

Image Descriptor Learning Using Deep Networks

Roberto Rigamonti, *Computer Vision Laboratory (Faculté I&C), École Polytechnique Fédérale de Lausanne*

Abstract—Traditional computer vision algorithms rely on carefully handcrafted strategies, pursuing the long-term goal of replicating the effectiveness of biological visual systems in solving hard vision problems. Albeit a few remarkable exceptions to the aforementioned tendency exist, hitherto no systematic exploitation of the insights derived from the analysis of mammals' visual cortex is present in the literature. In this proposal we plan to take advantage of the increasing computational power and better models of human brain's inner mechanisms to ponder the relative importance of the guiding principles emerging from physiological recordings, in order to contrive a new type of image descriptor. After presenting three key papers in the field we substantiate our proposition, reporting the results of early investigations concerning the effect of sparsity on the recognition rate.

Index Terms—Sparse coding, image descriptors, object recognition, Deep Boltzmann Machines, Deep Belief Networks

I. INTRODUCTION

A good modeling of sensory information is a pivotal element in a digital system. Computer vision has a long history of successful approaches that gather information from visual data in order to assemble a meaningful description that effectively blends representation power with computational aspects. Classical applications, like scene [1] and surface [2]

Proposal submitted to committee: September 6th, 2010; Candidacy exam date: September 13th, 2010; Candidacy exam committee: Roger Hersch, Pascal Fua, Vincent Lepetit, Sabine Süstrunk.

This research plan has been approved:

Date: _____

Doctoral candidate: _____
(Roberto Rigamonti) (signature)

Thesis director: _____
(Pascal Fua) (signature)

Thesis co-director: _____
(Vincent Lepetit) (signature)

Doct. prog. director: _____
(Rüdiger Urbanke) (signature)

reconstruction, image segmentation [3], object detection [4, 5] and recognition [6] strongly rely on these descriptors, and often require them to be computed in a dense way [1], so practical aspects like memory footprint [7], computation reuse [1], or cache friendliness [5] are desirable properties that guide their design.

Top-scoring computer vision approaches incorporate mechanisms that allegedly provide invariance to changes in lighting conditions, scale variations, translations, rotations, and other phenomena commonly experienced in real scenes. A few of them, like the renowned SIFT [8] and DAISY [1] descriptors, roughly take inspiration from biological findings. The inclusion of these insights is, however, mostly empirically driven, so this does not prevent those algorithms to quickly saturate their potential, and improvements in performance can be achieved only by clever tunings [9, 7].

Evolution has, however, a far longer experience in shaping the visual cortex to proficiently exploit the statistical regularities exhibited by natural images. It is reasonable to think of computational neuroscience's literature as a rich source of insights that might lead striking improvements in computer vision techniques, even though physiological recordings are highly susceptible to the interpretation they are given, and so it is difficult to understand whether a given property is fundamental to achieve good results or it is just the byproduct of another one [10, 11].

Among the broad set of open problems in computer vision, we will focus our attention on visual object recognition, in that it is a classical problem that requires both an effective representation and powerful inference techniques, making it an ideal benchmark. It should be noticed, however, that the representational step is common to all the aforesaid problems, and therefore enhances in it will have wide applicability. Comparing the results achieved by state-of-the-art methods with the generalization capabilities shown by humans and the easiness by which they solve hard classification tasks [12, 13], it is clear that the key factors that drive the recognition process are still concealed, and much has to be gained by focusing on biologically-inspired techniques rather than carefully handcrafted algorithms [14].

In this proposal we set up a plan for a thorough investigation of the key properties that influence the recognition process, devising the structure of a multilayer model that, by conjugating physiological findings with machine learning and optimization principles, forms a hierarchy of nonlinear feature detectors and uses them to construct an image descriptor able to cope with the challenges posed by real environments. Our work will ground on relevant results from computational neuroscience which has shown that unsupervised generative approaches can

produce structures which exhibit properties observed in cortical cells [15, 16], and on subsequent investigations resulted in multilayer object recognition architectures that are able to learn their parameters in a unsupervised manner, trying to replicate the process by which the cortex builds a hierarchy of nonlinear feature detectors [17, 18, 19]. CVLab has already designed two effective image descriptor, namely DAISY [1] and Dominant Orientation Templates [5]. DAISY, in particular, has successfully been applied to 3D reconstruction [1], image segmentation [3], and object recognition [6] problems. Our plan is to capitalize the wisdom gathered in these years, but avoiding the needed wizardry by introducing the use of machine learning techniques.

We finally present preliminary results showing that sparsity constraints, commonly adopted in the computer vision literature, convey no relevant benefit in terms of recognition rate when a single layer architecture is employed, even though they are still needed to derive filters that closely match those observed in the V1 level of the visual cortex [15], and also play an important role in image denoising [20].

II. EFFICIENT CODING OF NATURAL IMAGES

Stemmed from information theory principles, the redundancy reduction approach to sensor modeling sought by Barlow [10] finds an empirical validation in the inspiring work by Olshausen and Field [15]. There, a generative probabilistic framework is set up in order to represent an image in an overcomplete frame of a vector space, so that each vector in the frame set $\phi_i \in \Phi$ (which we will call, from now on, *filter*) contributes to the formation of the image patch \mathbf{x} according to

$$I(\mathbf{x}) = \sum_i a_i \phi_i(\mathbf{x}) + n(\mathbf{x}), \quad (1)$$

where $n(\cdot) \sim \mathcal{N}(\mu, \sigma_N^2)$ represents the added Gaussian white noise, and $a_i \in A$ is the synthesis coefficient that weights the contribution of the i -th filter.

Overcompleteness plays a major role, since it enables the learning procedure to find the set of coefficients and filters that best explains a given input image. Redundancy reduction is achieved by imposing the use of a sparse code, which means that the algorithm will opt for the cheapest solutions in terms of the number of filters involved.

From a probabilistic perspective, the probability of an image given the frame set is

$$P(I | \Phi) = \int P(I | \Phi, A) P(A) dA, \quad (2)$$

and by the imposed noise model we have

$$P(I | \Phi, A) = \frac{1}{Z_{\sigma_N}} \exp \left(- \frac{\sum_{\mathbf{x}} \left[I(\mathbf{x}) - \sum_i a_i \phi_i(\mathbf{x}) \right]^2}{2\sigma_N^2} \right), \quad (3)$$

where Z_{σ_N} is a normalization constant and the numerator of the exponent accounts for the reconstruction error. A factorial distribution is chosen for the coefficients A , and the prior on them is super-Gaussian in order to encourage sparsity. [15]

adopts a Cauchy prior, but a Laplacian prior, corresponding to L_1 regularization, has shown to be more effective in that it has more mass around zero [20].

The learning algorithm tries to make the image model closely resemble the real model $P^*(I)$ by maximizing the average log-likelihood

$$\langle P(I | \Phi) \rangle = \int P^*(I) P(I | \Phi) dI. \quad (4)$$

Finding $\Phi^* = \arg \max_{\Phi} \langle P(I | \Phi) \rangle$ involves, however, the computation of the integral in Eq. 2, which is an intractable problem, therefore [15] makes the assumption that the integrand in Eq. 2 is sharply peaked so that it can be approximated by its maximum value, obtaining an optimization problem in the form

$$\Phi^* = \arg \max_{\Phi} \left\langle \arg \max_A [\log P(I | \Phi, A) P(A)] \right\rangle. \quad (5)$$

Trivial solutions to the approximate objective function, obtained by increasing filters' norm in order to minimize the penalty related with the coefficients norm, are avoided by constraining the norm of the filter bank to one. This ensues a slightly different optimization problem, but that has empirically shown to lead to equivalent results [21].

The task can be restated as a minimization problem in an energy-based framework

$$\Phi^* = \arg \min_{\Phi} \left\langle \arg \min_A E(I, A | \Phi) \right\rangle, \quad (6)$$

where the energy term

$$E(I, A | \Phi) = \sum_{\mathbf{x}} \left(I(\mathbf{x}) - \sum_i a_i \phi_i(\mathbf{x}) \right)^2 + \lambda \sum_i S(a_i) \quad (7)$$

gives insights about the contribution of each component: the first term on the right hand side reckon with the reconstruction error, while the second term penalizes the activity of the coefficients (the $S(\cdot)$ function depends on the prior imposed on them, being the absolute value in the Laplacian case or $\log(1+x^2)$ in the Cauchy one). The regularization parameter λ controls the trade-off between the reconstruction error and the coefficients' penalty.

The minimization procedure is articulated in two phases, where the former optimizes the coefficient values using stochastic gradient descent with a fixed filter bank, while the latter updates the filters themselves by gradient descent. To speed up the learning process, image patches are whitened in a pre-processing step, so that correlations up to second order are removed, and therefore the optimization algorithm is at ease in learning higher order dependencies.

Fig. 1 reports a filter bank obtained with the reported learning algorithm. The filters exhibit the properties of being localized, oriented, and bandpass (that is, they present different filters for similar structures at different scales), and they tile both space and spatial frequency (even though they tend to forgather in the high frequency band).

The analysis performed in [15] presents several weaknesses:

- it restricts to a single-layer linear model. Multilayer architectures are able to take into account interdependencies

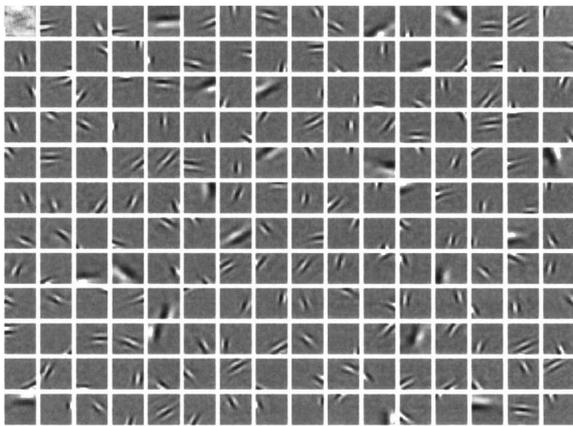


Fig. 1. Filter bank obtained by applying the learning algorithm on 16×16 patches extracted from natural images [16]. Several shifted copies of the same filter are present, owing to the fact that the algorithm is not translation-invariant.

between extracted features, albeit the introduction of nonlinearities is required in order to get benefit of them.

- the presented system is an instance of a so-called “decoder-only” architecture, in that it has no way to deduce the sparse representation for a given input image other than performing a costly iterative optimization procedure.
- the filters are learned without taking into account shifts that are common in images, resulting in a filter bank that is populated by equivalent filters that differ just by a translation.

The first issue is by far the most critical one, therefore the following Section will be devoted to the analysis of a recent proposal that tries to overcome this point.

III. LEARNING ALGORITHMS FOR DEEP ARCHITECTURES

Experimental evidence strongly suggests that the visual cortex is structured as a multilayer generative model [22, 23]. This is justifiable, from an analytical standpoint, by observing that most functions having a compact representation in a model constituted by several layers, require an exponentially larger number of components for being represented in a shallow architecture [24, 25]. Hence, besides obvious computational issues, a shallow model implies that huge training sets are required in the learning phase. Massive efforts were therefore invested in the development of deep machines, though their training has been regarded for a long time as a prohibitively hard optimization problem [26], because the learning algorithm has to recur to Markov Chain Monte Carlo techniques or variational approximations in order to compute the quantities of interest [27, 28].

A milestone in the generative deep networks’ literature is represented by *Boltzmann Machines* (BMs) [26, 28] (see Fig. 2 left). A BM is a network of binary, stochastic units with an energy associated to each configuration of the network. Units are split in a visible set \mathbf{v} and a hidden set \mathbf{h} , and the

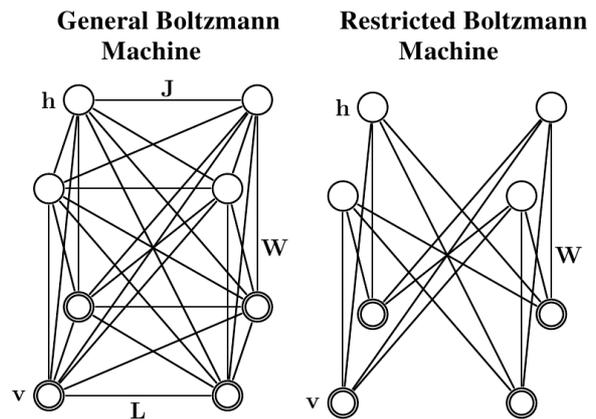


Fig. 2. A graphical representation of a general BM (left) and of a RBM (right) [17]. J represents the matrix of weights between hidden units, L the matrix of weights between visible units, and W the matrix of inter-layer weights.

probability assigned to an input vector $\bar{\mathbf{v}}$ is

$$p(\bar{\mathbf{v}}; \theta) = \frac{p^*(\bar{\mathbf{v}}; \theta)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp[-E(\bar{\mathbf{v}}, \mathbf{h}; \theta)], \quad (8)$$

where

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h}; \theta)] \quad (9)$$

is called *partition function*, $p^*(\cdot)$ denotes the unnormalized probability, $E(\mathbf{v}, \mathbf{h}; \theta)$ represents the energy of the configuration $\{\mathbf{v}, \mathbf{h}\}$, and θ is the parameter vector containing the weights and the biases of the model.

The gradient required for maximizing the log-likelihood of the data under the model has a formulation that is both local and easy, and simply accounts for changing the weights proportionally to the difference in the generated probability distributions [26]. Despite this, learning in BMs can only be approximated: a BM is a directed network, therefore it suffers from a phenomenon called *explaining away* [29], and moreover the partition function involves the summation of infinitely many terms [30].

BMs are, for these reasons, topologically altered by removing all the intra-layer connections. The resulting architecture, called *Restricted Boltzmann Machine* (RBM) (see Fig. 2 right), exhibits the nice property of having the hidden states conditionally independent given the visible states, and vice versa, making sampling from an RBM extremely efficient [31]. Besides, RBMs are shown to be equivalent to an infinite depth directed network where complementary priors have been used to remove the explaining away phenomenon that hinders exact inference in BMs [32].

A sensible objective for the learning is the maximization of the log-likelihood \mathcal{L} of input vectors. Intuitively, the idea is to lower the energy assigned to input vectors, raising at the same time the energy assigned to model’s fantasies (that is, vectors generated according to network’s internal representation). If we let W represent the weights between visible and hidden units, the required gradient is given by

$$\frac{\partial \mathcal{L}}{\partial W} = \mathbb{E}_{P_{data}} [\mathbf{v}\mathbf{h}^T] - \mathbb{E}_{P_{model}} [\mathbf{v}\mathbf{h}^T], \quad (10)$$

where

- $\mathbb{E}_{P_{data}}[\cdot]$ is the expectation with respect to the completed data distribution $P_{data}(\mathbf{v}, \mathbf{h}; \theta) = p(\mathbf{h} | \mathbf{v}; \theta) P_{data}(\mathbf{v})$, with $P_{data}(\mathbf{v}) = \frac{1}{N} \sum_n \delta(\mathbf{v} - \mathbf{v}_n)$ representing the empirical data distribution.
- $\mathbb{E}_{P_{model}}[\cdot]$ is the expectation with respect to the model's distribution of Eq. 8.

Since it is trivial to sample from $P_{data}(\mathbf{v})$ and $p(\mathbf{h} | \mathbf{v}; \theta)$ follows a factorial distribution, the former expectation poses no challenges in an RBM, but the latter expectation is intractable because it involves sampling from $P_{model}(\mathbf{v}, \mathbf{h})$, and this requires the Gibbs sampler to approach its stationary distribution [33].

RBM's are suitable for the exploitation of a particularly effective learning technique called *Contrastive Divergence* (CD) [34], which approximates the stationary distribution of the Gibbs Markov Chain with the distribution after one step of sampling [35]. Gradient's direction is not exact, but the data generated after one step of Gibbs sampling still captures the idea the model has of the world, and therefore the changes usually point in a good direction.

A more recent proposal, the *Persistent Contrastive Divergence* (PCD) algorithm [36], takes advantage of the fact that the model changes slightly between parameter updates by initializing the Markov Chain at the last state reached for the previous model. A sufficient condition for convergence is that the learning rate is sufficiently small compared with the mixing rate of the Markov Chain [37]. [17] suggests that many of these persistent chains might be run in parallel in order to estimate model's expectations also in general BMs, and refers to each of these chains as a *fantasy particle*.

Remaining in the context of general BMs, [17] proposes to rewrite a known lower bound on the log-likelihood of an input vector in a BM [38] in a form suitable to perform a mean-field approximation [39] of the posterior distribution $p(\mathbf{v}, \mathbf{h}; \theta)$. This posterior could not be computed explicitly owing to the aforementioned explaining away phenomenon, but is required to compute the data-dependent expectations required in the learning procedure. The full training algorithm for BMs obtained in this way is, however, not efficiently extendable to a multilayer setting. Stacking a set of BMs to form a deep generative network is in fact an hard task, and a learning algorithm relying on PCB and a mean-field approximation would be prohibitively slow.

Nonetheless, it is possible to train a set of RBMs separately using CD learning, and then stack them in order to form an architecture where the top layers use undirected connections whereas layers below present directed connections, leading to a so-called *Deep Belief Network* (DBN) model [32] (see Fig. 3, left panel). A DBN is not an RBM anymore, because the lower part is a directed generative model. An important variation of the DBN model is presented in [17] under the name *Deep Boltzmann Machine* (DBM) (see Fig. 3, right panel). The main difference is that a DBM incorporates top-down connections also in the lower layers, and therefore the learning procedure for the overall model can benefit both from a bottom-up pass and from a top-down feedback when dealing with ambiguous

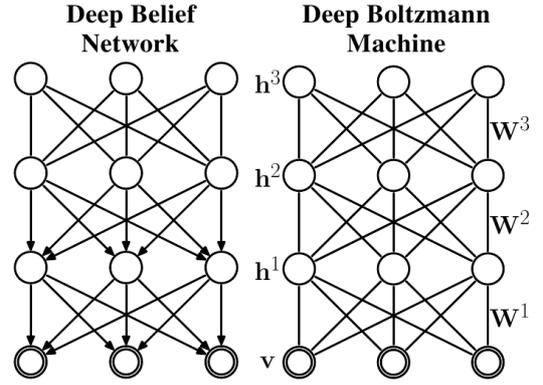


Fig. 3. Two examples of deep machines, a Deep Belief Network [32] (left) and a Deep Boltzmann Machine [17]. The lowest layers of a DBN are directed, whereas a DBM preserves the undirected connections of the constituent RBMs.

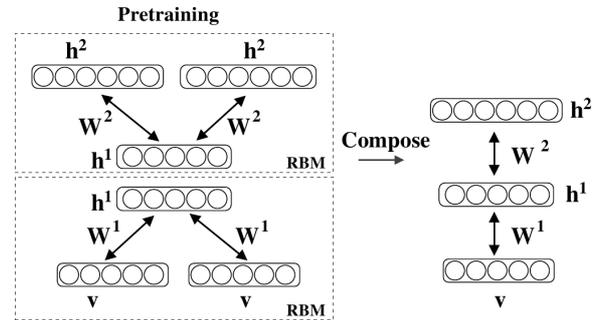


Fig. 4. Detail on the DBM training algorithm. On the left, two separate RBMs are pre-trained for level 1 and 2 of the DBM shown in Fig. 3, and the resulting machines are merged in the architecture shown on the right [17].

inputs.

The key observation is that, when the first RBM is learned (see Fig. 4 left), the generative model has the form

$$p(\mathbf{v}; \theta) = \sum_{\mathbf{h}^1} p(\mathbf{h}^1; W^1) p(\mathbf{v} | \mathbf{h}^1; W^1), \quad (11)$$

but the first term in the summation, $p(\mathbf{h}^1; W^1) = \sum_{\mathbf{v}} p(\mathbf{h}^1, \mathbf{v}; W^1)$ can be replaced by $p(\mathbf{h}^1; W^2) = \sum_{\mathbf{h}^2} p(\mathbf{h}^1, \mathbf{h}^2; W^2)$ which represents a better model of the posterior distribution over \mathbf{h}^1 (once the second RBM is properly trained, and assuming a correct initialization of W^2). A good idea would be to average the models of \mathbf{h}^1 coming bottom-up and top-down, but since \mathbf{h}^2 depends on \mathbf{v} , this would result in double-counting the evidence. Visible units are therefore replicated in the lowest layer, and CD learning is performed by keeping the weight matrices tied together as shown in Fig. 4. The same trick is applied for the hidden units of the top layer. When the two layers are finally combined, the resulting architecture halves the contributions to \mathbf{h}^1 coming from the top and the bottom, obtaining an undirected model that still presents symmetric weights. Training a deeper networks involves performing this replication on the two extrema only, since the same result can be achieved in intermediate layers just by halving the weights in both directions.

DBMs exhibit an additional, attractive property, namely it is possible to evaluate efficiently the partition function by using

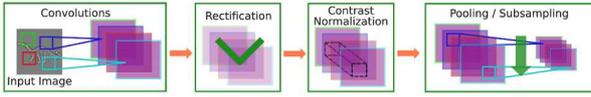


Fig. 5. Typical convolutional feature extraction layer, as introduced in [19].

Annealed Importance Sampling (AIS) [30]. Considering again the architecture of Fig. 4, since it is possible to analytically sum out the contributions of \mathbf{v} and \mathbf{h}^2 , AIS can be run on a small state space, leading to a reduced computational cost. Once the global partition function is estimated, the lower bound on the log-likelihood of the data reported in [38] can be computed and optimized by means of a mean-field approximation. This leads to a principled way of evaluating the generative power of a deep network, different from the rough measure derived by taking, for example, the recognition rate of a classification system built on top of the model.

The model introduced in [30] is based on RBMs, and therefore inherits their major weakness, that is, the difficulty of incorporating application-dependent prior knowledge (RBMs are, by definition, generic learning machines). In particular, DBMs are known to focus on modeling long-range dependencies in the input, which are not relevant when analyzing, for example, natural images, but fail to preserve its topological structure, which conveys a strong prior in the case of images. Some approaches have been proposed in order to solve these issues (see, e.g., [40]), but no definitive solution has been proposed thus far. For this reason in the next Section we will focus on the simpler model introduced in [19], which is more suitable for the analysis of image-related properties in the particular context of object recognition.

IV. MULTILAYER ARCHITECTURES FOR OBJECT RECOGNITION

A careful investigation of the building blocks of a multi-stage object recognition architecture is presented in [19]. Different filter banks, including learned filters, hardcoded, and random ones are tried, followed by a nonlinear transform, a normalization, and a pooling step. The guiding principle is the same adopted in [7], that is, analyze how the recognition rate is affected by the choices performed at the different steps. A remarkable distinction with previous work is that two-layer architectures are also examined in this practical fashion. Fig. 5 reports an example of a feature extraction layer.

A recent unsupervised learning technique, named *Predictive Sparse Decomposition* (PSD) [41] is adopted in order to approximate in an effective way the encoder that is missing in Olshausen and Field’s approach [15], avoiding in this way the costly iterative refinement required to get a sparse representation. This is achieved by simultaneously learning the filter bank used in the feature extraction and a nonlinear regressor used to approximate, at run-time, the sparse code corresponding to the input image.

The presented evaluation hinges on three points, namely the effect of the nonlinear transform, of the chosen filter bank, and of an additional feature extraction layer on the final classification score. Albeit the effectiveness of the second

layer of processing was easily predictable given the theoretical results available in literature, the proposed system achieves a remarkable results even when supervised learning is applied on the Caltech-101 dataset [42], which contradicts the most recent findings from the machine learning community [43]. A plausible explanation given in [19] is that the poor scores reported by supervised algorithms on this dataset are owing to incorrect choices for the nonlinear stage. Another noteworthy result is that randomly generated filters achieve good results when training set’s size is reduced (which is the case of Caltech-101, but not of the NORB dataset [44]).

As presented in the next Section, we have independently developed an architecture close to the one presented in [19] and observed similar outcomes, though we have focused on the analysis of different properties. Some results reported in this paper are not compatible with those we have obtained, for example much relevance is posed in [19] on the normalization step, while we have observed negligible improvements by introducing the local contrast normalization [12] in different stages of our model. This, however, might be related to the different dataset we focused on (the CIFAR-10 dataset [31] we employed is constituted by tiny images, where normalizations might entail losses in the local structure and therefore affect negatively the recognition rate).

V. EXPERIMENTAL RESULTS AND RESEARCH PROPOSAL

As Olshausen and Field’s analysis demonstrates [15], sparsity is a key property for the derivation of a set of feature extractors that exhibit properties akin to those observed in cortical cells.

At the beginning of our inquiries we were concerned about the role that sparsity plays on the final recognition score when a shallow object classification architecture is employed. We therefore extended the approach presented in [15] in a convolutional sense, obtaining an objective function in the form

$$\min_{\Phi, \{a_{ij}\}} \sum_i \left(\left\| \mathbf{x}_i - \sum_j \phi_j * a_{ij} \right\|_2^2 + \lambda \sum_j \|a_{ij}\|_1 \right), \quad (12)$$

where \mathbf{x}_i are the training images, ϕ_j are linear filters, and a_{ij} can now be seen as a set of images with the same size as the \mathbf{x}_i images, whose cardinality is equal to that of the filter bank. Similar intermediate representations have been called *feature maps* in the Convolutional Neural Networks literature [45]. The filter banks learned optimizing this objective function using stochastic gradient descents on the CIFAR-10 [31] and on the Caltech-101 [42] datasets are reported in Fig. 6.

At that point, we set up a recognition architecture structured on several stages, similarly to what Brown *et al.* do in [7], to ponder the effectiveness of the different choices available in literature for each stage and to estimate the value of a sparse representation in terms of correct classification score. The system we devised presents the following steps

- *pre-processing* : input images are converted to grayscale, whitened, and properly expanded in order to avoid border effects in convolutions.

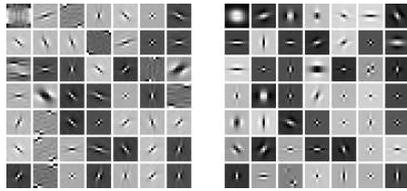


Fig. 6. Filter banks derived by the convolutional learning algorithm on the CIFAR-10 dataset (left panel) and on the Caltech-101 dataset (right panel). The different appearance of the filters might be related to the different scales of the images present in the datasets (the CIFAR-10 comprises 32×32 pixel images only, whereas the Caltech-101 dataset contains arbitrarily sized images).

- *feature extraction* : pre-processed images are convolved with the filter banks shown in Fig. 6 and other hardcoded filter banks taken from the literature.
- *feature refinement* : feature maps are refined using different techniques
 - no refinement (leave feature maps untouched).
 - gradient descent without enforcing any sparsity (equivalent to setting the λ parameter in Eq. 12 to zero).
 - gradient descent with different values for the regularization parameter.
 - Matching Pursuit [46], which is deemed to provide the sparsest representation of a given image.
- *non-linearity introduction* : feature maps resulting from the previous step are transformed using different nonlinear operators like absolute value or sigmoid.
- *pooling* : invariance to small displacements and distortions is introduced by pooling feature maps. Three different operators, namely Gaussian, boxcar, and MAX pooling are tried.
- *subspace projection* : pooled feature maps are projected on different subspaces using standard techniques like PCA, LDE [47], and Random Projections [48].
- *classification* : a descriptor is formed by concatenating the projected feature maps, and different choices for the classifier are explored. The most relevant are the Nearest Neighbor classifier, which conveys the real discriminative power of the representation, and Support Vector Machines, that led the best results.

The surprising result that emerged from our investigations is that enforcing sparsity in the feature refinement step does not help the final recognition rate, and at best leads to equivalent results. This was empirically verified by running the classification architecture on the CIFAR-10 dataset with hundreds of different parametrizations. The most effective combination of techniques for the different stages presents learned filters in the feature extraction stage, no refinement, a non-linearity that splits positive from negative parts (reversing the sign of the latter), Gaussian pooling with $\sigma = 3$, subspace projection using LDE over at most 256 eigenvectors (the exact number is obtained by an extensive search in the $[8 - 256]$ range with NN classification rate as performance measure), and classification using an SVM. Our best performing architecture achieved a 75.18% recognition rate (averaged over 5 random splits of

the training set, $\sigma = 0.2688$), that overwhelms the 64.84% classification score reported by the creators of the dataset in [31], and also the current state-of-the-art performance (71%) obtained in [49] by using a third-order Boltzmann machine and exploiting both color images and an enlarged, unlabeled version of the training set.

The research performed in the last year explored a restricted subset of the architectural issues involved with the design of a new descriptor type, and vastly focused on surveying the literature of the three main fields our research covers (computer vision, machine learning, and computational neuroscience). We can group the key aspects we plan to face in the future in the following list:

- We submitted the first part of the work we did to the last European Conference on Computer Vision, and received good scores and an encouraging feedback. Grounding on the suggestions received, at first we would like to validate our results over other widely used datasets, in particular the Caltech-101 [42] and NORB [44] ones.
- Zeiler et al. in [50] presented an interesting approach that might overcome some of the drawbacks related with the optimization based on stochastic gradient descent we employed, and we believe that an accurate comparison of the results achieved might lead interesting insights on the future directions to pursue.
- No strategies are adopted thus far in order to adapt to contrast changes, while this point is very important for practical applications. The fact that the learned features for the first layer are close to be gradient detectors (see Fig. 6) helps, but this is probably not enough.
- In light of the analysis performed in the previous sections, the adoption of multiple layers appears to be a mandatory structural choice.¹ This, however, implies that a learning algorithm suitable for the needs of real-time recognition has to be contrived. Moreover, a principled way of incorporating prior knowledge on the vision domain in a deep network has to be investigated.

The last stage of our proposal involves the integration of the resulting image descriptor in the different application settings, properly accounting for the constraints exhibited by each context.

REFERENCES

- [1] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.
- [2] Aydin Varol, Mathieu Salzmann, Engin Tola, and Pascal Fua. Template-Free Monocular Reconstruction of Deformable Surfaces. In *Proc. of the Int. Conf. on Comput. Vis.*, 2009.
- [3] Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Vincent Lepetit, and Pascal Fua. A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images. In *Med. Image. Comput. Assist. Interv.*, 2010.

¹Please note that the architecture we developed might be interpreted as a two-layer architecture, where the first layer performs feature extraction while the second layer achieves invariance through a pooling step. This view accords with Hubel and Wiesel's distinction between simple and complex cells [23]. We have, however, preferred Hinton's interpretation where each layer performs a feature extraction over the preceding layer's output, obtaining higher level abstractions.

- [4] Karim Ali, François Fleuret, David Hasler, and Pascal Fua. Joint Pose Estimator and Feature Learning for Object Detection. In *Proc. of the Int. Conf. on Comput. Vis.*, 2009.
- [5] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Pascal Fua, and Nassir Navab. Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2010.
- [6] Mustafa Özuysal, Vincent Lepetit, and Pascal Fua. Pose Estimation for Category Specific Multiview Object Localization. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2009.
- [7] Matthew Brown, Gang Hua, and Simon Winder. Discriminative Learning of Local Image Descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.
- [8] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 2004.
- [9] Simon Winder, Gang Hua, and Matthew Brown. Picking the best DAISY. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2009.
- [10] Horace B. Barlow. *Possible principles underlying the transformations of sensory messages*, chapter 13, pages 217–234. MIT Press, 1961.
- [11] Li Zhao. Is sparse and distributed the coding goal of simple cells? *Biol. Cybern.*, 2004.
- [12] Nicolas Pinto, David D. Cox, and James J. DiCarlo. Why is Real-World Visual Object Recognition Hard? *PLoS Comput. Biol.*, 2008.
- [13] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.
- [14] Shimon Edelman and Nathan Intrator. Unsupervised statistical learning in vision: computational principles, biological evidence. In *Proc. of the Europ. Conf. on Comput. Vis.*, 2004.
- [15] Bruno A. Olshausen and David J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Res.*, 1997.
- [16] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 1996.
- [17] Ruslan R. Salakhutdinov and Geoffrey E. Hinton. Deep Boltzmann Machines. In *Proc. of the Int. Conf. on Artif. Intell. and Statist.*, 2009.
- [18] Geoffrey E. Hinton. Learning to represent visual input. *Phil. Trans. R. Soc. B*, 2010.
- [19] Kevin Jarret, Koray Kavukcuoglu, Marc’Aurelio Ranzato, and Yann LeCun. What is the Best Multi-Stage Architecture for Object Recognition? In *Proc. of the Int. Conf. on Comput. Vis.*, 2009.
- [20] Michael S. Lewicki and Bruno A. Olshausen. Probabilistic framework for the adaptation and comparison of image codes. In *J. Opt. Soc. Am.*, 1999.
- [21] Bruno A. Olshausen and Jarrod K. Millman. Learning sparse codes with a mixture-of-Gaussians prior. In *Adv. Neural Inf. Process. Syst.*, 2000.
- [22] Geoffrey E. Hinton. Learning multiple layers of representation. *Phil. Trans. R. Soc. B*, 2007.
- [23] David H. Hubel and Torsten N. Wiesel. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat’s Visual Cortex. *J. Physiol.*, 1962.
- [24] Yoshua Bengio and Yann LeCun. *Scaling Learning Algorithms towards AI*. MIT Press, 2007.
- [25] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy Layer-Wise Training of Deep Networks. In *Adv. Neural Inf. Process. Syst.*, 2006.
- [26] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A Learning Algorithm for Boltzmann Machines. *Cognitive Sci.*, 1985.
- [27] Geoffrey E. Hinton and Terrence J. Sejnowski. Optimal perceptual inference. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 1983.
- [28] Geoffrey E. Hinton and Terrence J. Sejnowski. *Learning and Relearning in Boltzmann Machines*, volume 1, chapter 7, pages 282–317. MIT Press, 1986.
- [29] Michael P. Wellman and Max Henrion. Explaining “Explaining Away”. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1993.
- [30] Ruslan R. Salakhutdinov and Iain Murray. On the Quantitative Analysis of Deep Belief Networks. In *Proc. of the Int. Conf. on Mach. Learn.*, 2008.
- [31] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Master’s thesis, University of Toronto, 2009.
- [32] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computat.*, 2006.
- [33] Ilya Sutskever and Tijmen Tieleman. On the Convergence Properties of Contrastive Divergence. In *Proc. of the Int. Conf. on Artif. Intell. and Statist.*, 2010.
- [34] Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computat.*, 2002.
- [35] Miguel Á. Carreira-Perpiñán and Geoffrey E. Hinton. On Contrastive Divergence Learning. In *Proc. of the Int. Conf. on Artif. Intell. and Statist.*, 2005.
- [36] Tijmen Tieleman. Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. In *Proc. of the Int. Conf. on Mach. Learn.*, 2008.
- [37] Alan Yuille. The Convergence of Contrastive Divergences. In *Adv. Neural Inf. Process. Syst.*, 2004.
- [38] Radford M. Neal and Geoffrey E. Hinton. *A view of the EM algorithm that justifies incremental, sparse, and other variants*. MIT Press, 1998.
- [39] Carsten Peterson and James R. Anderson. A Mean Field Theory Learning Algorithm for Neural Networks. *Complex Syst.*, 1987.
- [40] Andreas Müller, Hannes Schulz, and Sven Behnke. Topological Features in Locally Connected RBMs. In *Proc. of the Int. Joint Conf. on Neural Networks*, 2010.
- [41] Koray Kavukcuoglu, Marc’Aurelio Ranzato, and Yann LeCun. Fast Inference in Sparse Coding Algorithms with Applications to Object Recognition. Technical report, The Courant Institute of Mathematical Sciences, New York University, 2008.
- [42] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2004.
- [43] Yoshua Bengio. *Learning Deep Architectures for AI*, volume 2, pages 1–127. Now Publishers, 2009.
- [44] Yann LeCun, Fu J. Huang, and Léon Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2004.
- [45] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proc. of the IEEE*, 1998.
- [46] François Bergeaud and Stéphane Mallat. Matching Pursuit of Images. In *Proc. of the Int. Conf. on Image Processing*, 1995.
- [47] Gang Hua, Matthew Brown, and Simon Winder. Discriminant Embedding for Local Image Descriptors. In *Proc. of the Int. Conf. on Comput. Vis.*, 2007.
- [48] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proc. of the ACM int. conf. on Knowledge discovery and data mining*, 2001.
- [49] Marc’Aurelio Ranzato and Geoffrey E. Hinton. Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2010.
- [50] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional Networks. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recogn.*, 2010.