Effect of normalization on feature extraction and classification in Deep Belief Networks

Conference Paper · January 2012

CITATIONS
0

READS
79

3 authors:

Ahmad Hasasneh
Palestine Ahliya University
13 PUBLICATIONS **54** CITATIONS
SEE PROFILE

Emmanuelle Frenoux
Université Paris-Sud 11
17 PUBLICATIONS **110** CITATIONS
SEE PROFILE

Tarroux Philippe
Université Paris-Sud 11
71 PUBLICATIONS **1,440** CITATIONS
SEE PROFILE

Effect of normalization on feature extraction and classification in Deep Belief Networks

Ahmad Hasasneh [1,2], Emmanuelle Frenoux [2,1] and Philippe Tarroux [2,3]

1- Paris Sud University - Department of Computer Science
Orsay, F-91405 - France

2- LIMSI - CNRS
B.P. 133, F-91403 - France

3- Ecole Normale Supérieure
45 rue d'Ulm, Paris, F-75230 - France

Abstract. Use a good set of features able to code images is of major importance for good classification scores. Most of the methods used the last few years are based on empirically determined features such that GiST, SURF or SIFT detectors. An alternative way is to try to compute an alphabet of features from which the initial set of images is statistically likely to have been generated. This recent approach based on Restricted Boltzmann Machines has been popularized by Hinton [**?**] who proposed an efficient algorithm for computing the underlying generative model. One of the major interest of the approach is that it is grounded on statistical theory of image reconstruction and that the RBM layers can be stacked so that the initial features can be non-linearly combined. These deep belief networks (DBNs) is reminiscent of the way the ventral pathway of primate cortex code images and scenes. The code obtained at the output of the network can be used for classification. In particular it has been used for semantic place recognition (SPR) in robotics [**?**], a problem in which a robot has to find its present location on the basis of the visual aspect of the scene. We have previously shown [**?**] that these DBNs provide state of the art results. For this purpose, the initial data were normalized as usual [] using a whitening procedure. However, when considering brain models, this whitening procedure is non-realistic since it is difficult to account for global computations in the brain. This is one of the reason why, in the present work, we replaced this global normalization procedure by a local one. Unexpectedly, this procedure gave better SPR classification results. In this paper, we explore the reason why local normalization could be a better way than whitening to prepare the data for classification. One of the main reason is that local normalization better preserves the spatial frequency composition of the images and thus retains more information on the organization of the image than whitening. Both empirical and theoretical evidences for that are illustrated in the paper. This work opens the way to the elaboration of a link between two important characteristics of images : their statistical properties and their frequency composition.

1 Introduction

SPR requires the use of an appropriate feature space that allows an accurate and rapid classification. Recent works have been developed for this problem

based on visual descriptors. In particular, these descriptors are either based on global images features (GiST and CENTRIST) (see, for instance, [10, 33, 34]), or on local signatures computed around interest points (SIFT and SURF) (see, for instance, [35, 36]). However, these representations first need to use Bag-of-Words (BoWs) methods, which consider only a set of interest in the image, to reduce their size and then followed by the use of vector quantization such that the image is eventually represented as a histogram. Discriminative approaches can be used to compute the probability to be in a given place according to the current observation. Generative approaches can also be used to compute the likelihood of an observation given a certain place within the framework of Bayesian filtering. Among of these approaches, some works [32] omit the quantization step and model the likelihood as a Gaussian Mixture Model (GMM). Recent approaches also propose to use naive Bayes classifiers and temporal integration that combine successive observations [37].

Contrarily to these empirical methods, new machine learning approaches have recently emerged strongly related to the way natural systems code images [14].

These methods are based on the consideration that natural image statistics are not Gaussian as it would be if they have had a completely random structure [25]. The auto-similar structure of natural images allowed the evolution to build optimal codes. These codes are made of statistically independent features and many different methods have been proposed to construct them from image datasets. Imposing locality and sparsity constraints to these features is very important. This is probably due to the fact that any simple algorithms based on such constraints can achieve linear signatures similar to the notion of receptive field in natural systems. Recent years have seen an interesting interest in computer vision algorithms, that rely on local sparse image representations, especially for the problems of image classification and object recognition

[9, 26, 27, 28, 30]. Moreover, from a generative point of view, the effectiveness of local sparse coding, for instance for image reconstruction [29], is justified by the fact that an natural image can be reconstructed by a smallest possible number of features. It has been shown that Independent Component Analysis (ICA) produces localized features. Besides it is efficient for distributions with high kurtosis well representative of natural image statistics dominated by rare events like contours; however the method is linear and not recursive. These two limitations are released by DBNs [12] that introduce non-linearities in the coding scheme and exhibit multiple layers. Each layer is made of a RBM, a simplified version of a Boltzmann machine proposed by Smolensky [13] and Hinton [11]. Each RBM is able to build a generative statistical model of its inputs using a relatively fast learning algorithm, Contrastive Divergence (CD), first introduced by Hinton [11]. Another important characteristic of the codes used in natural systems, the sparsity of the representation [14], is also achieved in DBNs. Moreover, it has been shown that these approaches remain robustness to extract local sparse efficient features from tiny images [32].

However, while a sparse representation has been assumed to be a linearly

separable in several works, for example [30, 31], and thus simplifies the overall classification problem, the question of whether smaller or larger sparse features are more beneficial to improve the recognition rates remains an open question. Therefore, the fundamental contributions of this paper are three-fold. First, it demonstrates that DBNs coupled with tiny images can be successfully used in the context of SPR. Second, it provides a simpler alternative way to the BoW methods. Third, it evaluates the influence of data normalization on the detection of features and thus on SPR performances. To our knowledge, there is no empirical study yet showing that larger sparse features based on DBNs improve recognition performance compared with smaller ones.

2 Model description

2.1 Image preprocessing

Usually, natural images are highly structured and contain significant statistical redundancies, i.e. their pixels have strong correlations [15, 16]. For example, it is well known that natural images bear considerable regularities in their first and second order statistics (spatial correlations), which can be measured using the autocorrelation function or the Fourier power spectral density [17]. These correlations are due to the redundant nature of natural images (adjacent pixels usually have strong correlations dur to low frequency background except around edges). The presence of these correlations allows image reconstruction using Markov Random Fields. It has thus been shown [18, 17, 20] that the edges are the main characteristics of the natural images and that they are rather coded by higher order statistical dependencies. Thus the statistics of natural images is not Gaussian since the moments greater than order-two are zero for Gaussian distributions. This statistics of natural images is then dominated by rare events like contours, leading to high-kurtosis long-tailed distributions.

2.1.1 Data whitening and local normalization

Statistical whitening The most popular method to remove these expected order-two correlations is known as whitening. This conventional whitening is a way to center and reduce a data cloud according to its principal directions. Centering the data means that the mean along to the different directions is subtracted to the initial data. The directions are indifferent provided they form a orthogonal set of axes of the same dimension as the initial data. reduction is operated along the principal directions of the data cloud. It means that the initial data is first projected onto the eigen vectors of the variance-covariance matrix and then devided by values proportional to the corresponding eigenvalues. The resulting equation is thus :

$$\tilde{x} = \Lambda^{-\frac{1}{2}} W x \tag{1}$$

The goal of whitening is to make the variance-covariance matrix of the tranformed data equals to unity. To do that consider a set of data already centered:

$$X = \begin{bmatrix} x_1^{(1)} & \dots & x_1^m \\ & \dots & \\ x_n^{(1)} & \dots & x_n^m \end{bmatrix} \tag{2}$$

Its variance covariance matrix is xx^τ (note that we are interested in the variance of the components accross the data).

We are searching for a transform $\tilde{x} = Dx$ such that $xx^\tau = I$.

$$\tilde{x}\tilde{x}^\tau = I \implies Dxx^\tau D^\tau = I \tag{3}$$

$Dxx^\tau D^\tau = DRD^\tau$ where R is the variance covariance matrix of x. This matrix results from the equation $R = W^\tau \Lambda W$ where W and Λ are respectively the eigenvector and eignevalue matrices of R. Imposing $DW^\tau \Lambda W D^\tau = I$ implies $D^{-1} = W^\tau \Lambda^{\frac{1}{2}}$ and thus:

$$D = \Lambda^{-\frac{1}{2}} W \tag{4}$$

This transform has the effect of rouding or sphericizing the data cloud. If the initial data cloud is distributed as a gaussian distribution, the shape of the final data is a fuzzy sphere since there is no more prefered direction in the data (the elongation along the main axes becomes identicalin all directions).

It has been shown that whitening is a useful pre-processing strategy in ICA [19, 2]. It seems also a mandatory step for the use of clustering methods in object recognition [21]. Whitening being a linear process, it does not remove the higher order statistics or regularities present in the data. The theoretical grouding of whitening is simple: after centering, the data vectors are projected onto their principal axes (computed as the eigen-vectors of the variance-covariance matrix) and then divided by the variance along these axes. In this way, the data cloud is sphericized, letting appear only the usually non orthogonal axes corresponding to its higher-order statistical dependencies.

Another way to pre-process data is to perform local normalization. In this case, each patch $x^{(i)}$ is normalized by subtracting the mean and dividing by the standard deviation of its elements. For visual data, this corresponds to local brightness and contrast normalization. One can find in [21] a study of whitening and local normalization and their effect on a further classification task. However we can note that this study has been performed using two databases, NORB and CIFAR, that have been especially designed for object recognition.

We can also note that in [22], the authors argue that whitening speeds-up the convergence of the algorithm without any justification. It could probably be related to [3] who showed that the Hessian matrix of the objective function is a function of the correlation matrix. In this article, we investigate the effect of whitening and local normalization on the detection of features using a RBM learning algorithm. These factors, orthogonal to the learning algorithm itself, can have a large impact on SPR performance.

1/f whitening Another approach to whitening has been introduced by Olshausen [**?**] firstly to remove the low frequency bias due to the sampling of square patches in the construction of features detectors in ICA. It is indeed true that in a suqare patch, the low frequencies are over-represented in the corners. To avoid this phenomenon, the Fourier transform of each image is filtered using an isotropic filter the Fourier transform of which is:

$$F_h(f) = f * e^{-\frac{1}{2}f^2/\sigma^2} \tag{5}$$

2.2 Unsupervised feature space construction

2.2.1 Gaussian-Bernoulli restricted Boltzmann machines

Unlike a classical Boltzmann machine, a RBM is a bipartite undirected graphical model $\theta = \{w_{ij}, b_i, c_j\}$, that learns a generative model of the observed data. It consists in two layers. The hidden layer, containing latent variables $\mathbf{h}$, is used to generate the visual layer, containing observed variables $\mathbf{v}$. While generation $P(\mathbf{v}|\mathbf{h})$ is learned, the undirected connections also allow recognition $P(\mathbf{h}|\mathbf{v})$. The two layers are fully connected through a set of weights w_{ij} and biases $\{b_i, c_j\}$, and there are no connections between units of the same layer. For a conventional RBM, a joint configuration of the binary visible units and the binary hidden units has an energy function, $E(\mathbf{v}, \mathbf{h}; \theta)$ given by:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_i \sum_j v_i h_j w_{ij} - \sum_{i \in \mathbf{v}} b_i v_i - \sum_{j \in \mathbf{h}} c_j h_j. \tag{6}$$

The probabilities of the state for a unit in one layer conditional to the state of the other layer can therefore be easily computed. According to Gibbs distribution:

$$P(\mathbf{v}, \mathbf{h}; \theta) = -\frac{1}{Z(\theta)} \exp^{-E(\mathbf{v}, \mathbf{h}; \theta)}, \tag{7}$$

where $Z(\theta)$ is a normalizing constant. Thus after marginalization, the probability of a particular hidden state configuration $\mathbf{h}$ can be derived as follows:

$$P(\mathbf{h}; \theta) = \sum_{\mathbf{v}} P(\mathbf{v}, \mathbf{h}; \theta) = \frac{\sum_{\mathbf{v}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}}. \tag{8}$$

However, according to [4], the above conditional probability can be derived using the logistic sigmoid function as follows:

$$P(h_j = 1 \mid \mathbf{v}; \theta) = \sigma(c_j + \sum_i w_{ij} v_i), \tag{9}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function. Once the hidden binary states are computed, we produce a "reconstruction" of the original patch by setting the state of each visible unit to be 1 with a probability:

$$P(v_i = 1 \mid \mathbf{h}; \theta) = \sigma(b_i + \sum_j w_{ij} h_j). \tag{10}$$

However, logistic or binary visible units are not appropriate for multi-valued inputs like pixel levels, because logistic units are a very poor representation for data such as patches of natural images. To overcome this problem, as suggested by [5], in the present work we replace the binary visible units by a zero-means Gaussian activation scheme as follows:

$$P(v_i = 1 \mid \mathbf{h}; \theta) \leftarrow \mathcal{N}(b_i + \sum_j w_{ij} h_j, \sigma^2), \tag{11}$$

where σ^2 denotes the variance of the noise. In this case the energy function of Gaussian-Bernoulli RBM is given by:

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_{i \in \mathbf{v}} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j \in \mathbf{h}} c_j h_j - \sum_i \sum_j \frac{v_i}{\sigma_i} h_j w_{ij}. \tag{12}$$

2.2.2 Training RBMs with a sparsity constraint

To learn RBM parameters, it is possible to maximize the log-likelihood in a gradient ascent procedure. Thus, the derivative of the log-likelihood of the model over a training set D is given by:

$$\frac{\partial}{\partial \theta} L(\theta) = \left\langle \frac{\partial E(\mathbf{v}, \theta)}{\partial \theta} \right\rangle_M - \left\langle \frac{\partial E(\mathbf{v}, \theta)}{\partial \theta} \right\rangle_D, \tag{13}$$

where the first term represents an average with respect to the model distribution and the second an expectation over the data. Although the second term is straightforward to compute, the first one is often intractable. This is due to the fact that computing the likelihood needs to compute the partition function, $Z(\theta)$, that is usually intractable. Markov-Chain Monte Carlo methods, like Gibbs sampling, can be used to approximate this expectation term. These methods, however, are very slow and suffer from high variance in their estimates.

In 2002, Hinton proposed a quick learning procedure called CD. This learning algorithm is based on the consideration that minimizing the energy of the network is equivalent to minimize the distance between the data and a statistical generative model of it. A comparison is made between the statistics of the data and the statistics of its representation generated by Gibbs sampling. Therefore, in CD learning, we try to minimize the Kullback-Leibler Divergence between the data distribution, Q^0, and the model distribution, Q^∞ as follows:

$$CD_n = KL(Q^0 || Q^\infty) - KL(Q^1 || Q^\infty). \tag{14}$$

The key benefit for the CD is that the intractable term, Q^∞, in the above equation cancels each other out, as explained in [11, 6]. It means that, in practice, we use usually only few steps of Gibbs sampling (most of the time reduced to one) to ensure convergence. For a RBM, the weights of the network can therefore be updated using the following equation:

$$-\frac{\partial}{\partial w_{ij}} \left(Q^0 \| Q^\infty - Q^1 \| Q^\infty \right) \approx \langle v_i^0 h_j^0 \rangle_{Q^0} - \langle v_i^n h_j^n \rangle_{Q^1}. \tag{15}$$

This equation can be rewritten as follows:

$$w_{ij} \leftarrow w_{ij} + \eta(\langle v_i^0 h_j^0 \rangle_{data} - \langle v_i^n h_j^n \rangle_{recon.}), \tag{16}$$

where η is the learning rate, v^0 corresponds to the initial data distribution, h^0 is computed using equation 4, v^n is sampled using the Gaussian distribution in equation 6 and with n full steps of Gibbs sampling, and h^n is again computed from equation 4. Also, for separate biases of visible and hidden neurons, the update rules are, in analogy to the update rule for the weights:

$$b_i \leftarrow b_i + \eta[\langle v_i^0 \rangle_{data} - \langle v_i^n \rangle_{recon.}], \tag{17}$$

and

$$c_j \leftarrow c_j + \eta[\langle h_j^0 \rangle_{data} - \langle h_j^n \rangle_{recon.}], \tag{18}$$

where v_i, h_j, b_i, and c_j denote the i^{th} visible neuron, the j^{th} hidden neuron, the i^{th} visible bias, and the j^{th} hidden bias respectively.

Concerning the sparsity constraint in RBMs, we follow the same approach developed in [38]. This method introduces a regularizer term that makes the average hidden variable activation low over the entire training examples. Thus, the activation of the model neurons become also sparse. In fact, this method is similar to the one used in other models [20]. Thus, as illustrated in [38], given a training set $\{v^{(1)}, \ldots, v^{(m)}\}$ including m examples, we pose the following optimization problem:

$$minimize_{\{w_{ij}, b_i, c_j\}} - \sum_{l=1}^{m} log\Big(\sum_{h} P(\mathbf{v}^{(l)}, \mathbf{h}^{(l)}) \Big) + \lambda \sum_{j=1}^{n} \Big| p - \frac{1}{m} \sum_{l=1}^{m} \mathbb{E}[h_j^{(l)} | \mathbf{v}^{(l)}] \Big|^2, \tag{19}$$

where $\mathbb{E}[.]$ is the conditional expectation given the data, p is the sparsity target controlling the sparseness of the hidden units h_j, and λ is the sparsity cost. Thus, after involving this regularization in the CD learning algorithm, the gradient of the sparsity regularization term over the parameters (weights w_{ij} and the hidden biases c_j) can be written as follows:

$$w_{ij} \leftarrow \mu * w_{ij} + \eta * \big[\big((\langle v_i^0 h_j^0 \rangle - \langle v_i^n h_j^n \rangle) \big) - \lambda * (p - \frac{1}{m} \sum_{l=1}^{m} p_j^{(l)}), \tag{20}$$

$$c_j \leftarrow c_j + \eta[\langle h_j^0 \rangle_{data} - \langle h_j^n \rangle_{recon}] - \lambda * (p - \frac{1}{m} \sum_{l=1}^{m} p_j^{(l)}), \tag{21}$$

where m in this case is the size of the mini-batch and $p_j^{(l)} \triangleq \sigma(\sum_i v_i^{(l)} w_{ij} + c_j)$.

It has been shown that an sparse RBM learning algorithm can capture interesting high-order features from natural image statistics [38]. The hope is that such a learning algorithm remains capable to capture higher-order features from various databases, like a database created for the purpose of robot localization.

2.2.3 Layerwise training for DBNs

RBM can be stacked to generate a DBN architecture, where the model parameters θ_i, at layer i are trained by keeping the model parameters in the lower layers constant. In other words, the DBN training algorithm trains the RBM layers in a greedy layerwise fashion. The model parameters at layer $i-1$ are frozen and the conditional probabilities of the hidden unit values are used to generate the data to train the model parameters at layer i. This process can be repeated across the layers to obtain sparse representations of the initial data that will be used as final input vectors to perform the classification process.

3 Results

3.1 Effect of normalization on the feature space

3.1.1 Studies on the van Hateren's natural image database

In order to investigate the impact of the data normalization on the detection of features, we use a popular dataset of natural images, the van Hateren's database [1]. It is a database of high-resolution calibrated monochrome images taken in defined illumination conditions, designed for various image processing tasks. It contains approximately 4000 images of $1536x1024$ pixels.

For this task, we sampled $100,000$ of $16x16$ random patches. These patches are then whitened using a whitening algorithm and normalized using a local normalization in two separate pre-processes as shown in figure 1.

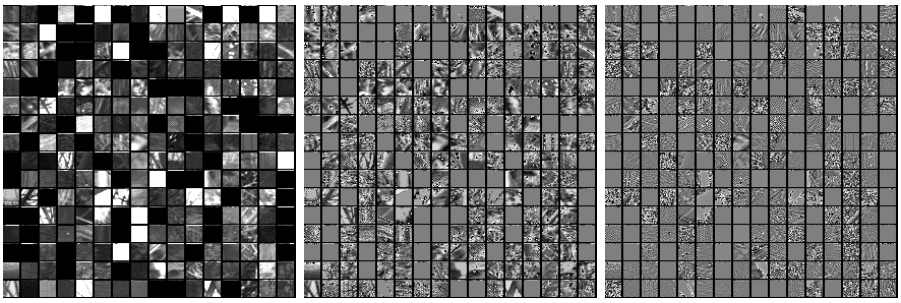

Fig. 1: **First column:** 256 tiny images randomly sampled from the van Hateren database. **Second column:** Normalized version. **Third column:** Whitened version.

For this task, we have conducted two experiments using a dataset of random patches sampled from van Hateren database. After whitening and normalizing these patches in two separate pre-processes as previously said, an over-complete structure $(256-512)$ of the first RBM layer was used.

[1] The van Hateren's Database is available at: http://www.kyb.tuebingen.mpg.de/?id=227

Figure 2 (left) shows features extracted using the locally normalized data, while figure 2 (right) shows features extracted using the whitened one. It is obvious that the features extracted from the whitened data are more localized. Data whitening clearly changes the characteristics of the learned bases. One explanation could be that the second order correlations are linked to the presence of low frequencies in the images. If the whitening algorithm removes these correlations in the original dataset, it leads to whitened data covering only high spatial frequencies. The RBM algorithm in this case finds only high frequency features.

However, the features learned from the normalization data are totally different from the ones learned with whitened data. They remain sparse but cover a broader spectrum of spatial frequencies. An interesting observation is that they look closer to the ones obtained with convolutional networks [39] for which no whitening is applied to the initial dataset. We can mention that these differences between normalized and whitened data have already been observed in [4] and related to better performances for the normalized data on CIFAR-10 in an object recognition task.

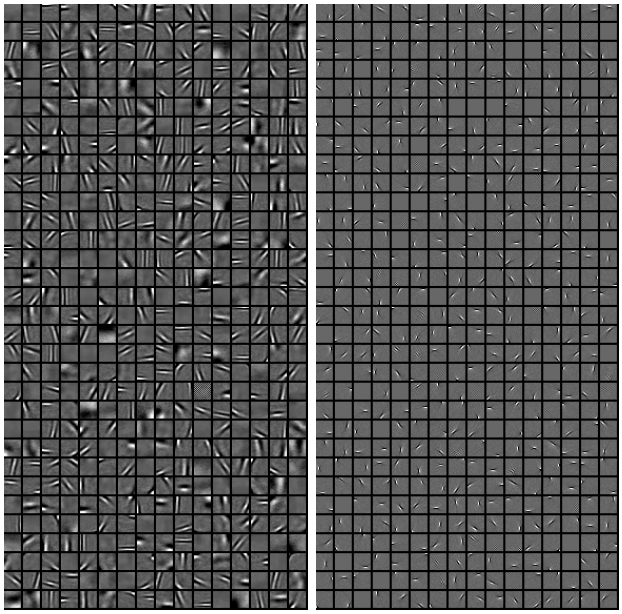

Fig. 2: Learned over-complete natural image bases. **Left:** 512 features learned by training the first RBM layer on normalized image patches (16×16) sampled from Van Hateren dataset. **Right:** The corresponding features learned by training the first RBM layer on whitened image patches (16×16) sampled from the same database. For both experiments, the training protocol is similar to the one proposed in [38] (300 epochs, a mini-batch size of 200, a learning rate of 0.02, an initial momentum of 0.5, a final momentum of 0.9, a weight decay of 0.0002, a sparsity target of 0.02, and a sparsity cost of 0.02).

To try to understand more deeply why features obtained from whitened or normalized patches are different, we computed the mean Fourier spectral density of the patches in the two conditions and we compared them to the same function for the original patches. We plotted the mean of the Log Fourier power spectral density of all the patches according to the Log of the frequencies shown in figure 3. The scale law in $1/f^{\alpha}$ characteristic of natural images is approximatively verified as expected for the initial patches. For the local normalization it is also conserved (the shift between the two curves is only due to a multiplicative difference in the signal amplitude between the original and the locally normalized patches). It means that the frequency composition of the locally normalized images differs from the initial one only by a constant factor. The relative frequency composition is the same as in initial images.

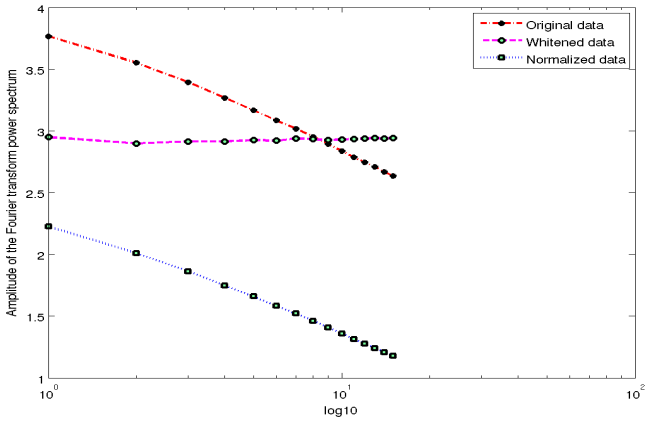

Fig. 3: The Log-Log representation of the mean Fourier power spectrum for image patches with and without normalization. 256 of 16×16 patches have been extracted from the van Hateren database and then normalized. The Log of the Fourier transform of each of these patches has been computed and plotted according to the Log of the spatial frequency.

On the contrary, whitening completely abolishes this dependency of the signal energy with frequency. This means that whitening equalizes the role of each frequency in the composition of the images [2]. This suggests a relationship between the scale law of natural images and the first two moments of the statistics of these images. It is interesting to underline that we have here a manifestation of the link between the statistical properties of an image and its structural properties (in terms of spatial frequencies). This link is well illustrated by the Wiener-Khintchine theorem and the relationship between the autocorrelation function of the image and its power spectral density. Concerning the extracted features, these observations allow to deduce that an equal representation (in

[2]That is an expected effect since whitening can be related to white noise, a noise in which all the frequencies are equally represented.

terms of amplitude) of all the frequencies in the initial signal gives rise to an over-representation of high frequencies in the obtained features. This could be due to the fact that, in the whitened data, the energy contained in each frequency band increases with the frequency while it is constant in initial or normalized images.

However, the result depends on the database used and consequently on the spatial frequencies contained in the initial patches. The fact that local normalization preserves (to a constant value) the same frequency composition as in initial data tends to prove that normalization does not entirely remove second-order correlations. Olshausen [23] showed that, with whitening, ICA mainly retains filters in a narrow range of spatial frequencies. Low spatial frequencies are under-represented in the obtained result. This is clearly what we obtain here with whitening but not with normalization, which tends to save a broader range of spatial frequencies.

We can argue that low frequency dependencies are related to the statistical correlation between neighbor pixels. Thus the suppression of these second order correlations would suppress these low frequencies in the whitened patches. The resulting features set is expected to contain a larger number of low frequency less localized features, what is actually observed.

We are going to see in the next section how the COLD database used to test our SPR model behaves according to these two normalization methods and how these changes in spatial frequency composition affect classification performances.

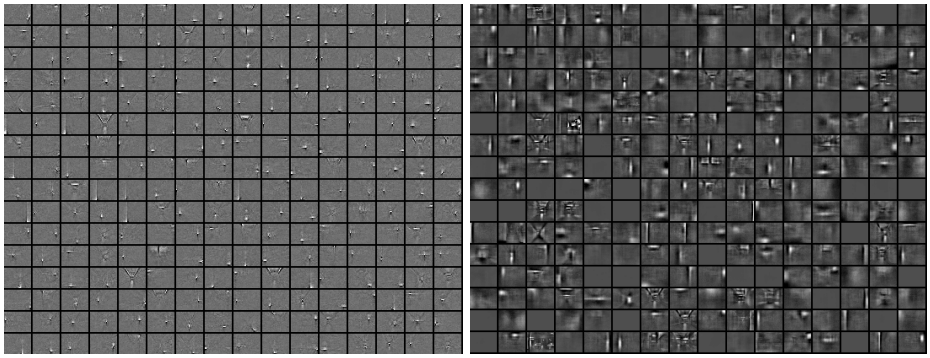

Fig. 4: **Left:** 256 filters obtained by training a first RBM layer on $32x24$ whitened image patches sampled from the COLD database. **Right:** 256 filters obtained by training a first RBM layer on $32x24$ normalized image patches sampled from the COLD database. The training protocol is similar to the one proposed in [8, 38] (300 epochs, a mini-batch size of 100, a learning rate of 0.002, a weight decay of 0.0002, an initial momentum of 0.5, a final momentum of 0.9, a sparsity target of 0.02, and a sparsity cost of 0.02).

3.2 Supervised learning of places

3.2.1 The COLD database

The COLD database (COsy Localization Database) was originally developed by [7] for the purpose of robot localization [3]. This database is a collection of labeled $640x480$ images acquired at five frames/sec during the robot exploration of three different laboratories (Freiburg, Ljubljana, and Saarbruecken). Two sets of paths (standard A and B) have been acquired under different illumination conditions (sunny, cloudy and night), and for each condition, one path consists in visiting the different rooms (corridors, printer areas, *etc.*). These walks across the labs are repeated several times. Although color images have been recorded during the exploration, only gray images are used since previous works have demonstrated that in the case of the COLD database colors are weakly informative and made the system more illumination dependent [7].

3.2.2 Use of tiny images for classification

The typical input dimension for a DBN is approximately 1000 units (*e.g.* $30x30$ pixels). Dealing with smaller patches could make the model unable to extract interesting features. Using larger patches can be extremely time-consuming during feature learning. Additionally the multiplication of the connexion weights acts negatively on the convergence of the CD algorithm. The question is therefore how could we scale the size of realistic images (*e.g.* $300x300$ pixels) to make them appropriate for DBNs?

Tiny images have been successfully used [32] for classifying and retrieving images from the 80-million images database developed at MIT. Torralba showed that the use of tiny images combined with a DBN approach led to code each image by a small binary vector defining the elements of a feature alphabet that can be used to optimally define the considered image. The binary vector acts as a bar-code while the alphabet of features is computed only once from a representative set of images. The power of this approach is well illustrated by the fact that a relatively small binary vector (like the ones we use as the output of our DBN structure) largely exceeds the number of images that have to be coded even in a huge database ($2^{256} \approx 10^{75}$). So, for all these reasons we have chosen image reduction. Thus, as proposed by [32] the image size is reduced to $32x24$ (see for instance figure 5). The final set of tiny images (a new database called tiny-COLD) is centered and whitened/normalized in order to eliminate order 2 statistics. Consequently the variance in equation 6 is set to 1. Contrarily to Torralba, the $32 \times 24 = 768$ pixels of the whitened/normalized images are used directly as the input vector of the network.

As in [**?**] we adopted a $768-256-128$ structure for the network. The features shown in figure 4 (left) have been extracted by training the first RBM layer on $137,069$ whitened image patches (32×24 pixels) sampled from the COLD database. Some of them represent parts of the corridor, which is over-represented

[3] The COLD Database is available at: http://cogvis.nada.kth.se/COLD/

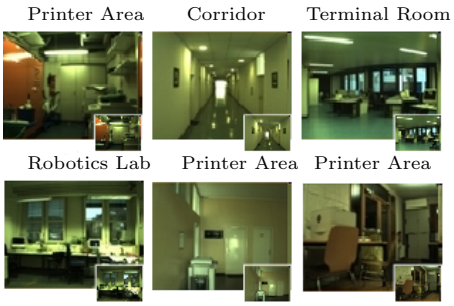

Fig. 5: Samples of the initial COLD database. The corresponding 32×24 tiny images are displayed bottom right. One can see that, despite the size reduction, these small images remain fully recognizable.

in the database and correspond to long sequences of images quite similar during the robot exploration. Some others are localized and correspond to small parts of the initial views, like edges and corners, that can be identified as room elements (*i.e.* they are not specific of a given room). The features shown in figure 4 (right) have been obtained using the normalized data. As previously observed for the van Hateren's database, the obtained features look very different. Parts of rooms are much more represented than for the whitened database and it seems that the range of spatial frequencies covered by the features is much broader. For both cases, the combinations of these initial features in higher layers correspond to larger structures more characteristic of the different rooms.

After achieving the appropriate coding based on DBNs, a classification was performed in the features space as shown in figure 6. As in [**?**] a simple regression method was used to perform the classification process in the initial case. To express the final result as a probability that a given view belongs to one room, we normalize the output with a softmax regression method. We have also investigated the classification phase using a nonlinear classifier, like Support Vector Machine (SVM). The motivation to use a nonlinear classifier is to demonstrate that the DBN computes a linear separable signature and thus deosn't affect the final classification results.

The samples taken from each laboratory and each different condition of illumination were trained separately as in [38]. For each image the softmax network output gives the probability of being in each of the visited rooms. According to maximum likelihood principles, the largest probability value gives the decision of the system. When we use features learned from the whitened data, we obtain an average of correct answers ranging from 65% to 80% according to the different conditions and laboratories as shown in figure 6 (first column). More precisely, we obtain 73.4%, 69.5%, and 71% for COLD-Ljubljana, COLD-Freiburg, and COLD-Saarbruecken laboratories respectively and with an overall average of correct answers of 71.3% for the three laboratories. In contrast, when we use features learned from the normalized data, we obtain an average of cor-

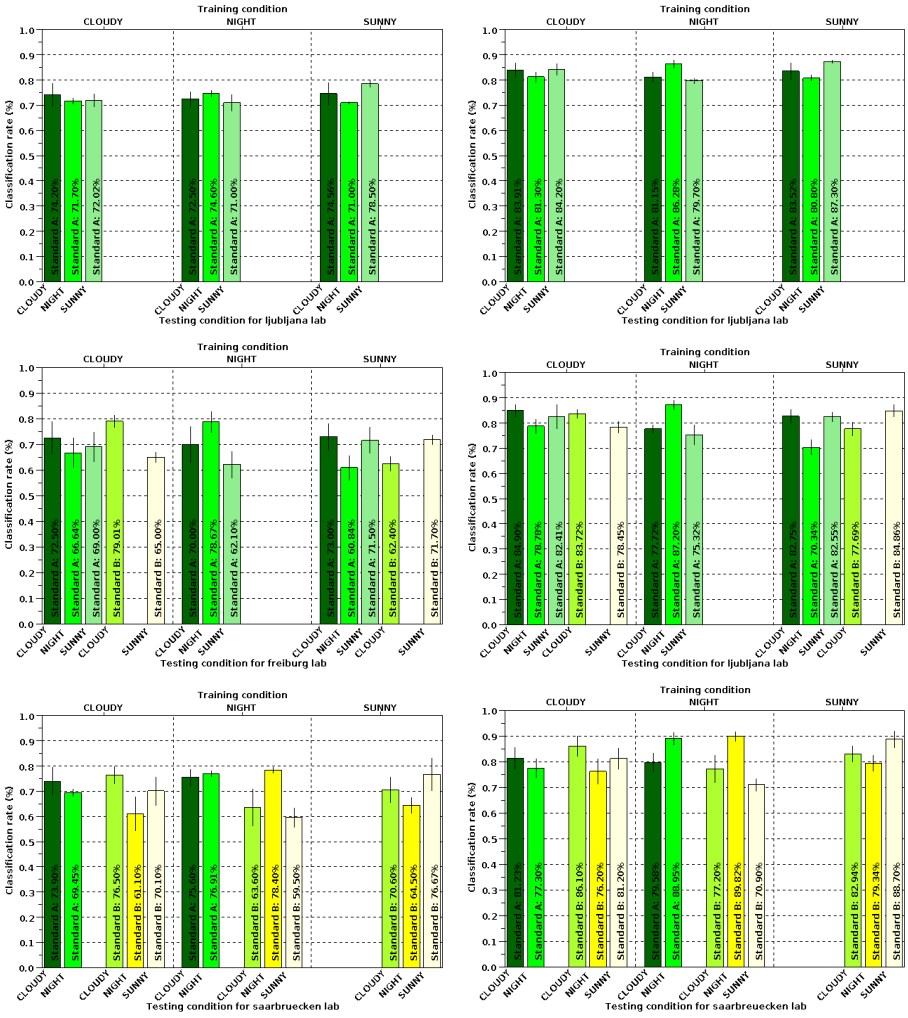

Fig. 6: Average classification rates from the three different laboratories. **Left column:** Whitened data. **Right column:** Normalized data.

rect answers ranging from 71% to 90% according to the different conditions and laboratories as shown in figure 6 (second column). More precisely, we obtain 83.13%, 80.515%, and 81.5% for COLD-Ljubljana, COLD-Freiburg, and COLD-Saarbruecken laboratories respectively and with an overall average of correct answers of 81.375% for the three laboratories. The latter results are then at the level of the best published ones [38]. The results remain robust to illumination variations as in [38].

These results demonstrate that, for classification, features trained on normalized data outperformed those obtained from an RBM trained on whitened

data. It illustrates the fact that the normalization process keeps much more information or structures of the initial views which are very important for the classification process. On the other hand, data whitening completely removes the first and second order statistics from the initial data which allows DBNs to extract higher-order features. This demonstrates that data whitening is not the optimal pre-processing method in the case of image classification. This is in accordance with the results in the literature showing that first and second order statistics based features are significantly better than higher order statistics in terms of classification [24, 4].

As in our previous works we have compared our results to the use of probabilistic thresholding. The detection rate presented in figure 6 has been computed from the classes with the highest probabilities, irrespective of the relative values of these probabilities. Some of them are close to the chance (in our case 0.20 or 0.25 depending on the number of categories to recognize) and it is obvious that, in such cases, the confidence in the decision made is weak. Thus below a given threshold, when the probability distribution tends to become uniform, one could consider that the answer given by the system is meaningless. This could be due to the fact that the given image contains common characteristics or structures that can be found in two or more classes. The effect of the threshold is then to discard the most uncertain results. Figure 7 (first column) shows the average classification results for a threshold of 0.55 (only the results where $\max_X p(X = c_k|I) \geq 0.55$, where $p(X = c_k$ is the probability that the current view I belongs to c_k, are retained). These results have been achieved using the features extracted from the whitened data. In this case, the average acceptance rate (the percentage of considered examples) ranges from 75% to 85%, depending on the laboratory, and the average results show values that outperform the best published ones [35]. When considering all the results obtained by training and testing on similar illumination conditions, we got an average classification rate of 90.68% for COLD-Saarbrucken laboratory, 89.88% for COLD-Freiburg laboratory and 90.66% for COLD-Ljubljana laboratory. Similarly to [35] results, the performance has decreased in case of the experiments under varying illumination conditions. In this case we have achieved classification rates of 83.683% for COLD-Saarbrucken laboratory, 83.14% for COLD-Freiburg laboratory and 84.62% for COLD-Ljubljana laboratory.

Similarly, we have also applied the threshold method on the results obtained in figure 6 (right) with locally normalized data. Figure 5.9 (second column) shows the average classification results using a similar threshold (0.55). In this case, the average rate of acceptance examples increases to be between 86% to 90%, depending on the laboratory, showing that more examples are used in the classification than the former experiment. Also, the average results, in this case, show scores that strongly outperform the best published one [Ullah et al., 2008]. This indicates that the linear separability of the data was significantly improved in the case of using the normalized data for features extraction.

Concerning the sensitivity to illumination for both cases, our results seem to be less sensitive to the illumination conditions compared to the results obtained

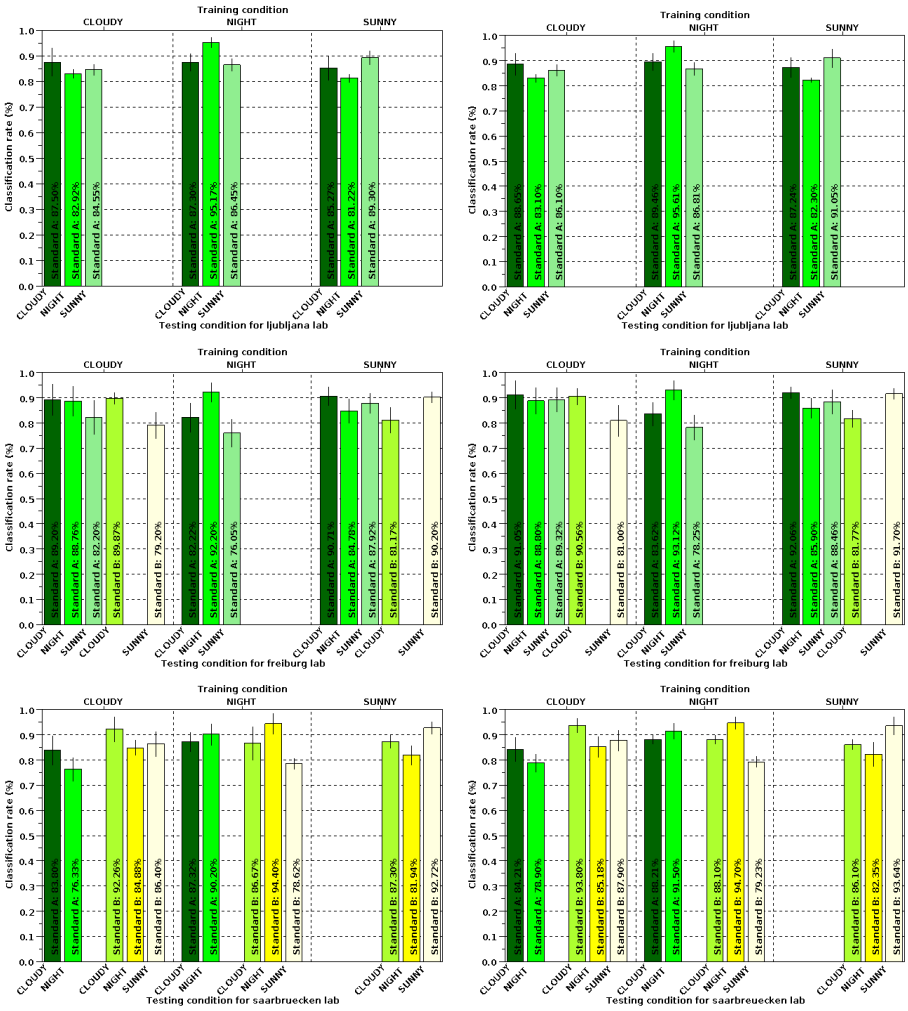

Fig. 7: Average classification rates from the three different laboratories with a threshold of 0.55. **Left column:** Whitened data. **Right column:** Normalized data.

in [Ullah et al., 2008]. As in previous experiments, we noted the lower performance on the COLD-Freiburg data, which confirms that this collection is the most challenging of the whole COLD database as indicated in [35]. However, in case of using features learned from the un-whitened data, with and without threshold our classification results for this laboratory outperforms the best ones obtained by [35].

Tables 1 and 2 show an overall comparison of our results with those from [35] for the three training conditions in a more synthetic view. It also shows the results obtained using a SVM classification instead of a softmax regression.

The results are quite comparable to softmax showing that the DBN computes a linearly separable signature. They underline the fact that features learned by DBNs approach are more robustness for a semantic place recognition task than the extraction of *ad hoc* features based on (gist, CENTRIST, SURF, SIFT).

Laboratory name	Saarbruecken			Freiburg			Ljubljana		
Condition / Training	Cloudy	Night	Sunny	Cloudy	Night	Sunny	Cloudy	Night	Sunny
Ullah	84.20%	86.52%	**87.53%**	79.57%	75.58%	77.85%	84.45%	87.54%	**85.77%**
No thr.	70.21%	70.80%	70.59%	70.43%	70.26%	67.89%	72.64%	72.70%	74.69%
SVM	69.92%	71.21%	70.70%	70.88%	70.46%	67.40%	72.20%	72.57%	74.93%
0.55 thr.	**84.73%**	**87.44%**	87.32%	**85.85%**	**83.49%**	**86.96%**	**84.99%**	**89.64%**	85.26%

Table 1: Average classification results. Whitened data. **First row:** Ullah's work; **second row:** rough results without threshold; **third row:** classification rates using a SVM classifier; **fourth row:** classification rates with threshold as indicated in text.

Laboratory name	Saarbruecken			Freiburg			Ljubljana		
Condition / Training	Cloudy	Night	Sunny	Cloudy	Night	Sunny	Cloudy	Night	Sunny
Ullah	84.20%	86.52%	**87.53%**	79.57%	75.58%	77.85%	84.45%	87.54%	85.77%
No thr.	80.41%	81.29%	83.66%	**81.65%**	**80.08%**	**79.64%**	83.14%	82.38%	83.87%
0.55 thr.	**86.00%**	**88.35%**	87.36%	**88.15%**	**85.00%**	**87.98%**	**85.95%**	**90.63%**	**86.86%**

Table 2: Average classification results. Normalized data. **First row:** Ullah's work; **second row:** rough results without threshold; **third row:** classification rates with threshold as indicated in text.

4 Conclusion and future work

Discussion on the ratio between the whole image and the patches

The aim of this paper was to investigate the role of normalization in feature extraction and image classification based on DBNs.

The main observation done in this work was that the normalization method greatly affects the spatial frequency content of the images. One immediate consequence is that the features extracted by the two methods strongly differ each from each other. We have shown that whitening suppresses the scale law usually found in natural images.

References

[1] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, MIT Press, Cambridge, MA, first edition, 2005.

[2] K. P. Soman, R. Loganathan and V. Ajay, *machine learning with SVM and other kernel methods*, PHI Learning Private Limited, M-97, New Delhi-110015, India, second edition, 2009.

[3] Y. LeCun, L. Bottou, G. Orr and K. Muller *Efficient backprop*, Neural Networks: Tricks of the Trade, Springer, 1998.

[4] A. Krizhevsky *Learning multiple layers of features from tiny images*, Master Sc. thesis, Department of Computer Science, University of Toronto, Toronto, Canada, 2009.

[5] G. E. Hinton *A Practical Guide to Training Restricted Boltzmann Machines*, Version 1, Technical report, Department of Computer Science, University of Toronto, Toronto, Canada, 2010.

[6] D. Andrzejewski *Training Binary Restricted Boltzmann Machines with Contrastive Divergence*, Technical report, Department of Computer Science, University of Wisconsih-Madison, Wisconsih, Madison, USA, 2009.

[7] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, and P. Jensfelt *The COLD Database*, Technical report, CAS - Center for Autonomous Systems. School of Computer Science and Communication, KTH Royal Institute of Technology, CVAP/CAS, Stockholm, Sweden, 2007.

[8] A. Krizhevsky *Convolutional deep belief networks on cifar-10*, Technical report, Department of Computer Science, University of Toronto, Toronto, Canada, 2010.

[9] J. Wright, Y. Ma, J. Mairal, G. Spairo, T. S. Huang, S. Yan, Sparse Representation for Computer Vision and Pattern Recognition, *Proceedings of the IEEE*, 98(6):1031-1044, Champaign, USA, 2010.

[10] J. Wu, and J. M. Rehg, CENTRIST: A Visual Descriptor for Scene Categorization, *IEEE Transaction Pattern Anal. Mach. Intell.*, 14(8):481-487, Champaign, USA, 2004.

[11] G. E. Hinton, Training products of experts by minimizing contrastive divergence, *Neural Computation*, 14(8):1771-1800, MIT Press Cambridge, MA, USA , 2002.

[12] G. E. Hinton, S. Osindero, and Y. Teh, A fast learning algorithm for deep belief nets, *Neural Computation*, 18(7):1527-1554, MIT Press Cambridge, MA, USA , 2006.

[13] P. Smolensky, Information processing in dynamical systems: foundations of harmony theory, *Parallel distributed processing: explorations in the microstructure of cognition*, 1:194-281, MIT Press Cambridge, MA, USA , 1986.

[14] B. A. Olshausen and D. Field, Sparse coding of sensory inputs, *Current Opinion in Neurobiology*, 14(4):481-487, INIST-CNRS, France 2004.

[15] F. Attneave, Some informational aspects of visual perception, *Psychological Review*, 61(3):183-193, 1954.

[16] H. Barlow, Redundancy reduction revisited, *Network: Computations in Neural Systems*, 12:241-253, 2001.

[17] D. J. Field, Relations between the statistics of natural images and the response properties of cortical cells, *Journal of Optical Society of America, A*, 4:2379-2394, 1987.

[18] A. Bell and T. J. Sejnowski, The 'Independent Components' of Natural Scenes are Edge Filters, *Vision Research*, 37:3327-3338, 1997.

[19] A. Hyvarinen and E. Oja, Independent component analysis: algorithms and applications, *Neural Networks*, 13:411-430, 2000.

[20] B. A. Olshausen and D. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *Nature*, 381(6583):607-609, 1996.

[21] A. Coates, A. Y. Ng and H. Lee, An Analysis of Single-Layer Networks in Unsupervised Feature Learning, *Journal of Machine Learning Research - Proceedings Track*, 15:215-223, 2011.

[22] M. Ranzato, A. Krizhevsky and G. E. Hinton, Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images, *Journal of Machine Learning Research - Proceedings Track*, 9:621-628, 2010.

[23] B. A. Olshausen and D. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1?, *Vision Research*, 37(23):3311-3325, 1997.

[24] N. Aggarwal and R. K. Agrawal, First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images, *Journal of Signal and Information Processing*, 3:146-153, 2012.

[25] D. J. Field, What is the goal of sensory coding?, *Neural Computation*, 6:559?601, 1994.

[26] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, Learning Mid-Level Features for Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2010), pages 2559-2566, June 13-18, San Francisco (Canada), 2010.

[27] M. A. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2007), pages 1-8, June 17-22, New York Univversity, New York (USA), 2007.

[28] J. Yang, K. Yu, Y. Gong, and T. Huang, Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2009), pages 1794-1801, June 20-25, 2009.

[29] K. Labusch and T. Martinetz, Learning Sparse Codes for Image Reconstruction. In *Proceedings of the 18^{th} European Symposium on Artificial Neural networks, Computational Intelligence and Machine Learning* (ESANN 2010), pages 241-246, April 28-30,, 2010.

[30] A. Hasasneh, E. Frenoux, and P. Tarroux, Semantic Place Recognition Based on Deep Belief Networks and Tiny Images. In *Proceedings of the 9^{th} International Conference on Informatics in Control, Automation and Robotics* (ICINCO 2012), pages 236-241, July 28-31, Rome, Italy, 2012.

[31] M. A. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, Effcient Learning of Sparse Representations With an Energy Based Model. In *Proceedings of the Advances in Neural Information Processing Systems* (NIPS 2006), pages 1137-1144, July 28-31,, 2006.

[32] A. Torralba and R. Fergus and Y. Weiss, Small Codes and Large Image Databases for Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2008), pages 1-8, June 24-26, Anchorage, Alaska, USA, 2008.

[33] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, Context-based vision system for place and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision* (ICCV 2003), pages 273-280, October 13-16, Nice, France, 2003.

[34] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, A discriminative approach to robust visual place recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 2006), pages 3829-3836, October 9-15, Beijing, China, 2006.

[35] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and H. I. Christensen, Towards robust place recognition for robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA 2008), pages 3829-3836, May 19-23, Pasadena, California, USA, 2008.

[36] D. Filliat, Interactive learning of visual topological navigation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (IROS 2008), pages 248-254, September 22-26, Nice, France, 2008.

[37] M. Dubois, H. Guillaume, E. Frenoux, and P. Tarroux, Visual place recognition using Bayesian Filtering with Markov Chains. In *Proceedings of the 19^{th} European Symposium on Artificial Neural Networks* (ESANN 2011), pages 435-440, April 27-29, Bruges, Belgium, 2011.

[38] H. Lee, C. Ekanadham and A. Y. Ng, Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems 20* (NIPS 2008), 2008.

[39] H. Lee, R. Grosse, R. Ranganath and A. Y. Ng, Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *Proceedings of the 26^{th} International Conference on Machine Learning*, pages 609-616, 2009.

[40] M. Norouzi, M. Ranjbar and G. Mori, Stacks of convolutional Restricted Boltzmann Machines for shift-invariant feature learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR 2009), pages 2735-2742, June 20-25, Fontainebleau Resort, Miami Beach, Florida, 2009.

[41] H. Guillaume, M. Dubois, E. Frenoux, and P. Tarroux, Temporal Bag-of-Words - A Generative Model for Visual Place Recognition using Temporal Integration. In *Proceedings of the 6^{th} International Conference on Computer Vision Theory and Applications* (VISAPP 2011), pages 286-295, March 05-07, Vilamoura, Algarve, Portugal, 2011.

[42] A. Krizhevsky and G. E. Hinton, Using Very Deep Autoencoders for Content-Based Image Retrieval. In *Proceedings of the 19^{th} European Symposium on Artificial Neural Networks* (ESANN 2011), April 27-29, Bruges, Belgium, 2011.

[43] G. E. Hinton, A. Krizhevsky and S. D. Wang, Transforming Auto-Encoders. In *Proceedings of the International Conference on Artificial Neural Networks* (ICANN 2011), pages 44-51, June 14-17^{th}, Espoo, Finland, 2011.