Decoupling sparse coding of SIFT descriptors for large-scale visual recognition

Zhengping Ji[a,b], James Theiler[c], Rick Chartrand[a], Garrett Kenyon[d] and Steven P. Brumby[c]

[a] T-5, Theoretical Division,
[b] Center for Nonlinear Studies,
[c] ISR-3, Intelligence and Space Research Division,
[d] P-21, Physics Division,
Los Alamos National Laboratory, Los Alamos, NM 87545

ABSTRACT

In recent years, sparse coding has drawn considerable research attention in developing feature representations for visual recognition problems. In this paper, we devise sparse coding algorithms to learn a dictionary of basis functions from Scale-Invariant Feature Transform (SIFT) descriptors extracted from images. The learned dictionary is used to code SIFT-based inputs for the feature representation that is further pooled via spatial pyramid matching kernels and fed into a Support Vector Machine (SVM) for object classification on the large-scale ImageNet dataset. We investigate the advantage of SIFT-based sparse coding approach by combining different dictionary learning and sparse representation algorithms. Our results also include favorable performance on different subsets of the ImageNet database.

Keywords: Visual recognition, sparse coding, SIFT, ImageNet

1. INTRODUCTION

The aim of visual recognition is to map pixel inputs to semantic meanings. This effort has become a major focus in computer vision research, and advancement toward this goal largely depends on learning effective feature representations from various scales of pixels. Recently, sparse coding has become a popular approach for adaptively learning such feature representations. Given a set of input signals $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N \in \mathbb{R}^{n\times 1}\}$, sparse coding seeks to reconstruct each input signal with a linear combination of over-complete bases $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_m] \in \mathbb{R}^{n\times m}$ using a sparse coefficient vector $\mathbf{a} \in \mathbb{R}^{m\times 1}$.

$$\min_{\mathbf{D},\mathbf{a}_i} \frac{1}{N} \sum_{i=1}^{N} \left\{ \frac{1}{2} ||\mathbf{x}_i - \mathbf{D}\mathbf{a}_i||_2^2 + \lambda\Omega(\mathbf{a}_i) \right\} \tag{1}$$

Eq. 1 describes the objective function for sparse coding where $\Omega(\mathbf{a}_t)$ is a function to enforce vector sparsity, and λ is a control parameter. Most sparse coding algorithms break the computation into two steps: (1) Given $\mathbf{D}$, learn the sparse vector $\mathbf{a}$ as a feature representation; (2) Given $\mathbf{a}$, learn a set of basis functions $\mathbf{D}$, also called “weights” or a “dictionary”.

A considerable amount of research assumes a known dictionary $\mathbf{D}$ and focuses on learning sparse feature representation by defining sparsity functions such as $||\mathbf{a}||_0$ or $||\mathbf{a}||_1$. The ℓ_0 norm counts the number of non-zero coefficients in a vector, and is a natural choice to encourage sparsity in vector $\mathbf{a}$. But solving Eq. 1 with the ℓ_0 norm is often intractable, so existing studies often seek an approximate solution using a greedy algorithm (*e.g.*, Matching Pursuit (MP),[1] Orthogonal Matching Pursuit (OMP)[2]), or else resort to a convex relaxation under certain assumptions. The most common convex formulation is the ℓ_1-decomposition problem, leading to solutions such as Basis Pursuit (BP),[3] FOCUSS[4] and the Lasso.[5]

Rather than using pre-defined dictionaries, another line of sparse coding algorithms aims to learn a dictionary of basis functions, which leads to an alternating minimization for both $\mathbf{D}$ and sparse vector $\mathbf{a}$. Well-known examples include the pioneering work of Olshausen and Field[6] to model neuronal responses in the V1 area of the brain, the K-SVD of Elad and Aharon,[7] the Online Sparse Coding of Mairal *et al.*,[8] and others.[9–13] The dictionary learning algorithms have two

Send correspondence to Zhengping Ji: E-mail: jizhengp@gmail.com

paired components: one for updating the dictionary elements and the other for learning the sparse representation. Some of the greedy searching and convex optimization techniques described above for sparse representation is applied here and integrated in the dictionary learning scheme.

In this paper, we describe sparse coding algorithms to learn a dictionary from Scale-invariant Feature Transforms (SIFT) descriptors of images (rather than from raw pixels), and then develop sparse feature representations for SIFT-based inputs given the learned dictionary. The sparse representation is further pooled via spatial pyramid matching kernels and fed into a support vector machine (SVM) for object classification on the large-scale ImageNet dataset. SIFT is a state-of-the-art algorithm in computer vision to detect and describe local areas of images via histograms of orientations.[14] Feature representation via sparse coding of SIFT descriptors has shown favorable performances[15–18] in various visual recognition tasks such as MNIST,[19] NORB,[20] CIFAR-10,[21] and Caltech-101,[22] but has not demonstrated in a large-scale visual recognition task like ImageNet.[23]

As the sparse representation of a SIFT descriptor only represents a local area in an image, we need a mechanism to combine the local representations for the whole image. A popular approach is the called Bags-of-Words (BoW) model,[24] which quantizes the feature representations into "visual words", and then computes a histogram representation of the whole image for semantic classification. However, the spatial order of local representations is discarded in the BoW model, which limits the descriptive power of image representation. A more advanced alternative is called spatial pyramid matching (SPM),[17,25] which extends the BoW model in multiple scales but maintains spatial order across the scales. SPM has had remarkable success on the sparse representation of local descriptors in a range of image classification tasks.[15,17,18]

We investigate the SIFT-based sparse coding approach by comparing different pairs of algorithms for dictionary learning and sparse representation. As mentioned above, dictionary learning algorithms are generally paired specific sparse representation approaches, but in our experiments, we mix and match the different dictionary learning algorithms with different sparse representation algorithms. From the comparison, we find that:

(1) From the perspective of learning sparse representations, ℓ_1-regularized optimization algorithms on average outperform greedy approximation algorithms using ℓ_0-based sparsity.

(2) Performance depends predominantly on choice of sparse representation algorithm; choice of dictionary is less important.

(3) Even using a dictionary with random SIFT patches (without training), the performance is comparable to those using the trained dictionaries.

In addition, our results show favorable performance on different subsets of ImageNet database, which contains 15M images in 22K object categories that were designed to exhibit a direct mapping to WordNet language concepts.

2. LEARNING FRAMEWORK

In this section, we will describe the learning framework as a pipeline process, with each computational component at each subsection.

2.1 SIFT Descriptor

Given an image, SIFT finds the keypoints with respect to local minimum or maximum given the difference of adjacent Gaussian smoothing operations, where each keypoint is associated with the information regarding its location, local scale and orientation. Based on the local region around the keypoint, a local image descriptor is computed as 16 histograms of 8 gradient orientations.

In this paper, we use a simpler and faster version of SIFT algorithm, called *dense SIFT*, which assumes that the location, scale and orientation of each keypoint is predefined rather than extracted from a scale-space extrema. In our implementation, 16×16 pixel patches were densely sampled from each image on a grid with step size 8 pixels, with the center of each patch considered the keypoint. This yields a representation of the image as a set of 128-dimensional (8 orientations$\times$ 16 histograms) vectors, with one descriptor representing each patch of the grid. Mathematically, each image $\mathbf{X}_i$ is represented as a matrix containing each SIFT descriptor as a column vector, i.e., $\mathbf{Y}_i = [\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)} ..., \mathbf{y}_i^{(\mathrm{p})}] \in \mathbb{R}^{128 \times \mathrm{p}}$, where is p is the number of SIFT vectors.

2.2 Dictionary Learning

Given a training set containing a number of images $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N\}$, we have a corresponding training set with SIFT description, i.e., $\mathcal{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_N\}$. Among all the SIFT descriptors for all the images in the training set, we randomly choose K descriptors, i.e., $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}..., \mathbf{y}^{(K)}\}$ $(\mathbf{y}^{(k)} \in \mathbb{R}^{128\times 1})$ for the learning of a dictionary $\mathbf{D}$ via sparse coding, such that

$$\min_{\mathbf{D},\mathbf{a}^{(k)}} \frac{1}{K} \sum_{k=1}^{K} \left\{ \frac{1}{2} ||\mathbf{y}^{(k)} - \mathbf{D}\mathbf{a}^{(k)}||_2^2 + \lambda\Omega(\mathbf{a}^{(k)}) \right\} \tag{2}$$

where the dictionary contains 1024 elements, each of dimension 128 (same as a SIFT input), such that $\mathbf{D} \in \mathbb{R}^{128\times 1024}$.

We applied three different dictionary learning algorithms in this paper:

1. **K-SVD**: K-SVD is a simple but efficient dictionary learning algorithm developed by Aharon *et al.*[7] It generalizes the idea of K-Means and solves Eq. 2 in an alternating manner. First, the sparse vector $\mathbf{a}$ is obtained using the aforementioned Orthogonal Matching Pursuit to approximate the solution to the non-convex ℓ_0-regularized sparse problem. Second, the dictionary is learned via a batch of input samples, where only one column of $\mathbf{D}$ is updated at a time using the singular value decomposition (SVD).

2. **Lagrange dual**: This is an efficient dictionary learning algorithm proposed by Lee *et al.*[11] that uses a sign-search algorithm to solve an ℓ_1-regularized least squares problem with respect to the sparse vector $\mathbf{a}$ in Eq. 2. Then a Lagrange dual method is used to solve the ℓ_2-constrained least squares problem with respect to a dictionary $\mathbf{D}$. Both problems above are known to be convex.

3. **SPAMS**: SPAMS is a SPArse Modeling Software package containing an optimization toolbox for various sparse estimation problems. We used its dictionary learning solver based on the paper published by Mairal *et al.*,[8] where a Cholesky-based implementation of the LARS-Lasso algorithm[26] is utilized to solve the ℓ_1-regularized sparse coding problem with respect to a sparse vector $\mathbf{a}$ and a new online optimization algorithm based on stochastic approximations is developed to learn a dictionary $\mathbf{D}$.

The detailed algorithm of each method is available at the corresponding paper cited above and beyond the scope of this paper.

2.3 Sparse Representation

We applied the trained dictionary to code every SIFT patch in every image and generate the sparse representation via an optimization step as below,

$$\forall \mathbf{y} \in \mathcal{Y}, \quad \min_{\mathbf{a}} \frac{1}{2} ||\mathbf{y} - \mathbf{D}\mathbf{a}||_2^2 + \lambda\Omega(\mathbf{a}) \tag{3}$$

Note that only the sparse vector $\mathbf{a}$ is learned here, with a fixed $\mathbf{D}$. Each sparse vector $\mathbf{a} \in \mathbb{R}^{1024\times 1}$ represents one SIFT path in one image.

We applied three different learning algorithms to compute sparse representation in Eq. 3: 1. Orthogonal Matching Pursuit;[2] 2. Sign-search optimization;[11] and 3. a variant of the LARS-Lasso algorithm.[26] In fact, each learning algorithm here for sparse representation is used in one of the dictionary learning algorithms above, but it is found that the *natural* choice of sparse representation algorithm that matches the dictionary learning (*e.g.*, Orthogonal Matching Pursuit with respect to K-SVD) is not necessary to provide feature representation for favorable classification performance. In other words, mismatching the learning algorithms for sparse representation in Eq. 2 and in Eq. 3 may surprisingly deliver more favorable results.

Given a SIFT description of an image $\mathbf{Y}_i = [\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)}..., \mathbf{y}_i^{(\mathrm{p})}] \in \mathbb{R}^{128\times \mathrm{p}}$, we now have a sparse feature representation $\mathbf{A}_i = [\mathbf{a}_i^{(1)}, \mathbf{a}_i^{(2)}..., \mathbf{a}_i^{(\mathrm{p})}] \in \mathbb{R}^{1024\times \mathrm{p}}$ via Eqs. 2 and 3.

2.4 Spatial Pyramid Matching (SPM)

As size of images varies in the training set, the number of SIFT patches varies as well, meaning that we have a different number of sparse vectors in each feature representation $\mathbf{A}_i$. Thus, further processing is needed, because we need representations with identical dimensions in order to feed them into a classification model. Given a set of sparse vectors $\{\mathbf{a}_i^{(1)}, \mathbf{a}_i^{(2)} ..., \mathbf{a}_i^{(\mathrm{p})}\}$ obtained via Eq. 3 for each image, a popular choice is to quantize the sparse vectors and then compute a histogram representation. This procedure is called a Bag-of-Words (BoW) model, where each image is represented by an unordered set of local descriptors.

In a more sophisticated SPM approach, we partition an image into 4×4, 2×2, and 1×1 segments respectively, and max-poole the sparse vectors within each of the 21 segments, so that in each segment region the d-th element of pooled vector $\mathbf{s}$ is as follows:

$$s_d = \max(|a_d^{(1)}|, |a_d^{(2)}|, ..., |a_d^{(\mathrm{q})}|), \tag{4}$$

where a_d is the d-th element in a sparse vector $\mathbf{a}$, and $\mathrm{q} \leq \mathrm{p}$ is the number of sparse vectors in that region.

The pooled vectors from various locations and scales are then concatenated to form a spatial pyramid representation of the image, which has 21504 (=1024 vector components × 21 scales) dimensions and is able to be fed into a classification model for decision making.

2.5 Classification Models

We used the libSVM implementation[27] of a linear support vector machines (SVM) to classify SPM representations. The training data is the set $\{(\mathbf{X}_i, c_i)\},\ i = 1, 2, ...N$, where $\mathbf{X}_i$ is an image input and $c_i \in \mathcal{C} = \{1, 2, ..., L\}$ is the corresponding class label of this image. Through the computational steps above, each image $\mathbf{X}_i$ is represented as an SPM representation $\mathbf{s}_i$. We used a one-against-all strategy to train L binary linear SVMs, each solving the convex optimization problem as follows

$$\min_{\mathbf{w},\xi,b} \left\{ \frac{1}{2}\|\mathbf{w}_l\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \tag{5}$$

$$\text{s.t. } f(c_i)(\mathbf{w}_l \cdot \mathbf{s}_i - b_l) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

where $f(c_i)$= 1 if $c_i = l$, otherwise $f(c_i) = -1$ ($l = 1, 2, ..., L$).

In addition to determining the predicted class label for a testing sample, which is given by

$$\arg\max_{l \in \mathcal{C}} (\mathbf{w}_l \cdot \mathbf{s} - b_l) \tag{6}$$

we further employed the libSVM toolbox to calibrate its output to correspond to probability estimates for class predictions. Thus, the output of a testing sample is a distribution of likelihoods associated with each class.[28]

3. EXPERIMENTAL RESULTS

In the experiments, we evaluated the learning framework on the ImageNet dataset.[23] ImageNet is a publicly available image database containing 15M labeled images belonging to 22K object categories, which are organized according to the WordNet hierarchy of meaningful concepts. About 1000 images are included in each category/label, and some of the images are further annotated for object detection purposes. A subset of ImageNet with 1000 categories (most of which are from leaf nodes in the semantic hierarchy) are extracted from the ImageNet to establish an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) since 2010. In the ILSVRC dataset, there are $\sim$1.2M training images, 50K validation images, and 150K testing images.

We selected 14, 64 and 196 categories and their associated images from the ILSVRC-2012 dataset to conduct visual object recognition using the described learning system. We preprocessed all the images to gray scale. If a shorter size of the image is larger than 256 pixels, we rescaled the image so that the shorter size is 256.

3.1 ImageNet-14

This is a subset of ImageNet containing 14 object categories, from which we randomly selected 500 images for training and 200 for testing. We utilized the learning framework to learn SIFT-based sparse representations (given a trained dictionary) and further generated SPM representations for image classification using linear SVMs. This small-scale problem is used to tune parameters (*e.g.*, λ in Eq. 2 and 3, C in Eq. 5) and evaluate the choice of learning algorithms (i.e., for Eq. 2 and Eq. 3 respectively) in the described framework.

Table 1 shows how the classification accuracy varies given several choices of parameters, as well as different pairs of learning algorithms for the dictionary and the sparse representation. Note that we did not explore the parameter space with all possible values, but referred to empirical studies about favorable settings of sparse coding parameters in visual recognition tasks.[16] As discussed in Secs. 2.2 and 2.3, three dictionary learning algorithms (K-SVD, Lagrange dual (LD) and SPAMS) and three sparse representation algorithms (Sign-search, LARS-Lasso and OMP) are included in this experiment, which represent state-of-the-art approaches at the current time.

Two numbers are reported for each specific setting (*i.e.*, in each cell of Table 1): The left one is for the top-1 accuracy rate – the fraction of testing images that is correctly predicted, and the right one is for top-5 accuracy rate – the fraction of testing images for which the correct label is among the five labels considered most probable by the model. The top-5 rate was adopted as a useful standard for the ImageNet dataset, since each image typically contains multiple objects.

Table 1. Classification accuracy for ImageNet-14.

	Sign Search ($\lambda = 0.15$)	LARS ($\lambda = 0.15$)	LARS ($\lambda = 0.3$)	OMP (L=10)	OMP (L=100)
K-SVD	70.21%, 94.93%	69.11%, 94.18%	69.21%, 93.86%	65.36%, 93.14%	54.34%, 89.01%
LD	69.93%, 94.86%	69.00% 94.75%	67.04% 94.11%	65.89% 92.64%	56.11%, 88.62%
SPAMS	69.96%, 94.50%	70.07%, 94.57%	69.50%, 94.64%	65.57%, 93.11%	55.43%, 89.18%
Random	70.36%, 94.32%	68.50%,95.18%	67.82%, 94.62%	63.32%, 92.04%	53.12% , 86.44%

In Table 1, we observe that: (1) When learning sparse representations, ℓ_1-regularized optimization algorithms (the first three columns) on average perform better than greedy algorithms using ℓ_0-based sparsity (the last two columns). (2) Regardless of dictionary choice, learning algorithms for sparse representation mainly result in performance variance, as indicated in Fig. 1. (3) Even using a dictionary with random SIFT patches (without training), the performance is comparable to those using the trained dictionaries (see the last row in Table 1).

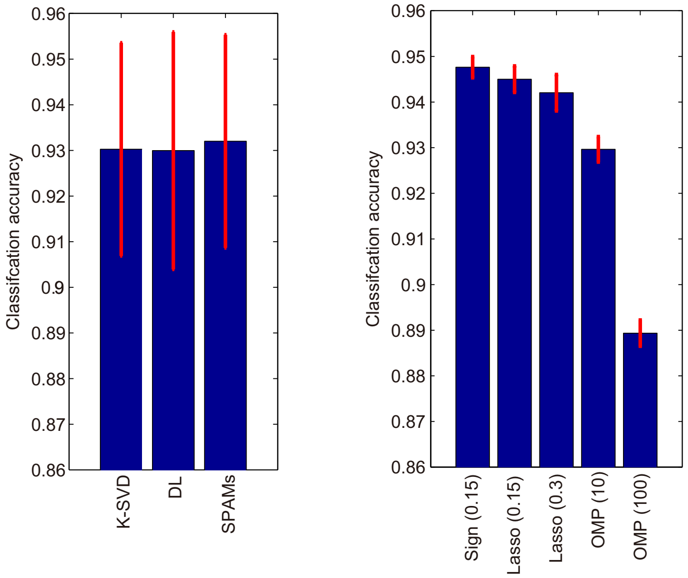
As discussed in Sec. 2.5, we applied probabilistic estimates of SVM prediction to the output classes. In Fig. 2, we plotted some examples of testing images, along with their predicted probabilistic outputs for the top 5 classes. The left column illustrates some easy cases (with high confidence regarding a particular class that is correctly predicted) and the right column illustrates some tough cases of the same class, showing very different distribution patterns (far more uncertainty, and many incorrect choices).

3.2 ImageNet-64 and ImageNet-196

The results in Table 1 indicated that a *natural* choice of sparse representation algorithm that matches the one included in dictionary learning (*e.g.*, Orthogonal Matching Pursuit with respect to K-SVD) may not be optimal to provide favorable feature representation for classification performance. Based on the observation of the previous experiment, we selected the K-SVD algorithm for dictionary learning and Sign-search algorithm for the sparse representation of each input, and conducted visual recognition with more classes. The regularization parameter of λ and C is set to 0.15 and 10 respectively.

Fig. 3 shows the classification performance with top-5 accuracy rate for ImageNet-14, 64, and 196, using K-SVD and Sign-search. All the parameters are set the same as suggested above. The classification performance here is, although inferior to that reported by the winner of ILSVRC-2012,[29] much better than the baseline performance initially provided by the inventor of the datset.[30] On-going studies that focus on the fusion of feature representation (especially integrating color information) and hierarchical decision models (dealing with probabilistic class output for further processing) show great promise in substantially improving classification performance.

The top-5 accuracy for each class varies in ImageNet-196. We illustrated several examples for the object classes delivering worst performance (lower than 40%, see Fig. 4) and best performance (higher than 90%, see Fig. 5). These



(a) (b)

Figure 1. Mean and variance of top-5 accuracy rates with respect to (a) the dictionary learning algorithms across different sparse representations. (b) the sparse representation algorithms (given various parameters) across different dictionaries.

figures show that examples within each class are highly variable (*e.g.*, different object forms, backgrounds, poses, colors and light conditions, *etc.*), which illustrates why identifying the correct label is so challenging.

4. CONCLUSION

In this paper, we performed experiments using the SIFT-based sparse coding and Spatial Pyramid Matching framework for the visual recognition of ImageNet subsets. We investigated the pairs of algorithms for dictionary learning and development of sparse representation. The results show that the algorithms for sparse representation mainly determined the classification accuracy, and that the choice of dictionary was less important. Matching the sparse representation algorithm with the one included in each dictionary learning does not guarantee better performance. Instead, the choice of sparsity itself plays a key role, where ℓ_1-regularized sparse optimization in general was found to be superior to greedy approximation in the ℓ_0-based sparse formulation. In fact, even using an unlearned dictionary with imprinted random patches, once we choose suitable algorithms for sparse representation, we still observed performance comparable to those with expensively trained dictionaries. Future improvement can be made by integrating sparse representations containing different information channels, especially colors, which are discarded in the current study. As our learning framework provides probabilistic likelihood regarding each class, a hierarchical decision model that assesses the distribution of probabilistic outputs and target the difficult cases for further fine-tuned classification is promising to boost the performance as well.

ACKNOWLEDGMENTS

This work is supported by the Laboratory Directed Research and Development (LDRD) program at Los Alamos National Laboratory.

REFERENCES

[1] Mallat, S. and Zhang, Z., "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing* **41**, 3397–3415 (1993).

[2] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S., "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition.," in [*The 27th Asilomar Conf. on Signals, Systems, and Computers*], (1993).

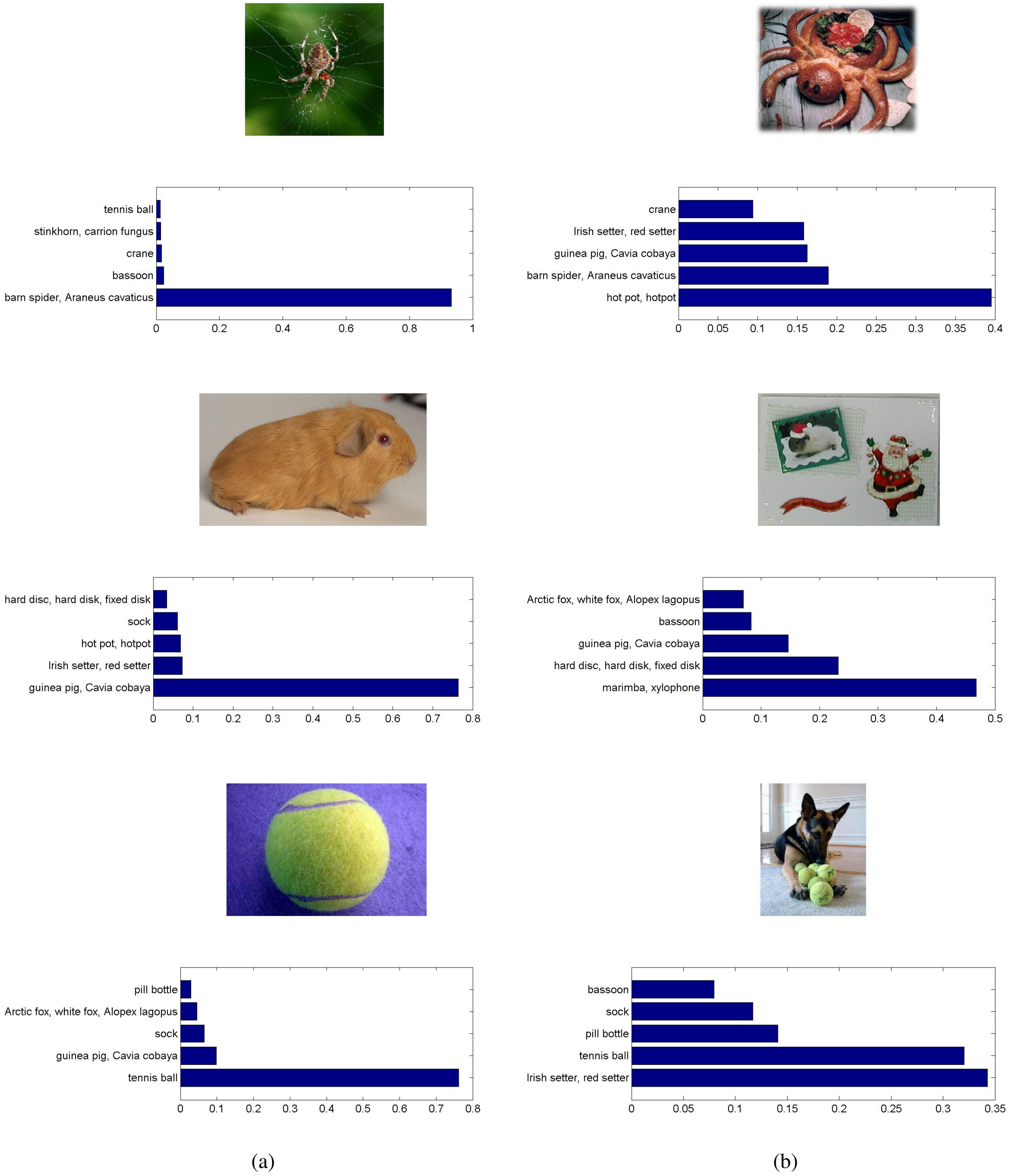


Figure 2. Examples of testing images and predicted probabilistic outputs for top-5 classes. Each row denotes one class, where the left example shows an easy case and the right one shows a hard case. Each probabilistic output is normalized with respect to the top-5 classes.

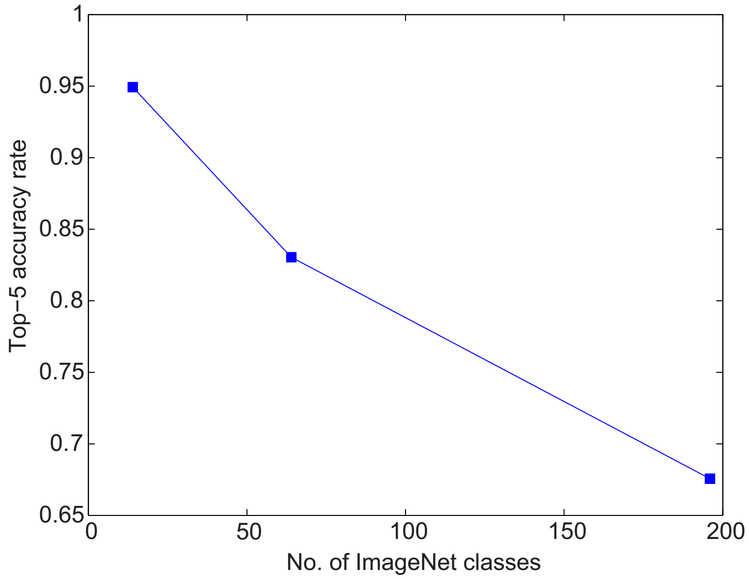


Figure 3. Top-5 accuracy rate for ImageNet-14, 64, and 196.

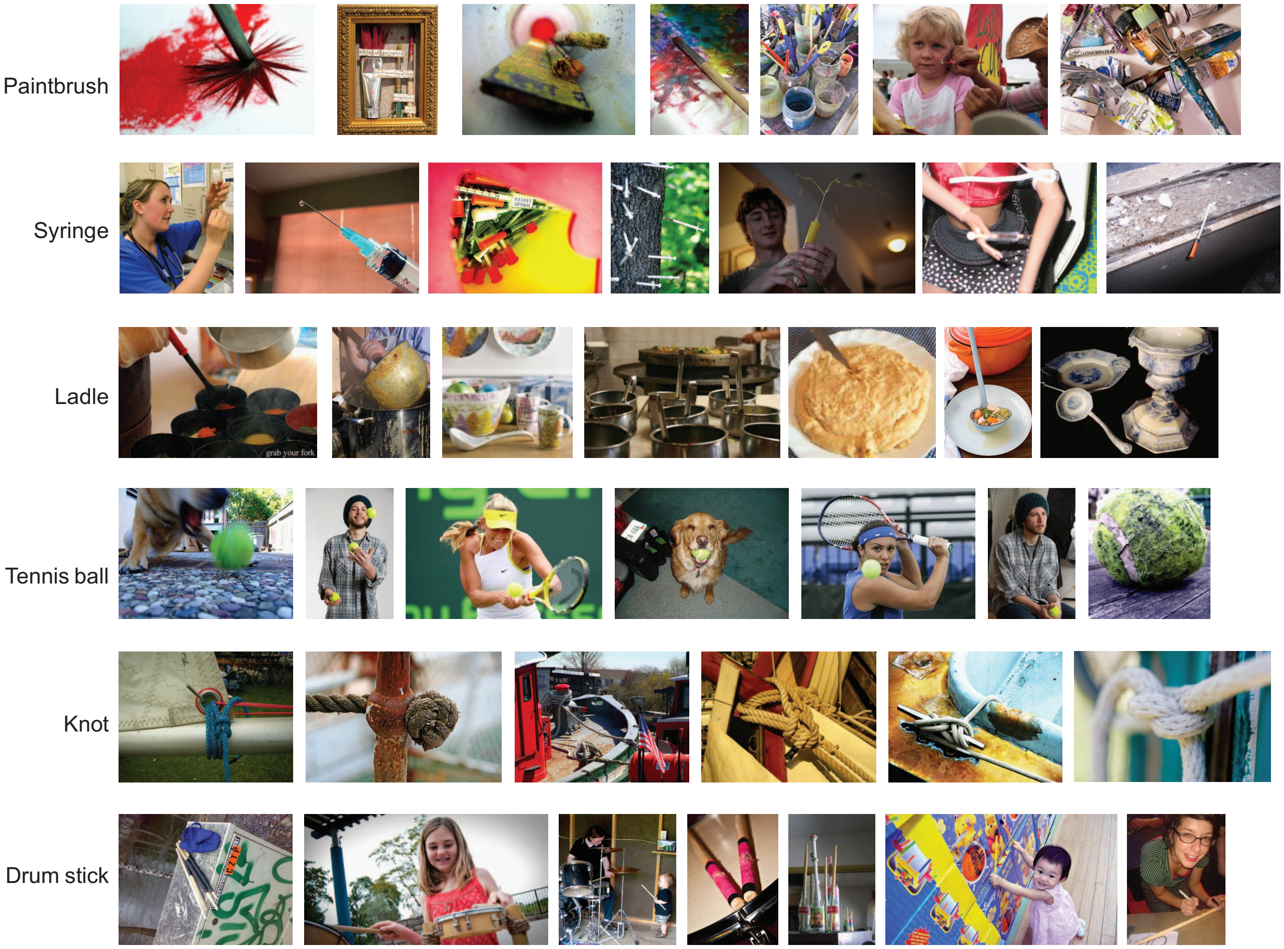


Figure 4. Examples of object classes that deliver the lowest top-5 accuracy ($< 40\%$) in ImageNet-196.

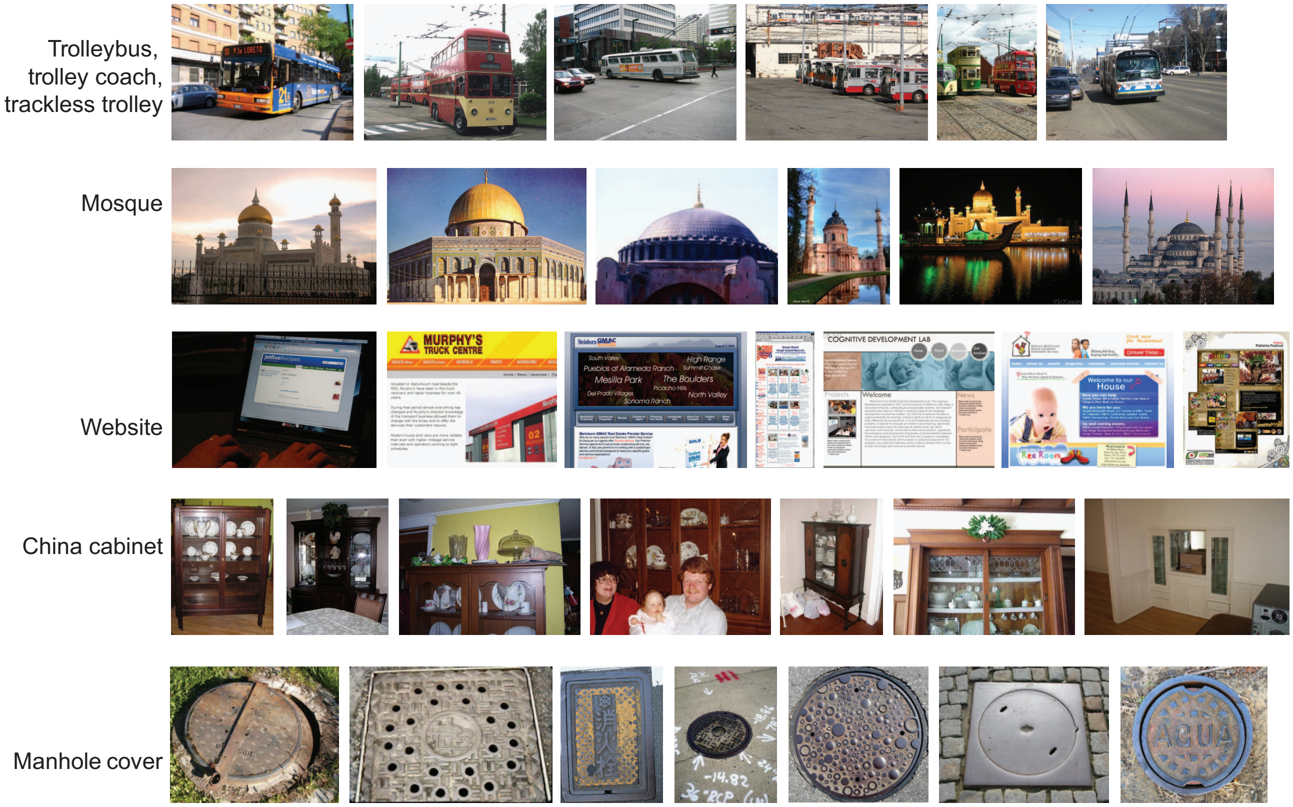


Figure 5. Examples of object classes that deliver the highest top-5 accuracy ($> 90\%$) in ImageNet-196.

[3] Chen, S., Donoho, D., , and Saunders, M., “Automatic decomposition by basis pursuit,” *SIAM Journal of Scientific Computation* **1**(3), 33–61 (1998).

[4] Gorodnitsky, I. F. and Rao, B. D., “Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm,” *IEEE Transactions on Signal Processing* **45**(3), 600–616 (1997).

[5] Tibshirani, R., “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society, Series B* **58**(1), 267–288 (1996).

[6] Olshaushen, B. A. and Field, D. J., “Sparse coding with an overcomplete basis set: A strategy used by V1?,” *Vision Research* **37**(23), 3311–3325 (1997).

[7] Aharon, M., Elad, M., and Bruckstein, A., “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. on Signal Processing* **54**(11), 4311–4322 (2006).

[8] Mairal, J., Bach, F., Ponce, J., and Sapiro, G., “Online learning for matrix factorization and sparse coding,” *Journal of Machine Learning* **11**, 19–60 (2010).

[9] Engan, K., Aase, S. O., and Husoy, J. H., “Method of optimal directions for frame design,” in [*IEEE International Confrence on Acoustics, Speech, and Signal Processing*], (1999).

[10] Lewicki, M. S. and Sejnowski, T. J., “Learning overcomplete representations,” *Neural Computation* **12**(2), 337–365 (2000).

[11] Lee, H., Battle, A., Raina, R., and Ng, A. Y., “Efficient sparse coding algorithms,” in [*Advances in Neural Information Processing Systems*], 1137–1144 (2007).

[12] Ji, Z., Huang, W., Kenyon, G., and Bettencourt, L. M. A., “Hierarchical discriminative sparse coding via bidirectional connections,” in [*Proc. International Joint Conference on Neural Networks*], (2011).

[13] Ji, Z., Huang, W., and Brumby, S., “Learning sparse representation via a nonlinear shrinkage encoder and a linear sparse decoder,” in [*IEEE International Joint Conference on Neural Networks*], (2012).

[14] Lowe, D. G., “Distinctive image features from scale invariant keypoints,” *International Journal of Computer Vision* **60**(2), 91–110 (2004).

[15] Boureau, Y., Bach, F., LeCun, Y., and Ponce, J., "Learning mid-level features for recognition," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2010).

[16] Coates, A. and Ng, A., "The importance of encoding versus training with sparse coding and vector quantization," in [*Proc. IEEE International Conference on Machine Learning*], (2011).

[17] Yang, J., Yu, K., Gong, Y., and Huang, T., "Linear spatial pyramid matching using sparse coding for image classification," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2009).

[18] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Guo, Y., "Locality-constrained linear coding for image classification," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2010).

[19] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of IEEE* **86**(11), 2278–2324 (1998).

[20] LeCun, Y., Huang, F., and Bottou, L., "Learning methods for generic object recognition with invariance to pose and lighting," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2004).

[21] Krizhevsky, A., "Learning multiple layers of features from tiny images," in [*Master Thesis, Dept. of Computer Science*], (2009).

[22] Fei-Fei, L., Fergus, R., and Perona, P., "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision*], (2004).

[23] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "ImageNet: A large-scale hierarchical image database," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2009).

[24] Fei-Fei, L. and Perona, P., "A Bayesian hierarchical model for learning natural scene categories," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2005).

[25] Lazebnik, S., Schmid, C., and Ponce, J., "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in [*Proc. IEEE International Conference on Computer Vision and Pattern Recognition*], (2006).

[26] Osborne, M., Presnell, B., and Turlach, B., "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis* **20**, 389–403 (2000).

[27] Chang, C.-C. and Lin, C.-J., "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology* **2**, 27:1–27:27 (2011).

[28] fan Wu, T., Lin, C.-J., and Weng, R. C., "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research* **5**, 975–1005 (2003).

[29] Krizhevsky, A., Sutskever, I., , and Hinton, G., "ImageNet classification with deep convolutional neural networks," in [*Advances in Neural Information Processing Systems*], (2012).

[30] Fei-Fei, L., "ImageNet: crowdsourcing, benchmarking & other cool things," in [*CMU VASC Seminar*], (March, 2010).