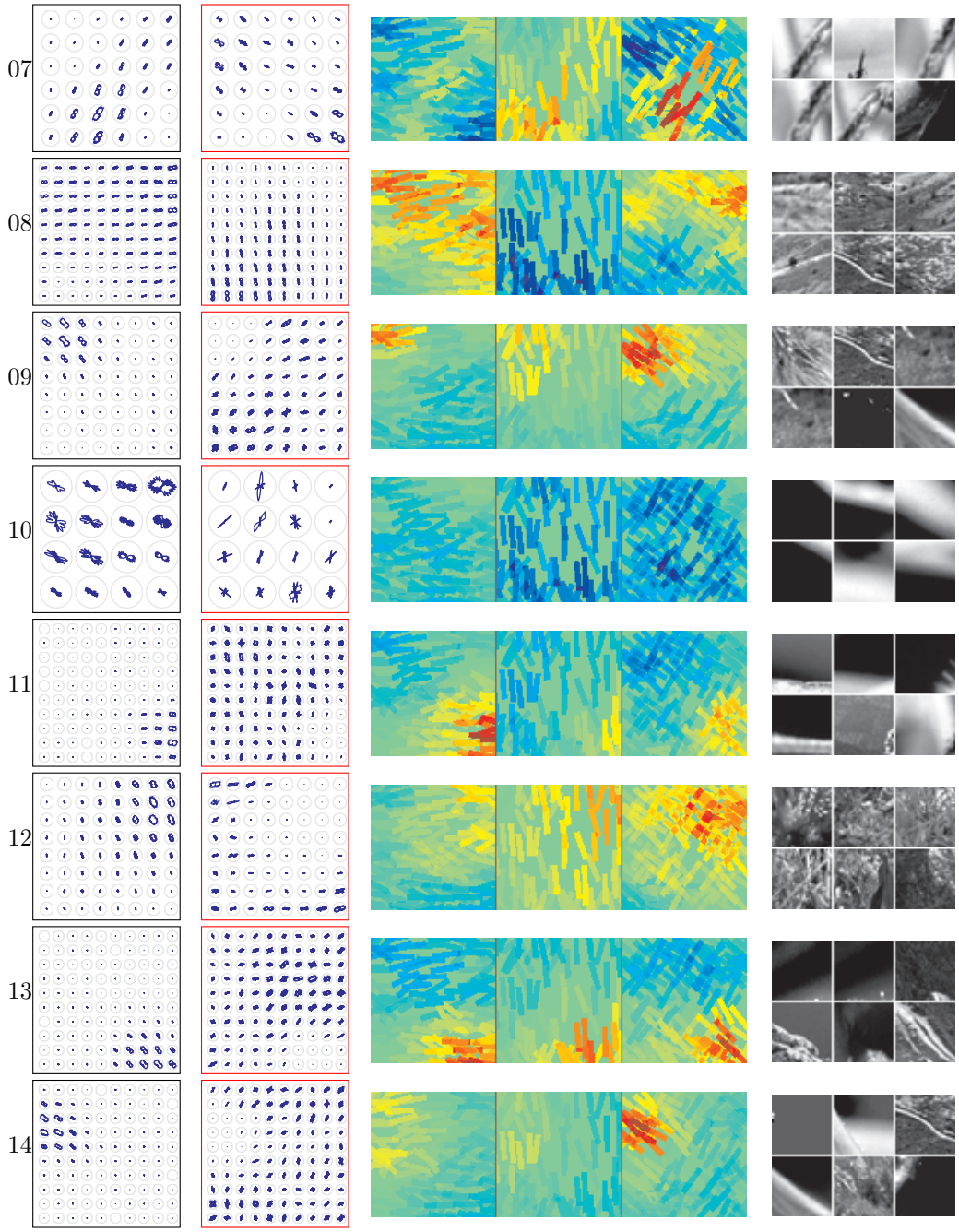
Figure A.13: Tiny images: complete set of learned features for layer one and two (part 2), visualized as in Figure 3.

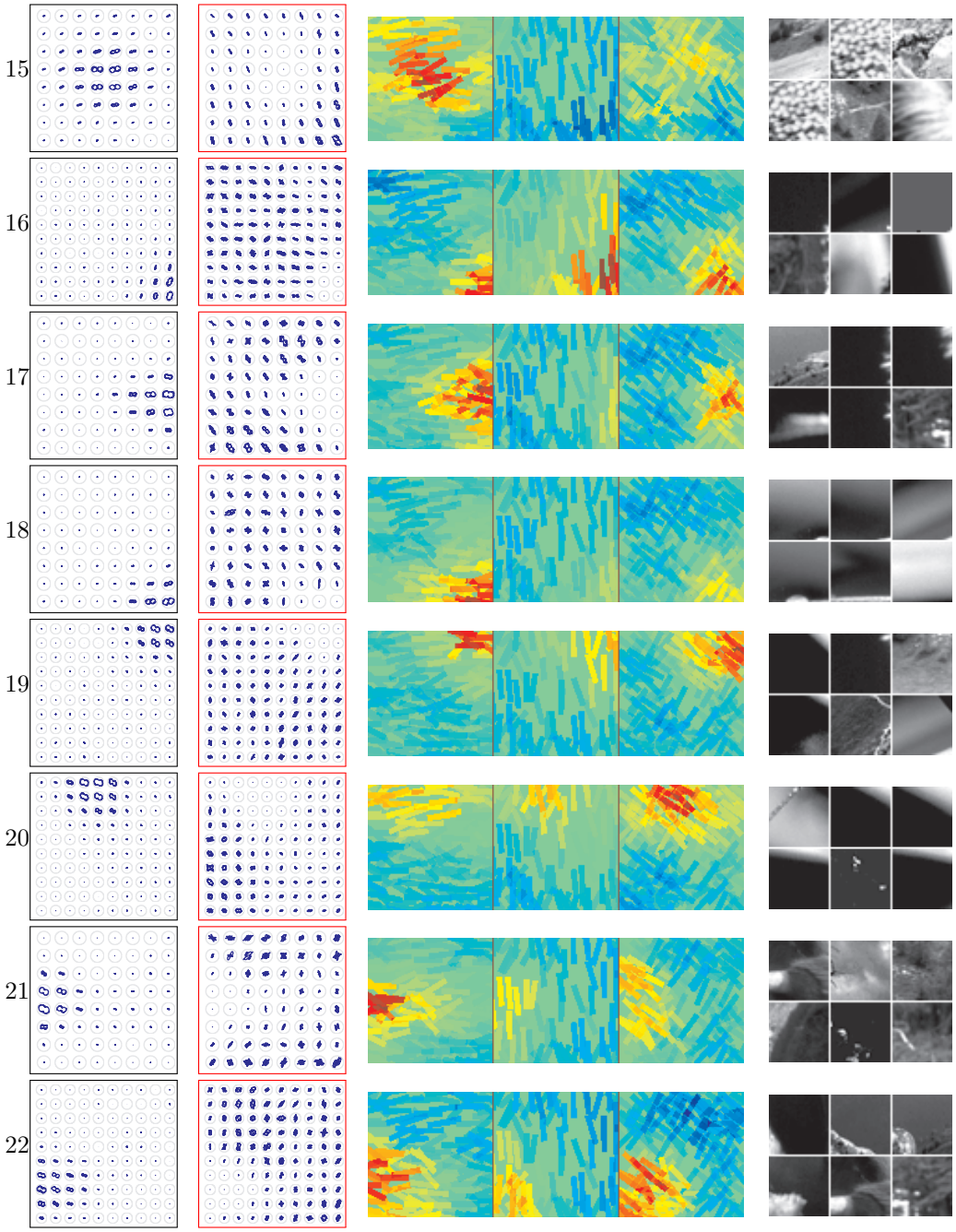
Appendix B. Complete set of third-layer features

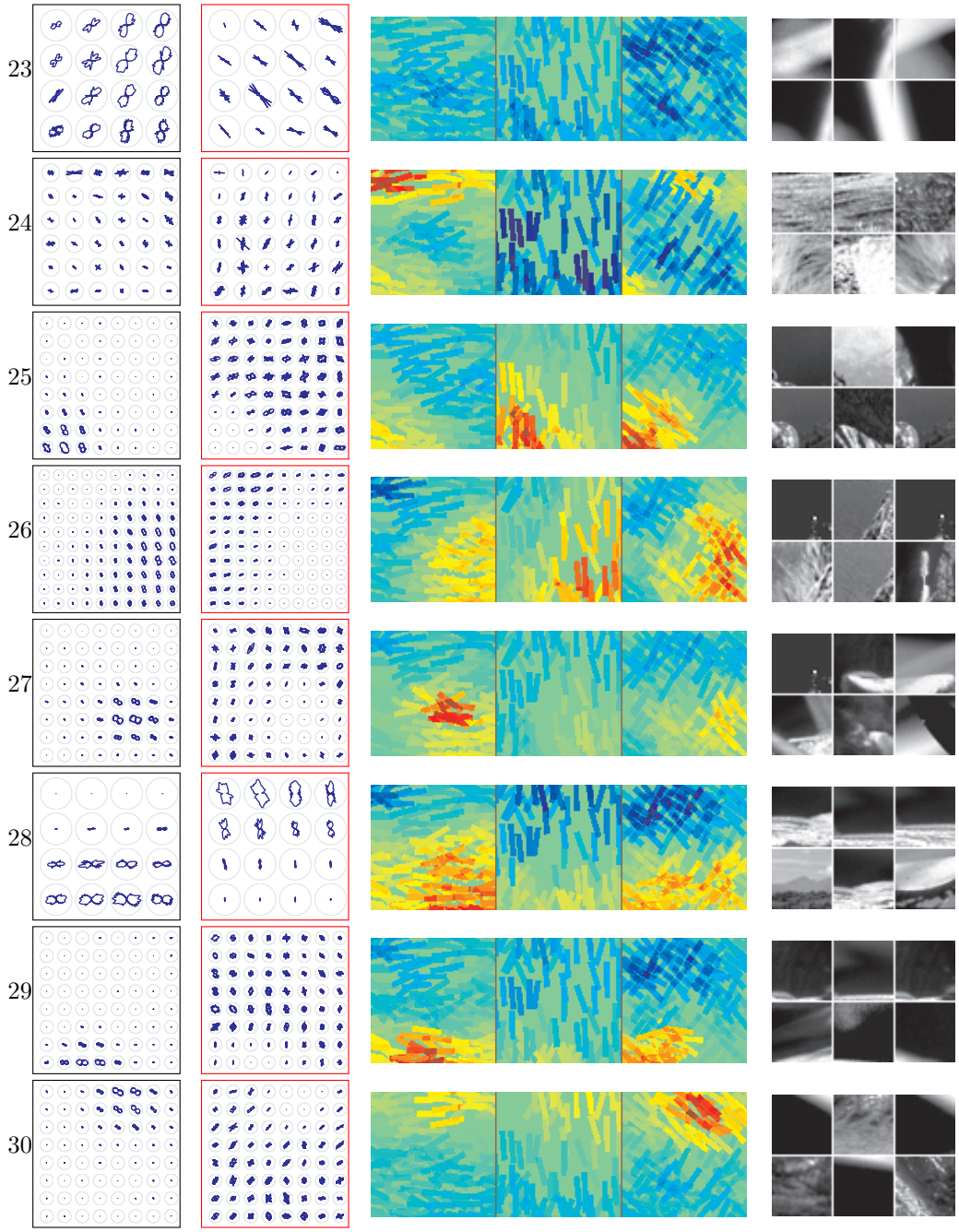
We visualize here the complete set of the learned third-layer features. For patch data, the features are enumerated using upright numbers. For tiny images, we use italic numbers. The features are visualized as in Figure 5 and 6. Note that icons for low-frequency second-layer units were not used in the visualization of the optimal second-layer activity, as pointed out in the main text.

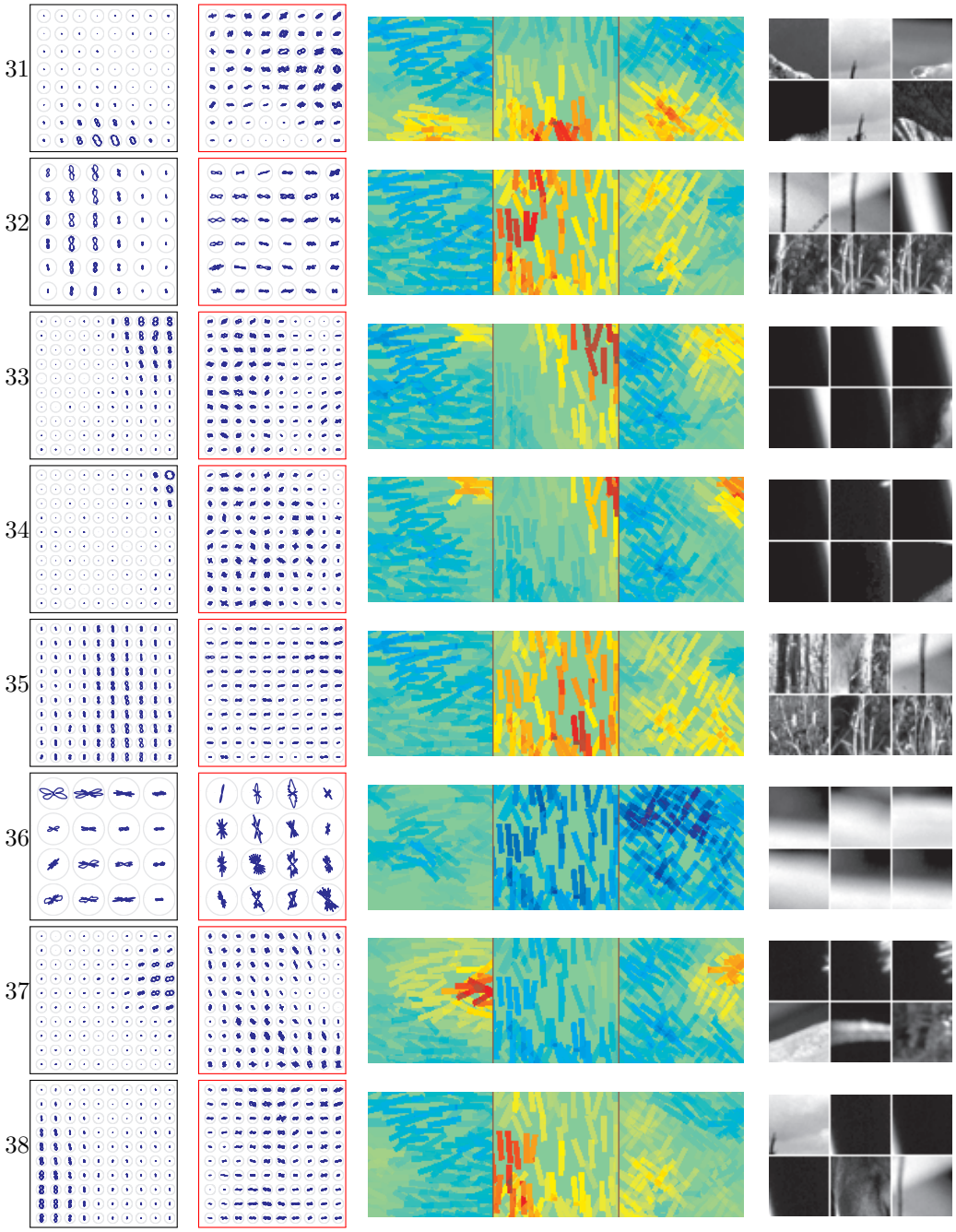
Third-layer features obtained for patch data

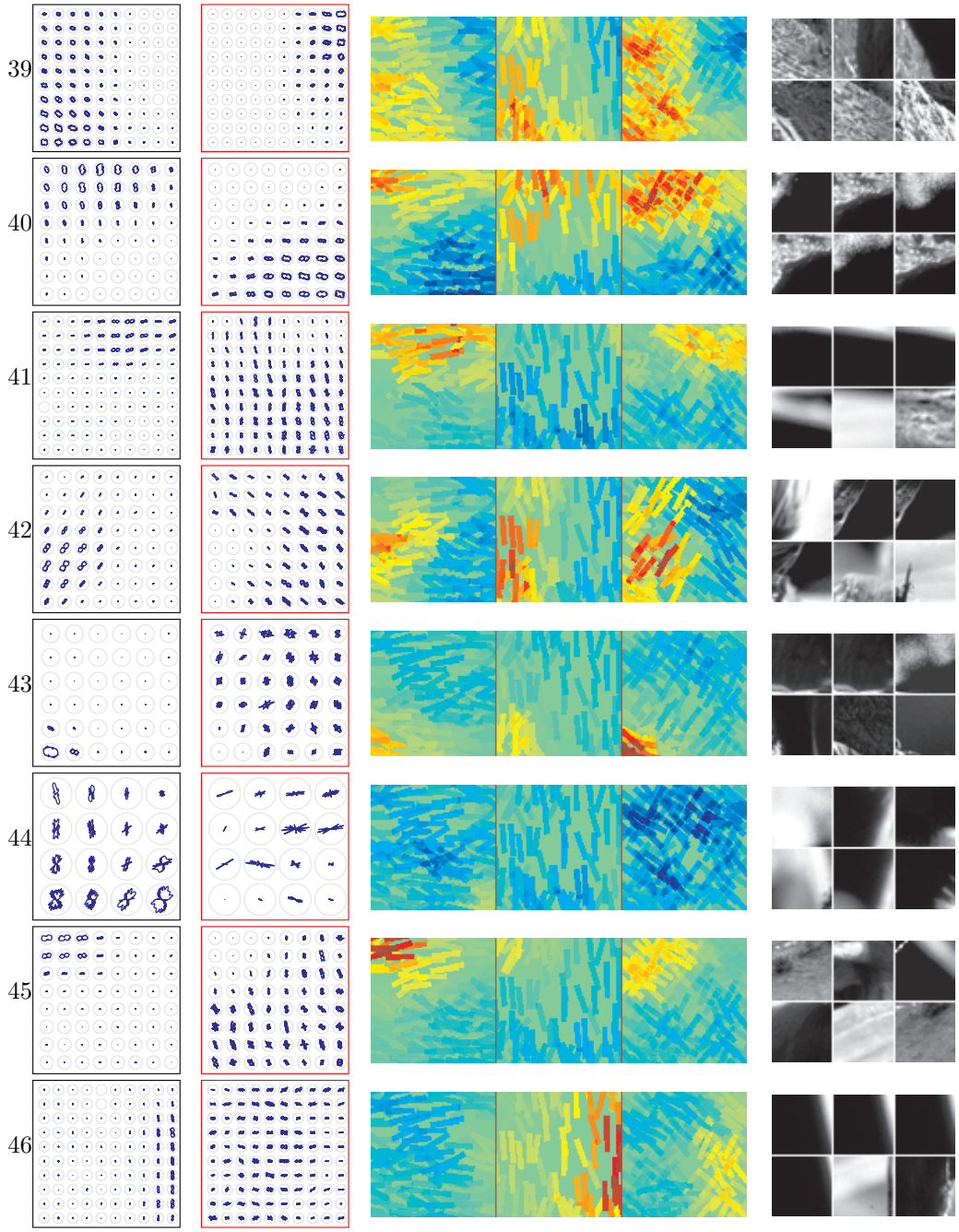


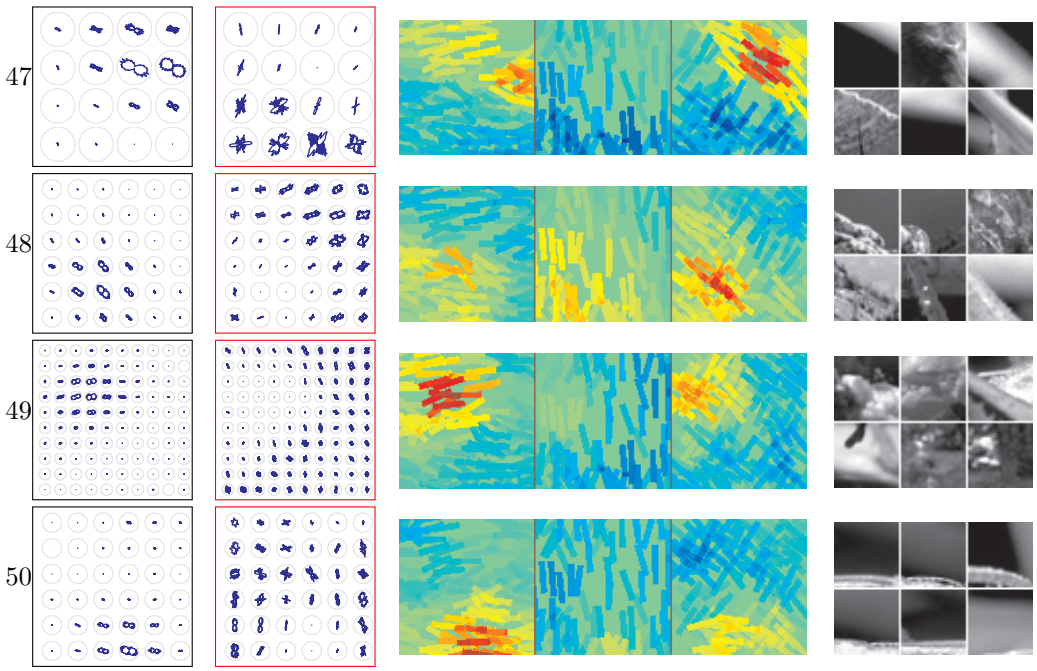

07
08
09
10
11
12
13
14


15
16
17
18
19
20
21
22

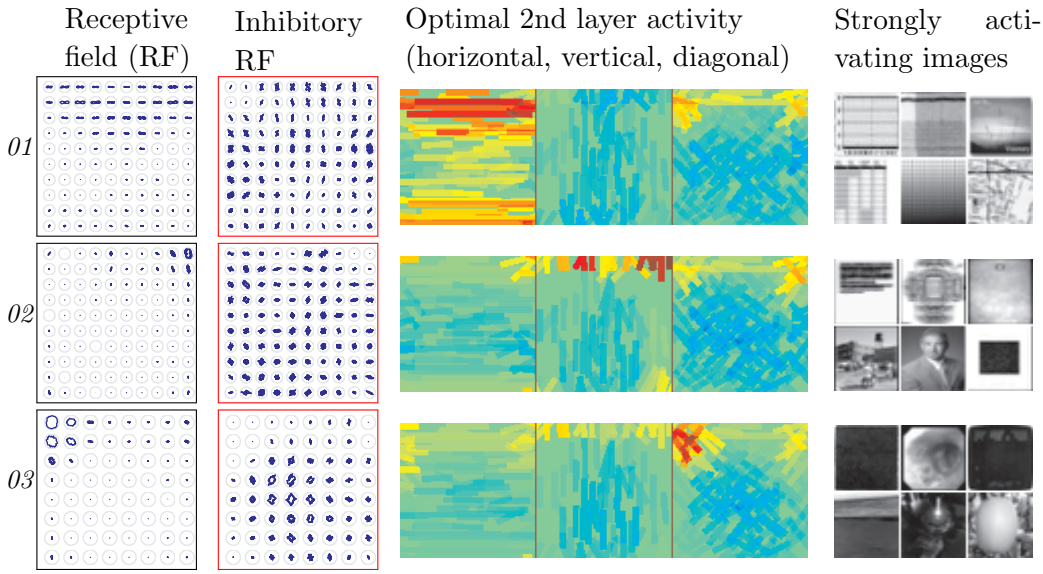

23
24
25
26
27
28
29
30

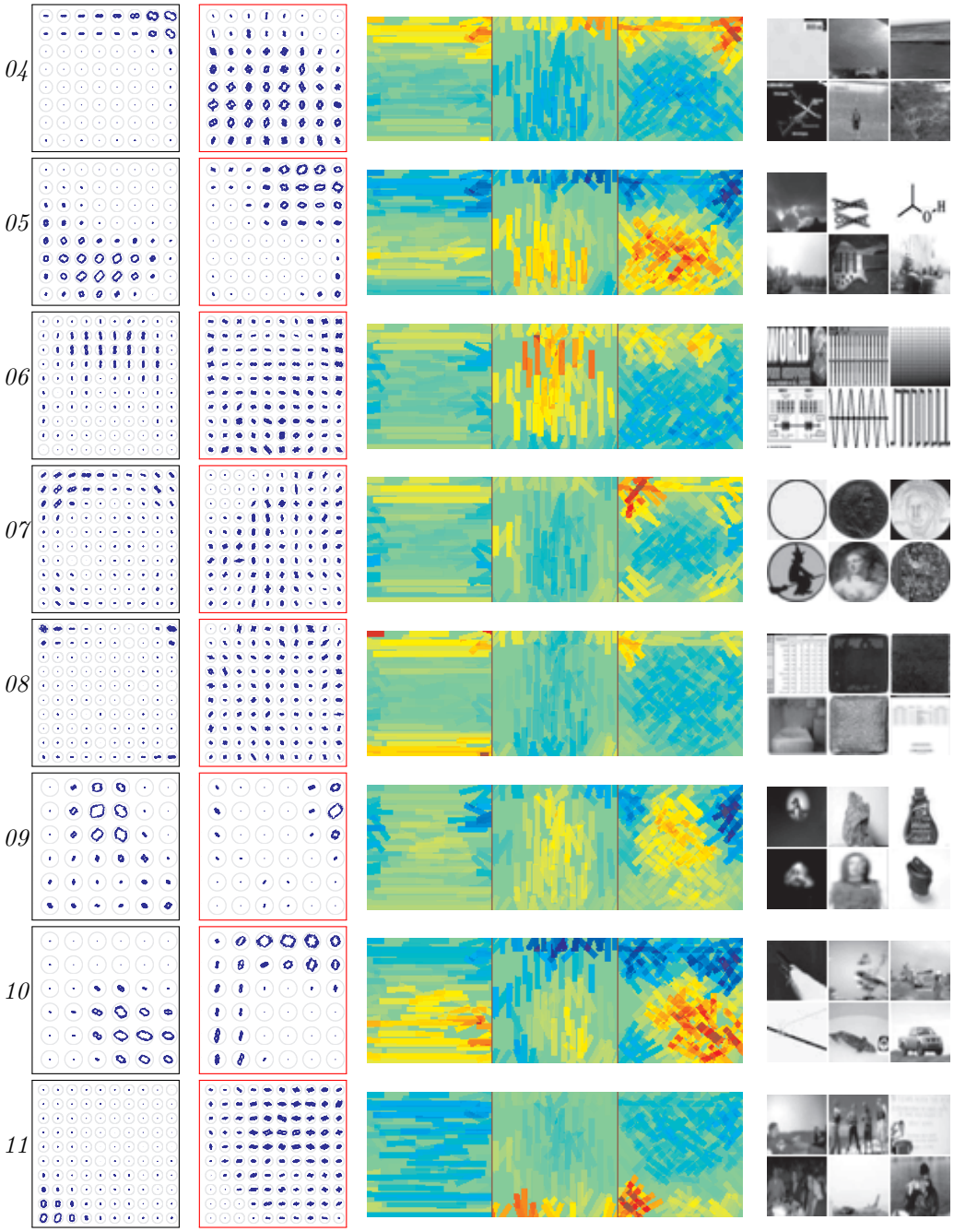

31
32
33
34
35
36
37
38

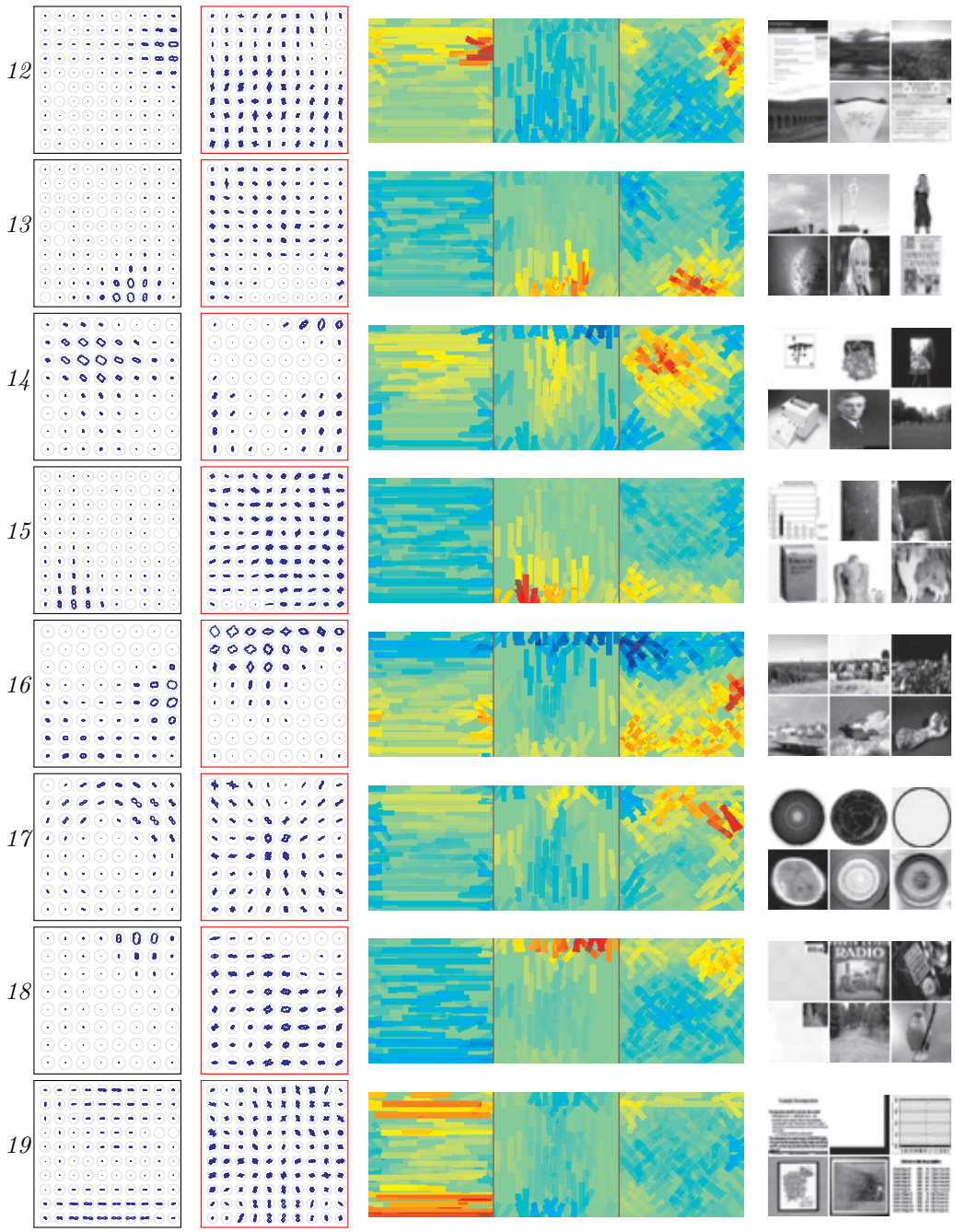

39
40
41
42
43
44
45
46

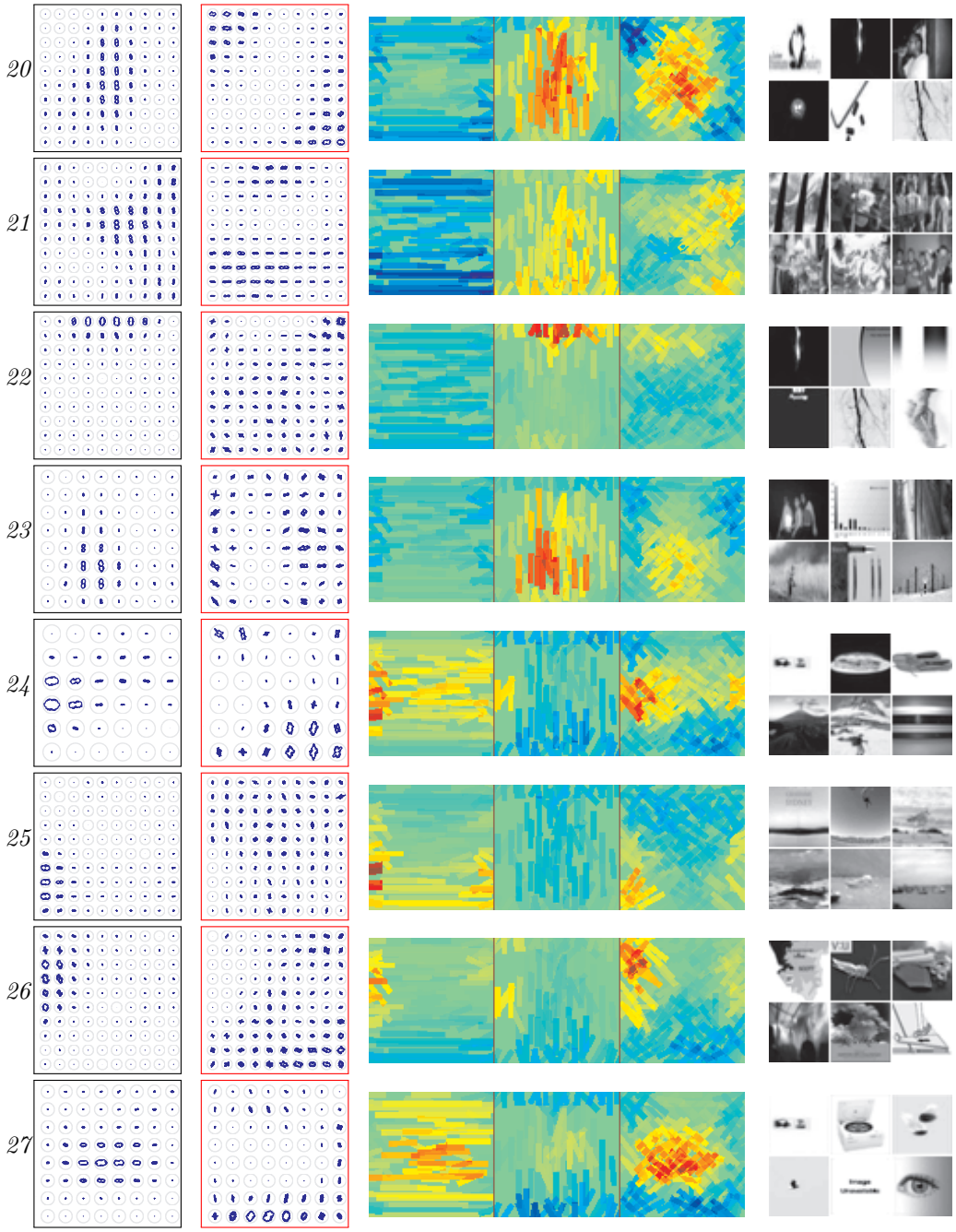


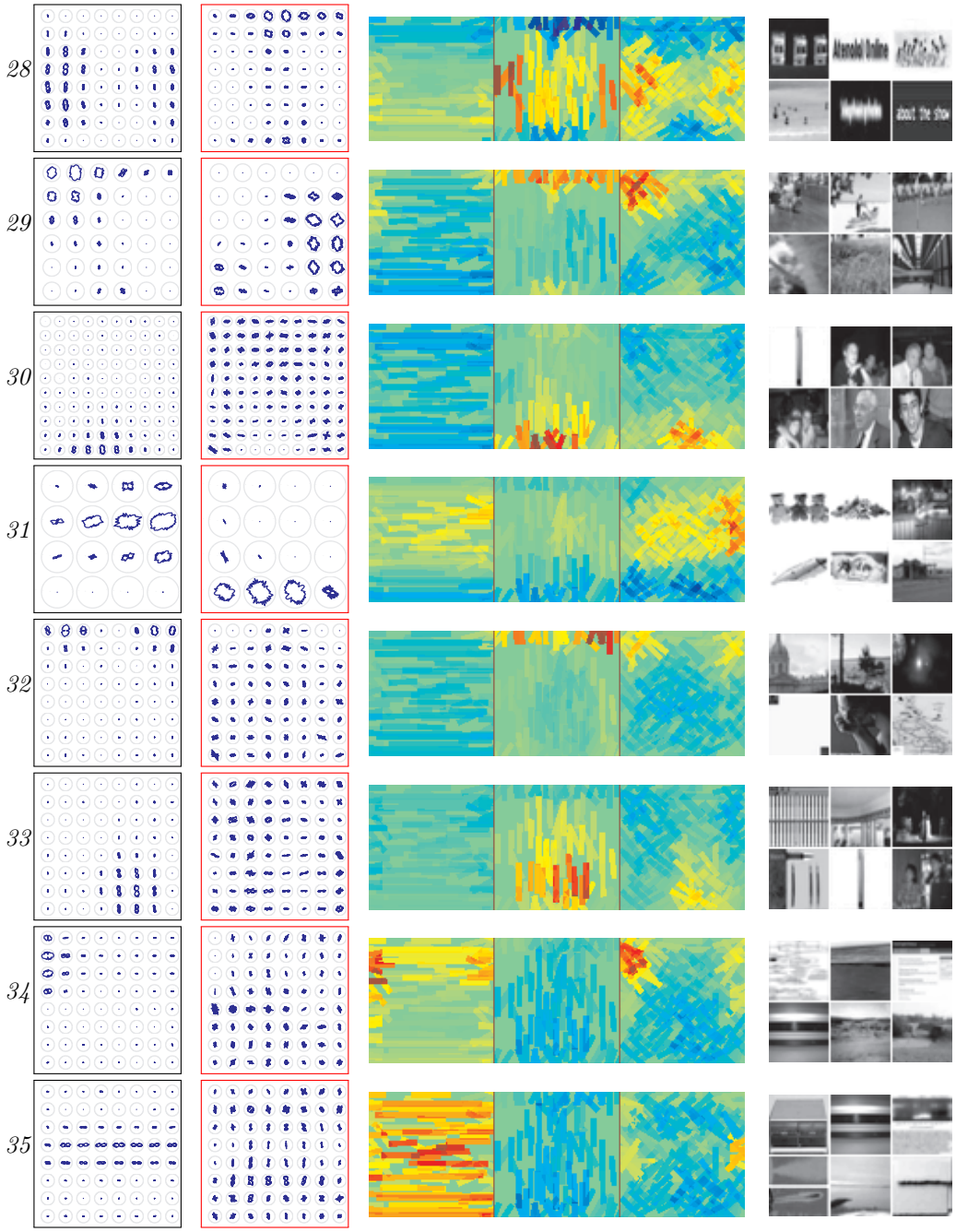
Third-layer features obtained for tiny images

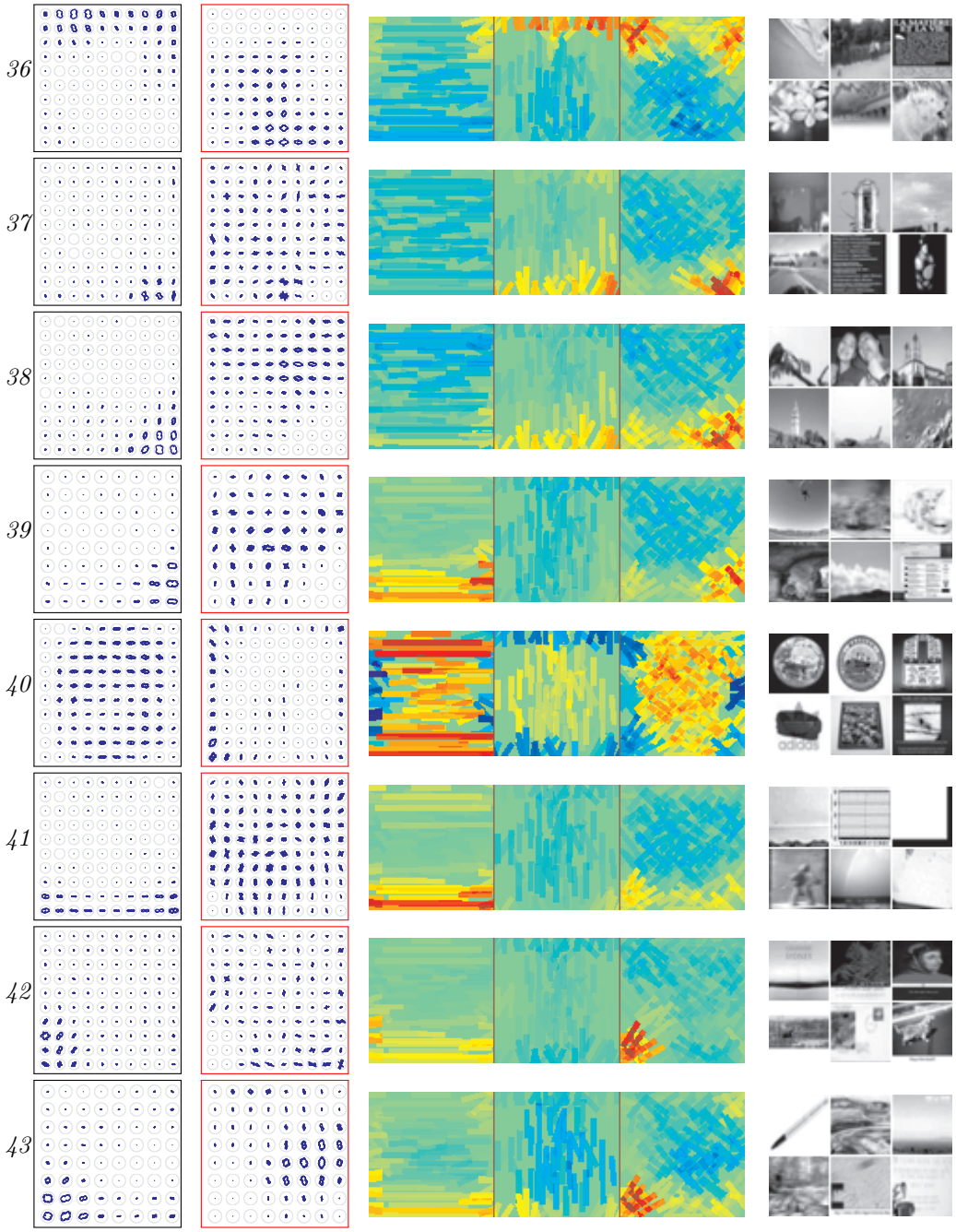


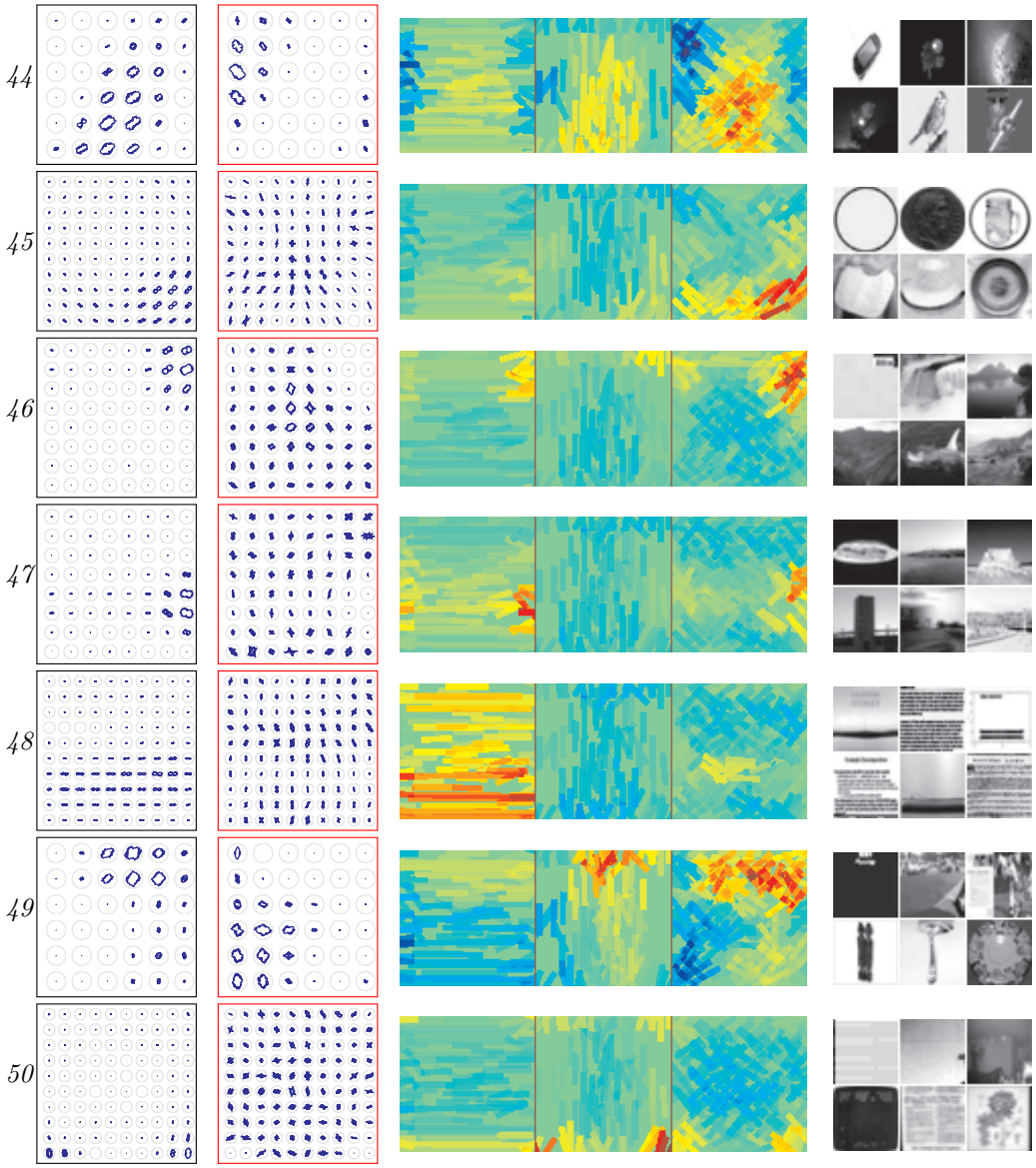

04
05
06
07
08
09
10
11


12
13
14
15
16
17
18
19
RADIO


20
21
22
23
24
25
26
27

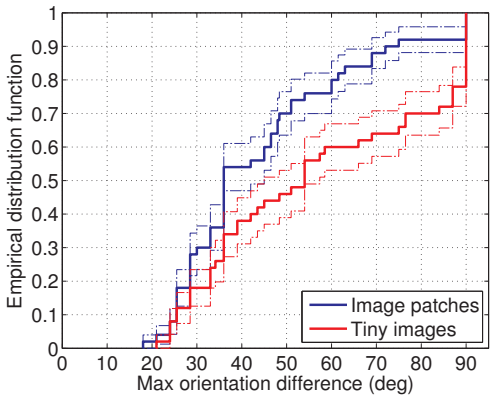

28
29
30
31
32
33
34
35


36
37
38
39
40
41
42
43

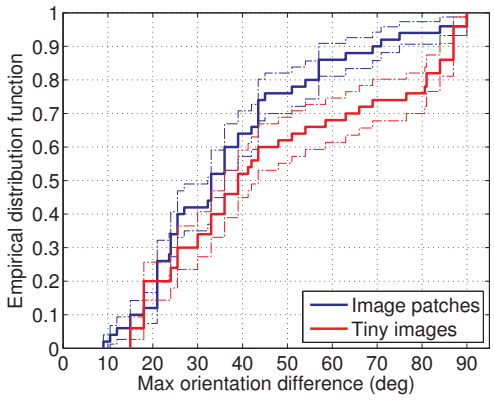

44
45
46
47
48
49
50

Appendix C. Homogeneity of the receptive fields

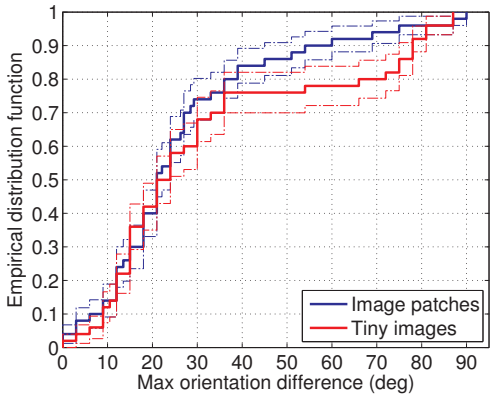
In Figure C.14, we show the distribution of the maximal difference in orientation tuning within a receptive field for the population of learned third-layer units.



(a) $r = 0.25$



(b) $r = 0.5$



(c) $r = 0.75$

Figure C.14: Cumulative distribution functions (cdfs) for the maximal difference in orientation tuning within a third-layer receptive field. The distribution functions were computed for 50 units. The dash-dotted curves show the cdfs $\pm$ one standard error. Locations within a receptive field which yielded a response less than r times the maximal response were excluded from the analysis, as in Figure 7b in the main text. The reason is that the preferred orientation cannot be computed reliably if the response is small.

Appendix D. Sparsity of the feature outputs

We analyze here the sparsity of the feature outputs across the hierarchy. We use three different indices to measure sparsity. We first define the indices in a more general way and explain then how to apply them to measure lifetime or population sparsity.

Assume we would like to measure the sparsity of a vector $\mathbf{r} = (r_1 \dots r_m)$ which consists of m non-negative entries r_k. The first index that we use, S_1, is the Gini index which can be computed as

$$S_1(\mathbf{r}) = 1 - \frac{2}{\sum_k r_k} \sum_{k=1}^{m} r_{(k)} \left(1 - \frac{k - \frac{1}{2}}{m} \right), \tag{D.1}$$

where $r_{(k)}$ denotes the k-th smallest entry in $\mathbf{r}$, that is, $r_{(1)} \leq r_{(2)} \leq \ldots \leq r_{(m)}$ (see, for example, Hurley and Rickard, 2009). A value of zero indicates minimal sparsity, it is obtained if r_k is the same for all k. Maximal sparsity is obtained if $\mathbf{r}$ contains only one non-zero element, where S_1 equals $1 - 1/m$, which tends to one with increasing m. The Gini index was shown to have a number of desirable properties to measure sparsity (Hurley and Rickard, 2009). The following two indices, S_2 and S_3, are based on the "1-2 mean" $a(\mathbf{r})$ which shares the same desirable properties (Hurley and Rickard, 2009),[6]

$$a(\mathbf{r}) = \frac{\frac{1}{m} \sum_k r_k}{\sqrt{\frac{1}{m} \sum_k r_k^2}}. \tag{D.2}$$

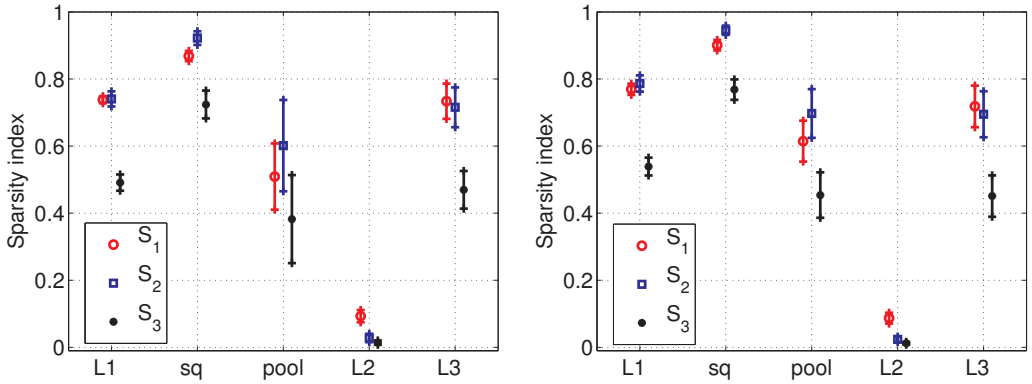
Note that $a(\mathbf{r})$ is one if r_k is the same for all k and $1/\sqrt{m}$ if $\mathbf{r}$ contains only one non-zero element. The indices S_2 and S_3 are two different transformations of $a(\mathbf{r})$ so that zero indicates minimal and one maximal sparsity,

$$S_2(\mathbf{r}) = 1 - (a(\mathbf{r}))^2 \qquad\qquad S_3(\mathbf{r}) = 1 - a(\mathbf{r}). \tag{D.3}$$

Strictly speaking, S_2 and S_3 attain one only in the limit of large m. Index S_2 was, for example, used by Willmore et al. (2011) to measure the lifetime sparsity of cortical cells. With a different normalizing factor,[7] index S_3 was used by Hoyer (2004) to measure sparsity. For binary vectors $\mathbf{r}$, $a = \sqrt{p}$ where p is the fraction of ones in $\mathbf{r}$. Hence, for binary vectors, $S_2 = 1 - p$ measures the fraction of zeros in $\mathbf{r}$, and $S_3 = 1 - \sqrt{p}$.

[6] In the definition by Hurley and Rickard (2009), the sign is reversed.

[7] Hoyer (2004) proposed an index equal to $1/(1 - 1/\sqrt{m})S_3$ whose max is one for all m.



(a) Results for patch data

(b) Results for the tiny images

Figure D.15: Lifetime sparsity of the feature outputs across the hierarchy. L1 refers to the first, L2 to the second and L3 to the third-layer feature outputs, as defined in Section 2.2. The labels "sq" and "pool" denote the intermediate quantities $(y_i^{(1)})^2$ and $\mathbf{w}_i^{(2)} \cdot (\mathbf{y}^{(1)})^2$ that occur in the computation of the second-layer outputs $y_i^{(2)}$. We use three indices, S_1 to S_3, to measure sparsity, see $(D.1)$ to $(D.3)$ for their definitions. Zero indicates minimal, one maximal sparsity. The markers denote averages, the vertical lines are two standard deviations long. We find, first, that sparsity in layer one and three is about the same, second, that pooling reduces sparsity, and third, that squaring increases sparsity while taking the logarithm reduces it.

For the measurement of lifetime sparsity, the vector $\mathbf{r}$ contains the outputs of a single feature for several different input images. We use a test set of $m = 10,000$ natural images. For each feature output, we can then compute the three indices S_1 to S_3. Averaging over the different features in the same layer gives aggregate sparsity indices for each layer. These aggregate indices are shown in Figure D.15, together with their standard deviations: D.15a shows the results for patch data, D.15b the results for the tiny images. The figures suggest three points: First, sparsity in layer one and three is about the same. There is not evidence that, on average, features in layer three are more sparse than in layer one. Second, pooling in the second layer reduces sparsity. Third, the point-wise nonlinearities that occur in the definition of $y_i^{(2)}$ in (2) affect sparsity in opposite ways: squaring increases sparsity while taking the logarithm reduces it.

For the measurement of population sparsity, the vector $\mathbf{r}$ contains for

a single input the outputs of the m features which form the population. Aggregate indices can be obtained by averaging over different inputs. For the measurement of population sparsity, the role of the population and the inputs is thus reversed. The fact that the number of features in each layer is smaller than the number of test images, and that each layer contains a different number of features makes measuring population sparsity, however, more difficult. With these caveats in mind, we report that the indices for average population sparsity, computed directly as outlined above and without any adjustment to the different population sizes, take similar values as for lifetime sparsity.