Unsupervised Category Discovery in Images Using Sparse Neural Coding

Stephen Waydo and Christof Koch
Control & Dynamical Systems and Computation & Neural Systems
California Institute of Technology
Pasadena, CA 91125, USA
waydo@cds.caltech.edu

Abstract

We present an unsupervised method for learning and recognizing object categories from unlabeled images. Motivated by the existence of highly selective, sparsely firing cells observed in the human medial temporal lobe (MTL), we apply a sparse generative model to the outputs of a biologically faithful model of the primate ventral visual system. In our model, a network of nonlinear neurons learns a sparse representation of its inputs through an unsupervised expectation-maximization process. In recognition, this model is used in a maximum-likelihood manner to classify unseen images, and we find units emerging from learning that respond selectively to specific image categories. A significant advantage of this approach is that there is no need to specify the number of categories present in the training set. We present classification accuracy using three different evaluation metrics.

1 Introduction

Highly sparse representations of objects in the visual environment in which individual neurons display a strong selectivity for only one or a few stimuli (such as familiar individuals or landmark buildings) out of perhaps 100 presented to a test subject have been observed in the human medial temporal lobe (MTL), a brain area crucial to the formation of new memories [11, 18]. While highly selective for a particular object or category, these cells are remarkably insensitive to different presentations (i.e. different poses and views) of their preferred stimulus. By contrast, neurons in the inferotemporal cortex (IT), immediately earlier in the visual pathway, respond in a much less sparse manner [14]. A natural question to ask is thus "how do neurons in the MTL learn their sparse and invariant representations from the incoming visual information?" From a machine vision standpoint, this question can be viewed as a problem in unsupervised image classification: given a set of unlabeled training images, can we design an algorithm that will group these images into categories corresponding to those human observers would impose? This is clearly distinct from the more common approach to object recognition in which a labeled training set is used to learn features common to the category which can then be used to classify unlabeled images [1, 4].

Motivated by the neurobiological results, we study the effects of applying a sparse-coding model to the outputs of a biologically faithful model of the primate ventral visual

BMVC 2007 doi:10.5244/C.21.102

cortex [13, 15]. The sparse-coding model, which itself employs biologically plausible learning operations, is derived from that of Olshausen & Field [10], which they used to develop a sparse representation of natural images much like that observed in primate visual cortex. We seek to use a similar learning algorithm to build a representation in which individual units of our output layer respond in a selective and invariant manner to specific object categories.

1.1 Related Work

Unsupervised image classification has only recently begun to attract attention in the literature. Sivic et al. [17] apply techniques from unsupervised topic discovery in text to "words" derived from SIFT descriptors to discover categories in images. While their approach is very different from that taken here, the problem they attempt to solve is the same (and we evaluate our results on many of the same datasets). An important distinction is that they found it important to restrict the number of categories searched for to the number truly present in their datasets, while our method is robust to varying numbers of input categories. Fergus, Perona, and Zisserman [4] use an unsupervised generative learning algorithm to build representations of particular image categories, but only images from a single category are presented to the model, which is then tested in a category-versus-background setting. In contrast, our model simultaneously learns representations for multiple image categories without *a priori* specification of the labels (or even the number of categories present). Weber, Welling, and Perona [19] also cast the unsupervised categorization problem as emergent population coding, but without the sparseness constraint that is key to our results. Serre, Wolf, and Poggio [16] developed the underlying vision system model we use here, and they show that the features generated are sufficient to classify our input categories with high accuracy (using a supervised classifier).

Sparse coding as a computational tool has attracted a great deal of attention in recent years, both in the context of vision and elsewhere. Olshausen and Field developed the algorithm we apply here and showed that, when applied to natural image patches, it generates a code much like that observed in simple cells in primary visual cortex [9, 10]. Hinton and Ghahramani [6] also cast sparse representation in a generative modeling framework, but as with Olshausen and Field they work directly at the image level. Sparse coding is closely related to *Independent Components Analysis* [2], which has been used to generate natural image codes similar to those obtained from sparse coding [3]. Li et al. [7] discuss the use of sparse representation for blind source separation, including the notion that the number of sources (in our nomenclature, categories) need not be specified, but they do not address the application we present here. Mutch and Lowe [8] improve the performance of the underlying vision system model we use here, in part using sparsification to enhance selectivity. Ranzato et al. [12] take an energy-based approach to the unsupervised learning of sparse representations of natural images and briefly discuss its extension to a hierarchical model. Both of these efforts are at a much lower level of the hierarchy and so do not address categorization.

2 Approach

We first generate an invariant feature-based representation of our images (analogous to that found in IT) using the hierarchical feedforward model of object recognition described

by Serre et al. [15] and available at `http://cbcl.mit.edu`. The output of this stage - applied to many images from several different image categories - is then sent into a sparse coding model (modified from [10]). This network attempts to identify sparse structure in its inputs via unsupervised learning on sample input data. To evaluate performance we examine the selectivity of the trained network to unseen images from the same categories as the training data.

2.1 Input Processing

All images used in this investigation were taken from the Caltech-256 database of images from 256 categories [5]. Images were resized (using MATLAB's `imresize` with nearest-neighbor interpolation) so that the smaller dimension was 128 pixels while preserving the aspect ratio. The outputs of the C2b and C3 layers of the visual processing model [15] were computed using a feature set derived from training on 500 natural images (no new features were learned - this investigation used the filters included in the standard distribution of this model). There were 1000 units in each of these layers, for a total of 2000 outputs. These outputs were then normalized so that each output unit's responses had zero mean and unit variance across the input set for a given experiment. These normalized outputs were used as inputs to the sparse coding model described below.

2.2 Sparse Coding

We seek to build a generative model $\mathscr{G}$ of the inputs $u \in \mathbb{R}^n$ (here, $n = 2000$) to our model with the assumption that there exists some sparse set of causes $v \in \mathbb{R}^m$ (with $m \ll n$) underlying the observed data. In our case the u are the responses of the underlying vision system model to the input images, while each element v_i of v will come to represent an image category. In general, we wish to find probability density functions $f(v|\mathscr{G})$ and $f(u|v,\mathscr{G})$ such that the distribution of generated inputs

$$f(u|\mathscr{G}) = \int_v f(u|v,\mathscr{G}) f(v|\mathscr{G}) \tag{1}$$

closely matches $f(u)$, the distribution of inputs observed in the training data. Once such distributions have been found, we can attribute causes to inputs by a deterministic maximum-likelihood process, or

$$v(u) = \arg\max_v f(v|u,\mathscr{G}). \tag{2}$$

Following the approach of Olshausen & Field [10], we can use this framework to search for a sparse code for our inputs. First, we assume the causes underlying the inputs are sparse and independent, setting

$$f(v|\mathscr{G}) \propto \prod_{i=1}^{m} \exp(S(v_i)), \tag{3}$$

where $v_i \in \mathbb{R}$ is the i^{th} element of v and $S(v_i)$ is defined such that the resulting distribution is sparse. For simplicity we omit the proportionality constant required to make this distribution integrate to 1. In [10], where this strategy was used to develop a V1-like sparse code for natural images, the sparse prior S followed a Cauchy distribution. Because we

seek to develop units that respond in a more-or-less binary fashion (i.e. most responses are close to 0 or 1), we instead use a weighted sum of two Gaussians with variance σ^2, one centered at 0 with weight $1-t$ and the other at 1 with weight t.

Second, we assume that the distribution of inputs given a cause is Gaussian with a mean given by a linear function of the causes, that is $E[u] = Gv$ for some $G \in \mathbb{R}^{n \times m}$, and diagonal covariance matrix $COV[u] = \lambda I$. The columns of G are thus basis functions for representing the inputs u. We further place a zero-mean Gaussian prior distribution with variance γ^2 on the elements g_{ij} of G to avoid an extra normalization step required in earlier work.

Our generative model $\mathscr{G}$ is now parameterized by the matrix G. The function to be maximized with respect to G is the average log-likelihood of the data within the model,

$$\begin{aligned} \mathscr{F}(v(u), G) &= \langle \ln f(v(u), u, G) \rangle \\ &= \left\langle -\frac{1}{2\lambda}\|u - Gv(u)\|^2 + \sum_{i=1}^{m} S(v_i(u)) - \frac{1}{2\gamma}\sum_{i=1}^{n}\sum_{j=1}^{m} g_{ij}^2 \right\rangle. \end{aligned}$$

The first term in $\mathscr{F}$ penalizes a mismatch between the true input u and the modeled input $Gv(u)$, the second term rewards responses that are likely according to the sparse prior, and the third term penalizes large weights in G.

We optimize this function via expectation maximization. In the E phase, for each input u we seek to compute the most likely cause $v(u)$ (i.e. the arg max of $\mathscr{F}$). Performing gradient ascent on $\mathscr{F}$ with respect to v we obtain the differential equation

$$\dot{v} = \frac{1}{\lambda} G^T (u - Gv) + S'(v) \tag{4}$$

where the vector-valued function $S(v)$ is shorthand for S evaluated on each v_i and S' is the derivative of S with respect to v. This system can be implemented as a two-layer recurrent neural network with nonlinear dynamics in the output layer given by S'. This stage of the optimization computes the set of basis functions that best represent the input, subject to the sparseness constraint imposed by S.

In the M phase, we compute the optimal G for the current $v(u)$. Taking the derivative of $\mathscr{F}$ with respect to G, setting equal to zero, and solving for G we obtain the update rule

$$G \to \langle uv^T \rangle \left(\frac{\lambda}{\gamma} I + \langle vv^T \rangle \right)^{-1}. \tag{5}$$

This rule yields the global optimum for G given $v(u)$ and so lets us take large steps toward the optimum of $\mathscr{F}$ in the M phase. This in turn leads to much faster convergence than the incremental update used in previous work [9, 10]. If, however, we wish to perform on-line learning in which images are presented one at a time, gradient ascent yields a Hebbian-with-decay update rule.

3 Classification Experiments

We performed several experiments with this model. In all cases the number of outputs from the C2b and C3 layers of the visual system model [15] - and thus the input to the

sparse learning network - was $n = 2000$, and the number of output units was $m = 10$. The matrix G was initialized with uniformly distributed random weights between -0.5 and 0.5. Equation 4 was simulated in MATLAB for sufficient time to reach equilibrium with the additional constraint that all responses v_i be nonnegative (using MATLAB's "NonNegative" `odeset` property) and parameters $\lambda = 10$, $t = 0.05$, and $\sigma^2 = 0.04$. The weight penalty was $\gamma = 100$. In each experiment we used the batch update rule (eq. 5) and terminated the optimization when the average change in the weights g_{ij} was less than 1%. Except for experiment (D), for which fewer images were available, we used 40 random images from each category for training and reserved 40 different images for testing. After training, the recognition model (eq. 4) was run on the novel testing images.

We performed the following four experiments:

(A) Three object categories. The model was trained and tested on images of motorbikes, airplanes, and faces. This is directly comparable to experiment (C) of [17].

(B) Four object categories. We added a fourth category (cars) to the training set from experiment (A). This is similar to experiment (D) of [17], except that we used side- rather than rear-views of cars.

(C) Four object categories. As the images from experiment (B) are relatively easy to classify (a supervised classifier operating on the same inputs can perform this task at near 100% accuracy), we performed the same experiment with four more difficult categories: blimps, elephants, ketches (a type of sailboat), and leopards.

(D) Five individuals. We sorted the face images from the Caltech 256 database into categories consisting of images of the same individual. We then presented images of 5 of these individuals. We presented 10 images of each individual in the training stage, reserving 10 different images of each individual for testing.

4 Results

We ran each experiment 10 times with different random initial conditions for G. All model parameters were identical between the four experiments - no adjustment was required to account for different number or type of input categories between experiments.

4.1 Response Profiles

We here focus on describing the response profiles of the output units from a typical run of experiment (B); results from the other trials and experiments were qualitatively similar. Figure 1 depicts the responses of two of the selective units (from the same session) that emerged in training. For each unit this figure shows 20 of the 40 images that evoked the strongest responses (every other response is omitted for clarity) as well as a histogram of all responses. The ROC curve for each unit treated as a classifier for its preferred category is inset in the histogram, along with the ROC curve for the best principal component for that category for comparison. We see from these figures that category tuning has spontaneously emerged from the learning process.

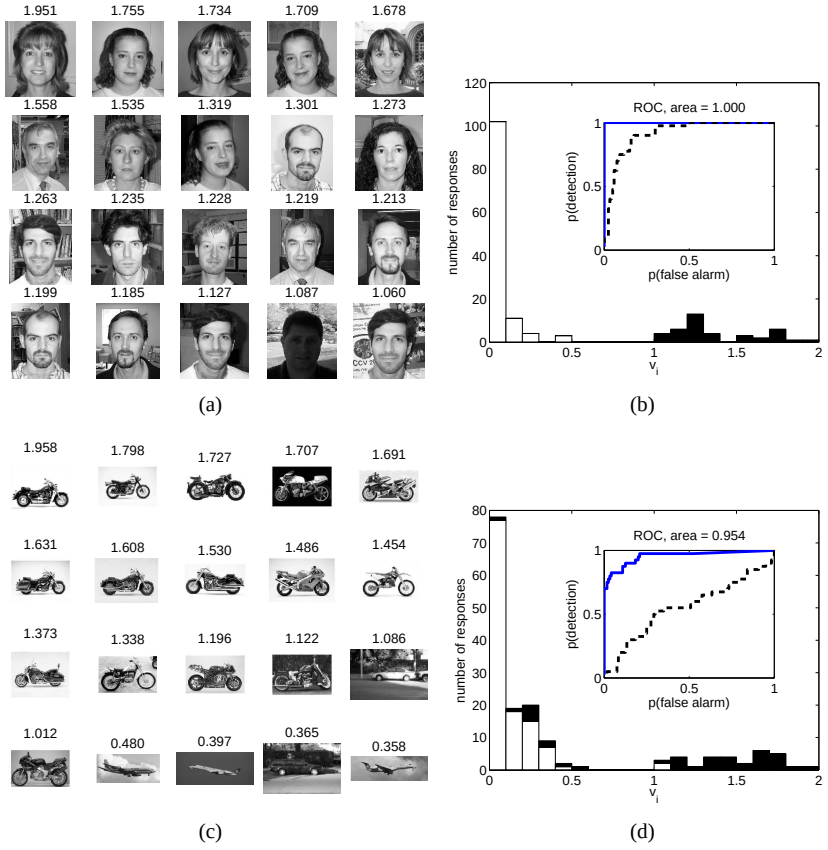


Figure 1: Responses of two selective units after the unsupervised category learning. (a,c): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b,d): response histograms. *x*-axis is the activation level; *y*-axis is the number of test images (160 total) evoking a response at that level. Responses to preferred category in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit, dashed line is ROC curve for best principal component. ROC equal-error accuracies were 100% and 88%.

4.2 Classification Accuracy

Given that we use a purely unsupervised training process, and that our model is free to identify fewer or more categories than are present in the training set, there are several possibilities for evaluating the classification accuracy of this system. We consider three metrics here, two of which are weakly supervised as they require us to decide what category each unit is selective for, and one of which is fully unsupervised:

Metric 1: Single-category classifier. We consider each unit individually as a classifier for its most selective category. The accuracy figure we use is the receiver-operating characteristic (ROC) equal error rate (i.e. p(true positive) = 1-p(false positive)) testing against the other categories. Chance level in this case is 50%. The metric is the average accuracy of our best classifier for each category.

Metric 2: Weakly supervised classifier. We use all selective units together to classify each input image into one of the input categories. To do so, we first manually assign to each unit a category for which it is most selective as before (so multiple units could be assigned the same category). We then classify each image according to which unit responded the most strongly. The accuracy is then the percentage of testing images correctly classified, and the chance level is one over the number of categories.

Metric 3: Unsupervised classifier. In the fully unsupervised setting we rely on the output units to both define the categories and assign images to them. Each image is assigned to a putative category based on which output unit responded the most strongly. We then form a confusion matrix in which element (i, j) is the percentage of images from input category j assigned to output category i and rearrange this matrix to maximize the average of the diagonal elements, thereby picking the output categories that best correspond to the input categories. This average is then the classification accuracy, and chance level is one over the number of output units (in this case 10).

Note that each of these metrics says something different about the behavior of the network, and none of them by itself describes exactly the sparse, invariant selectivity that is our goal. Metric 1 quantifies how selective individual units are for particular categories, but disregards the separation between on- and off- responses. Metric 3 quantifies how precisely the categories discovered by the network correspond to those we defined, but a network that divides one or more categories into subcategories would score poorly here despite qualitatively good performance. Metric 2 alleviates this issue, but could disregard excessive subcategorization. Hence, sparse, invariant representation of the input categories is only captured by good scores according to all three metrics.

The results of each experiment as measured by these metrics averaged over 10 trials are summarized in Table 1. As a baseline for comparison, we also evaluated the performance of Principal Components Analysis (PCA) applied to the same inputs as our sparse coding network against these three metrics. As we had 10 units in the output layer of the sparse coding network, we used the top 10 principal components for this comparison. We also found the best performance we could achieve using a supervised SVM classifier applied to the same inputs, which provides a reasonable upper bound on achievable performance and an objective measure of task difficulty. For metric 1 we report the average accuracy of a binary SVM classifier for each category versus the others, while for metric

Ex	Metric 1				Metric 2			Metric 3			
	SN	PCA	SVM	ch	SN	PCA	ch	SN	PCA	SVM	ch
A	91.7	69.2	98.1	50.0	90.6	55.0	33.3	64.0	37.5	96.7	10.0
B	89.8	71.9	97.4	50.0	82.6	46.9	25.0	66.1	40.6	96.9	10.0
C	77.0	69.2	88.1	50.0	63.8	47.5	25.0	41.4	36.3	81.9	10.0
D	94.8	85.0	98.0	50.0	83.6	62.0	20.0	75.0	70.0	100.0	10.0

Table 1: Classification accuracy computed using different metrics averaged over 10 trials with random initial conditions. In all cases unseen images were used for testing. For each metric we report the classification accuracy (as a percentage) for the sparse network (SN) and for PCA applied to the same inputs, as well as chance level. For metrics 1 and 3 we also provide the accuracy of a supervised SVM classifier applied to the same inputs.

3 we report the accuracy of a multi-way SVM.

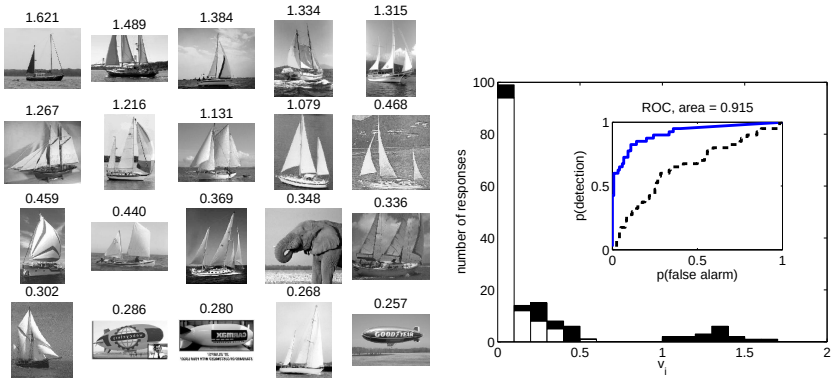
One surprising aspect of these results was the excellent performance in experiment (D), the 5-way face discrimination task which we initially tried as a presumably more difficult test of our methods. While the distinction between different faces is clearly more subtle than the distinction between categories, there is also less within-category variation in the face images than in the images from other categories, so different images of the same individual are likely to be tightly clustered in feature space. From this we see that the within-class homogeneity drives classification accuracy as much as the inter-class separation. Experiment (D) also highlights the importance of the statistics of the input set to the representation learned. In experiments (A) and (B), faces were present often in the inputs, but no particular individual was present often. In this case we obtain a representation for "face," but no individuation within that class. In experiment (D), particular individuals were present often, giving the network enough information to identify multiple individuals and represent them separately.

The seemingly poor results from experiment (C) still occur in the context of units that show very clean selectivity for each category. However, in each case the units responded strongly only to a *subset* of the category in question. Figure 2 gives an example of such a unit which responded selectively to some but not all of the ketch images.

5 Conclusions and Future Work

We here demonstrated a system that is able to group unlabeled images into appropriate categories through unsupervised learning on image features. This model has at its core the notion that underlying the high-dimensional vector of features from the model is a sparse set of causes, and that these causes can be uncovered by optimizing a sparse generative model of the inputs. This model performs quite well on benchmark image classification tasks despite being both entirely unsupervised and motivated primarily by the relevant biology rather than by optimizing machine vision performance. This model has the further important feature that it is not necessary to specify *a priori* the number of categories to search for, except of course to ensure that enough output units are available to represent all the input categories.

Many open questions remain. The simplest is how well this technique scales to larger



(a) (b)

Figure 2: Responses of a ketch unit from experiment (C). (a): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b): response histogram. *x*-axis is the activation level; *y*-axis is the number of test images (160 total) evoking a response at that level. Responses to ketches in black; responses to all other images in white. Inset: ROC curve. Solid line is ROC curve for this unit, dashed line is ROC curve for best principal component. ROC equal error accuracy with respect to all ketches was 85%.

numbers of categories and categories that resemble one another more closely or are more diverse. It remains to be seen whether the feature set used in this investigation is sufficient to discover more (or more similar) categories in this unsupervised setting, or if the underlying visual system model itself is sophisticated enough to scale regardless of the number of features used. Our immediate future work will investigate this scalability.

Acknowledgments. We thank Thomas Serre and Minjoon Kouh of MIT for providing the visual system model used here as well as assistance with its operation, and Richard Murray, Jerry Marsden, and Pietro Perona at Caltech and Bruno Olshausen at Berkeley for valuable feedback. This work was funded by a Fannie and John Hertz Foundation Fellowship (to S.W.), as well as by grants from the ONR, NIMH, NSF, and DARPA.

References

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. "Matching words and pictures". *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] A.J. Bell and T.J. Sejnowski. "An Information-Maximization Approach to Blind Separation and Blind Deconvolution". *Neural Computation*, 7:1129–1159, 1995.

[3] A.J. Bell and T.J. Sejnowski. "The 'Independent Components' of Natural Scenes are Edge Filters". *Vision Research*, 37(23):3327–3338, 1997.

[4] R. Fergus, P. Perona, and A. Zisserman. “Object Class Recognition by Unsupervised Scale-Invariant Learning”. In *Proc. CVPR*, 2003.

[5] G. Griffin, A.D. Holub, and P. Perona. “The Caltech 256”. Technical report, Caltech, 2006.

[6] G.E. Hinton and Z. Ghahramani. “Generative models for discovering sparse distributed representations”. *Phil. Trans. R. Soc. Lond. B*, 352:1177–1190, 1997.

[7] Y. Li, A. Cichocki, and S. Amari. “Analysis of Sparse Representation and Blind Source Separation”. *Neural Computation*, 16:1193–1234, 2004.

[8] J. Mutch and D.G. Lowe. “Multiclass Object Recognition with Sparse, Localized Features”. In *Proc. CVPR*, 2006.

[9] B.A. Olshausen and D.J. Field. “Emergence of simple-cell receptive field properties by learning a sparse code for natural images”. *Nature*, 381:607–609, 1996.

[10] B.A. Olshausen and D.J. Field. “Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?”. *Vision Research*, 37(23):3311–3325, 1997.

[11] R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. “Invariant visual representation by single neurons in the human brain”. *Nature*, 435:1102–1107, 2005.

[12] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. “Efficient Learning of Sparse Representations with an Energy-Based Model”. In *Advances in Neural Information Processing (NIPS 2006)*, 2006.

[13] M. Riesenhuber and T. Poggio. “Hierarchical models of object recognition in cortex”. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[14] E.T. Rolls and M.J. Tovee. “Sparseness of the Neuronal Representation of Stimuli in the Primate Temporal Visual Cortex”. *Journal of Neurophysiology*, 73(2):713–726, 1995.

[15] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. “Object recognition with cortex-like mechanisms”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[16] T. Serre, L. Wolf, and T. Poggio. “Object recognition with features inspired by visual cortex”. In *Proc. CVPR*, 2005.

[17] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. “Discovering Object Categories in Image Collections”. Technical Report MIT-CSAIL-TR-2005-012, MIT CSAIL, 2005.

[18] S. Waydo, A. Kraskov, R. Quian Quiroga, I. Fried, and C. Koch. “Sparse Representation in the Human Medial Temporal Lobe”. *Journal of Neuroscience*, 26(40):10232–10234, 2006.

[19] M. Weber, M. Welling, and P. Perona. “Towards Automatic Discovery of Object Categories”. In *Proc. CVPR*, 2000.