Spatially Localized Visual Dictionary Learning

Anonymous Submission

ABSTRACT

This paper addresses the challenge of devising new *representation learning algorithms* that overcome the lack of *interpretability* of classical visual models. Therefore, it introduces a new recursive visual patch selection technique built on top of a *Shared Nearest Neighbors* embedding method. The main contribution of the paper is to drastically reduce the high-dimensionality of such over-complete representation thanks to a recursive feature elimination method. We show that the number of *spatial atoms* of the representation can be reduced by up to two orders of magnitude without much degrading the encoded information. The resulting representations are shown to provide competitive image classification performance with the state-of-the-art while enabling to learn highly interpretable visual models.

1. INTRODUCTION

Over recent years, specialized image classification challenges such as plants, vehicles, buildings or logos recognition have received a lot of attention. Many supervised classification algorithms have shown very good performance on such datasets reducing more and more the gap between humans and machines. A very interesting and promising challenge would be to transfer the knowledge of these learning algorithms to humans so that we can gain insights into which part of the data is used by the learning algorithm to discriminate between different visual concepts. This would allow humans such as domain experts to (i) understand which visual patterns are discriminant or ambiguous from a concept to an other, (ii) detect some errors or limitations in the machine learning process, and (iii) improve their knowledge of the objects of interest by taking advantage of the machine to discover fine relevant details. However, the visual representations used in classical visual models are too abstract to fulfill these interpretability objectives. The learned *atoms* (*e.g.* latent variables in probabilistic models or *visual words* in codebook learning methods) do actually not have a uniquely defined and easily interpretable visual appearance. They can typically not be directly *visualizable* or mapped without any ambiguity onto localized visual contents in the training set. The challenge addressed in this paper is to devise new image representation learning algorithms that overcome this lack of *interpretability*. Therefore, we propose a supervised method for learning a compact vocabulary of *discriminant* and *spatially localized* visual patches to be used as atoms of a highly interpretable image representation. To do so, we introduce a recursive visual patch selection technique built on top of a recently introduced embedding scheme called Shared Nearest Neighbor match kernel [10]. The interesting property of such embedding is that it explicitly maps the visual content of a given image onto a potentially huge set of visual patches. So that the image can be represented through a very high-dimensional feature vector encoding its similarity to each visual patch in the training set. In this paper, we propose to drastically reduce the dimensionality of such brute-force and over-complete representation thanks to a recursive feature elimination method. We show that the number of *spatial atoms* of the representation can be reduced by up to two orders of magnitude without much degrading the encoded information. The resulting representations are shown to provide competitive image classification performance with the state-of-the-art while enabling to learn highly interpretable visual models.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICMR'16, June -, 2016, New York, US.
Copyright 2016 ACM 978-1-4503-3274-3/15/06 ...$15.00.
DOI: .

2. RELATED WORKS

Codebook learning methods: One of the most popular *representation learning* algorithm is the so-called Bags-Of-Visual-Words paradigm (BoVW) [7]. It mainly consists in learning a visual vocabulary from a set of local feature vectors through an unsupervised learning algorithm (*e.g* KMeans or GMM). More effective *codebook learning methods* were proposed in the following years using aggregated-based methods such as the Fishers Vectors [13] and VLAD schemes [14] that do not only encode the number of occurrences of each visual word but also encode additional information about the distribution of the descriptors. Such very high-dimensional representations are effective for classification tasks but they are not adapted to our interpretability objectives. Sparse Coding and in particular the supervised dictionary learning method of Mairal *et al.* [9] are a way of learning much more compact visual representations. However, as discussed in the introduction, the atoms of the learned vocabularies do still not correspond to easily interpretative visual patterns. They do not have a uniquely defined visual representation and might embed different visual patterns in the same atom.

Spatially localized representations: In [5], the authors introduce an unsupervised scheme to learn a visual dictionary by randomly picking spatially localized local features selection and ranking them with an information gain criterion combined with a saliency score. Krapac et al proposed in [6] a alternative approach rather based on a prototype selection approach: local descriptors are all kept in their original form (without quantization) and a distance-adaptive prototype is trained for each of them in a supervised way. Our method is different in two main points: first, the elementary atoms of our representation are not individual local features but sets of spatially neighboring local features. This allows embedding much more information in a single *spatial atom*. Secondly, we use a *recursive* feature elimination approach that allows selecting a much better set of spatial atoms than one-shot approaches.

3. PROPOSED METHOD (RVPS)

We define a *spatially localized vocabulary* as a set $\mathcal{Z}$ of *spatial atoms* Z_j, $j \in [1, N]$, each uniquely corresponding to a spatial region R_j of an image in the training set. We define each *spatial atom* Z_j as being itself composed of a set of spatially localized d-dimensional feature vectors $\mathbf{z_j^i}$, $i \in [1, |Z_j|]$, extracted from R_j and representing its local visual content. Our aim is to automatically learn a *spatially localized vocabulary* $\mathcal{Z}$ that is as much compact as possible while still containing the most explanatory visual patterns of the labeled classes in the training set. We therefore introduce a new Recursive Visual Patch Selection algorithm (RVPS) that is summarized in **Algorithm 1**. Its principle is to progressively compress the *spatially localized vocabulary* $\mathcal{Z}$ by recursively eliminating the less discriminant atoms. Each recursion includes 3 main steps: (i) the computation of the SNN representations [10] of the images in the training set $\mathcal{X}$ (based on the current *spatially localized vocabulary* $\mathcal{Z}^{(t)}$), (ii) the learning of a multi-class support vector machines on top of the computed SNN representations and (iii), the elimination of the less discriminant spatial atoms $Z_j^{(t)}$ in $\mathcal{Z}^{(t)}$. These 3 steps are repeated T times. The main parameter of the algorithm is the filtering ratio s that fixes the percentage of non-eliminated atoms at each iteration (*e.g.* $s = 0.9$ means that 90% of the atoms are kept within the *SpatialAtomsFiltering* function). The initialization of the algorithm as well as the description of the different steps of each recursion are detailed hereafter.

Algorithm 1: *RecursivePatchSelection*

input : Vocabulary $\mathcal{Z}$, filtering ratio s, training set $\mathcal{X}$, image labels $\mathcal{Y}$, Number of iterations T
output: Filtered Vocabulary $\mathcal{Z}^{(T)}$

if (T>1)
  $\mathcal{Z}^{(T-1)}$=*RecursivePatchSelection*($\mathcal{Z}$,s,$\mathcal{X}$, $\mathcal{Y}$, $T-1$);
else
  $\mathcal{Z}^{(T-1)}$=$\mathcal{Z}$;
Φ=*ComputeSNN*($\mathcal{X}$,$\mathcal{Z}^{(T-1)}$);
$\mathbf{W}$=*LearnSVM*(Φ,$\mathcal{Y}$);
$\mathcal{Z}^{(T)}$=*SpatialAtomsFiltering*($\mathcal{Z}^{(T-1)}$,$\mathbf{W}$,s);
return $\mathcal{Z}^{(T)}$

Initialialization: The initial vocabulary $\mathcal{Z}^{(0)}$ to be used as input of the *RecursivePatchSelection* algorithm is created by randomly picking N_0 spatial atoms within the images of the training set $\mathcal{X}$. When N_0 is very large (e.g. 1 million of potentially overlapping regions), this allows starting the vocabulary learning with an over-complete representation to be progressively reduced afterwards. More practically, we uniformly draw N_0 local features $\mathbf{z_j}$ from the raw set of all spatially localized local features extracted from the images (be they hand crafted such as SIFT features or off-the-shelf low level features learned through a convolutional neural network). The j-th spatial atom Z_j is then formed by $\mathbf{z_j}$ itself and by the set of its top-m spatially neighboring local features $\mathbf{z_j^i}$, $i \in [1, m]$.

SNN representations computation: The goal of this step is to compute intermediate representations of the images $X \in \mathcal{X}$ based on a *spatially localized vocabulary* $\mathcal{Z}$. Each image in X is supposed to be described by a set of d-dimensional spatially localized local features $X = \{\mathbf{x}\}$. To map these local features onto the N spatial atoms Z_j of the vocabulary $\mathcal{Z}$, we use the Shared Nearest Neighbor embedding method introduced in [10] and from which we can derive the following explicit embedding function:

$$\mathbf{\Phi}(X) = \sum_{\mathbf{j}=1}^{N} \Phi_j(X).\vec{e_j} = \sum_{\mathbf{j}=1}^{N} \frac{1}{|X|} \sum_{\mathbf{z} \in Z_j} \sum_{\mathbf{x} \in X} \varphi(r_{\mathbf{x}}(\mathbf{z})).\vec{e_j} \qquad (1)$$

where $r_{\mathbf{x}}(\mathbf{z}) : \mathbb{R}^d \rightarrow \mathbb{N}^+$ is a ranking function returning the rank of $\mathbf{z}$ according to its L_2 distance to $\mathbf{x}$. The function $\varphi(r)$ is a rank-based activation function that is decreasing with r and that is close to zero when the rank r becomes sufficiently high (see [10] for more details). Intuitively, each component $\Phi_j(X)$ of the high-dimensional representation $\mathbf{\Phi}(X)$ quantifies how likely it is that the image X contains the same visual pattern than the one depicted by the spatial atom Z_j. In practice, because of scalability issues, it is not possible to compute the exact rankings $r_{\mathbf{x}}(\mathbf{z})$ for all $\mathbf{x} \in \mathcal{X}$ and all $\mathbf{z} \in \mathcal{Z}$. Therefore, an approximate nearest neighbor search method is used and only the approximate K nearest neighbors of each $\mathbf{x}$ are considered in the SNN embedding.
In [10], Leveau *et al.* propose a variant of the base SNN scheme for embedding rigid geometry constraints. This is done by multiplying the activation function $r_{\mathbf{x}}(\mathbf{z})$ by a geometry consistency indicator function $\delta_{\mathbf{x}}(\mathbf{z})$ equals to one if $\mathbf{z}$ is an inlier of a RANSAC-like algorithm estimating the best affine transformation between the image X containing $\mathbf{x}$ and the spatial atom Z_j containing $\mathbf{z}$ (and zero otherwise). In our experiments, we used this variant for the datasets that involve rigid objects, *i.e.* buildings and logos.

SVM learning and spatial atoms filtering: To select the most discriminant atoms for a given classification task, we adopt a SVM-based multi class feature selection strategy first proposed in Guyon *et al.* [4] and Chapelle *et al.* [12]. We therefore consider that each image in the training set $\mathcal{X}$ is associated to a class label $y \in [1, C]$. Now, the principle is to define a filtering criterion ρ_j for the j-th component Φ_j of the representation space by analyzing the weights w_{jk} with $k \in [1, C]$ across the C one-versus-all L_2 regularized Support Vector Machine (SVM) classifiers learned on the task. A very simple and theoretically elegant filtering criterion is

the l_2 norm of the vector $\mathbf{w_j} = \sum_{k=0}^{C} w_{jk}\vec{e_i}$ so that the more optimal component j^* to remove is given by :

$$j^* = \underset{j}{\operatorname{argmin}} \sum_{k=1}^{c} w_{jk}^2 = \underset{j}{\operatorname{argmin}} |\mathbf{w_j}|_2^2 \quad (2)$$

The filtering score of an atom Z_j can then be computed as $\rho_j = |\mathbf{w_j}|_2^2$ and the filtering consists in ranking all the components thanks to ρ_j and keep only the top sN atoms (where N is the total number of atoms in $\mathcal{Z}$ and s the filtering ratio). Note that when an atom Z_j is pruned, all the local features $\mathbf{z_j^i}$ belonging to it are definitely removed from the vocabulary.

Discussion: We highlight the fact that our Recursive Visual Patch Selection algorithm (RVPS) is actually different from a classical Recursive Feature Elimination (RFE) [12]. The RFE method actually relies on a fixed representation space and attempt to find the optimal subspace by eliminating the less informative components. On the contrary, the representation space induced by our manifold learning method is evolving at each iteration. Not only some components atoms are removed from the vocabulary but the contribution of the remaining ones do evolve as well. This is mainly due to the rank-based activation function of the SNN embedding. When removing some atoms, the rank $r_{\mathbf{x}}(\mathbf{z})$ of the kept features can only decrease and, as a consequence, the contribution $\Phi_j(X)$ of the remaining atoms can only increase. So that the selected atoms do progressively increase their contribution to the representation of more and more pictures. In other words, we do progressively improve the encoding of the manifold structure of the data thanks to the selection of more and more contributive data items. If we did not recompute the SNN representations after each atom elimination step, we would select some discriminant atoms but we would not select the most generative ones.

4. EXPERIMENTS

To evaluate our method, we used three datasets of the literature: (i) **FlickrLogos32** [3] containing 2,240 images labeled with 32 logo classes (split into 1,280 training images and 960 test images without considering distractors of the original dataset), (ii) **Paris Buildings** [2] containing 6,392 photographs of labeled with 12 Parisian buildings (split into 3,199 training images and 3,193 test images, and (iii) **Oxford Flower** [8] containing 8,189 pictures labeled with 102 flower species (split into 2,040 training images and 6,149 test images). For the *FlickrLogos32* and *ParisBuildings* datasets, SIFT features were extracted around Harris-Hessian-Laplace interest points. For the OxfordFlower dataset, we rather used off-the-shelf CNN-based features learnt with the GoogleNet CNN architecture pretrained on the ImageNet dataset [1]. Images were forwarded to the inception_3a layer output leading to 784 densely sampled 256-dimensional spatially localized features for each image. All descriptors were L_2-normalized to the unit ball and square rooted. For the SNN embedding computation, we used the same parameters than in [10] except for the for the knn quality search parameter α that we fixed to $\alpha = 40\%$ and the length b of the hash codes that was fixed to 128 bits for SIFT features and 256 bits for the CNN features. We used the spatially consistent variant of the SNN embedding only for the two datasets involving rigid objects, *i.e.* *FlickrLogos32* and *ParisBuildings*. The number N_0 of random spatial atoms in the initial vocabulary $\mathcal{Z}^{(0)}$ was fixed to $N_0 = 2^{20}$ the spatial neighborhoods size of of each atom was fixed to $m = 256$ local features.

Recursive filtering impact: To study the impact of the filtering ratio s of our RVPS algorithm, we ran it with five different values, *i.e.* $s = 0.5$, $s = 0.3$, $s = 0.1$ and $s = 0.01$. The recursively computed image representations were then used as input of a L_1 regularized logistic regression (with default regularization constant $C = 1$). Figure 2 displays the resulting classification accuracy on the *FlickrLogos32* dataset as a function of the number of spatial atoms in the learnt vocabulary. It shows that if the filtering ratio is too strong (*e.g.* $s = 0.1$ or $s = 0.01$), the classification performance quickly degrades. On the other side, with a reasonable filtering ratio of $s = 0.5$ or $s = 0.3$, the classification performance remains rather stable with up to two orders of magnitude less atoms in the vocabulary. When the vocabulary contains only 256 spatial atoms, the accuracy is still very acceptable meaning that they are highly informative for the classification task.

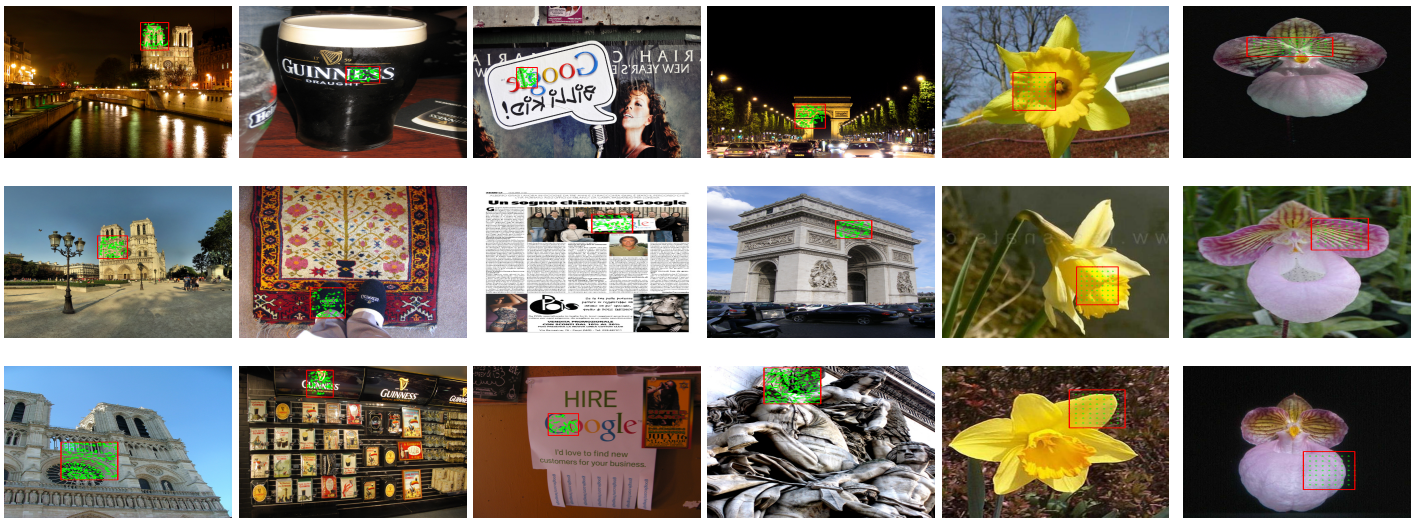
RVPS vs. RFE: To further study the effectiveness of our Recursive Visual Patch Selection algorithm (RVPS), we compared it with a classical Recursive Feature Elimination (RFE) computed on top of our initial SNN representations (*i.e.* the ones based on the initial vocabulary $\mathcal{Z}^{(I)}$). The results are provided in Table 1. They show that at constant dimensionality, the representations learned by RVPS are much more effective for the classification task than the ones learned by RFE. This is not sufficient to conclude that the selected spatial atoms are better in terms of interpretability (the generative aspect is probably even more important). But this proves that they do provide a better generalization ability which is a already an interesting criterion.

# atoms	**256**	**1,024**	**4,096**	**16,384**	**65,536**
RFE	12.19	15.94	39.2	75.6	81.32
RVPS	78.4	83.6	84.4	86.9	86.15

Table 1: RVPS vs. RFE classification accuracy

RVPS vs. CNN: we also compared the classification accuracy obtained from the RVPS representations to the one of the GoogleNet convolutional neural network [11], with or without pre-training on ImageNet (we respectively used a learning rate of 0.001 with local multipliers of 10 on the last fully connected layers and a learning rate of 0.01 without adding additional local multiplier). For both finetuning and no-finetuning, we used a weight decay parameter of 0.0005 and a momentum of 0.9. The results are provided in Table 2 for the three datasets. They show that the RVPS-based representations are quite competitive with the CNN ones with slightly lower performance than the network fine-tuned on ImageNet but much better performance than the one trained on the same data than our RVPS method. Now, the main advantage of RVPS is to allow interpreting very easily which visual patterns of the training set were learnt. Indeed, each spatial atom of the spatially localized visual vocabulary has a uniquely defined visual representation.

Interpretabilty of the learnt vocabulary: to qualitatively illustrate how interpretable the spatial atoms of our visual rep-



Notre Dame | Guiness | Google | Arc de Triomphe | Daffodil | Hard-leaved pocket orchid

Figure 1: Learned Spatial Atoms for 6 classes

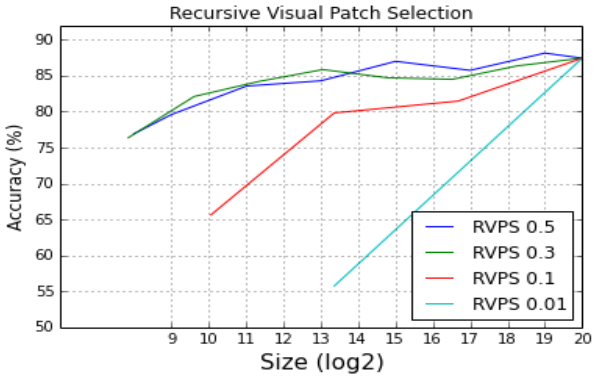


Figure 2: RVPS accuracy vs. number of spatial atoms

Method	*FlickersLogos*	*Paris*	*Flower*
GoogleNet FT	87.5	70.5	89.56
GoogleNet no FT	66.8	54.4	61.7
RVPS - 16,384 atoms	86.9	73.2	86.43
RVPS - 4,096 atoms	84.4	70.9	86.36
RVPS - 1,024 atoms	83.6	67.6	84.31

Table 2: RVPS vs. CNN classification Accuracy

resentations are, Figure 1 diplays the top-3 most contributive atoms of several classes based on a vocabulary of size 1024 atoms for each dataset. We therefore trained one-vs-all SVM's on top of our representations and ranked the atoms according to their class-wise weight w_{jk}. We see that for some domain such as logos, the chosen patches often correspond to variants of the same visual pattern or to the same visual pattern but under different view conditions. For more complex visual entities such as buildings or plants, the chosen patches rather correspond to different parts of the whole entity which might be very helpful for domain experts.

5. CONCLUSION AND PERSPECTIVES

This paper addressed the challenge of devising a new representation learning algorithm to overcome the lack of *interpretability* of classical visual models. We did show that the proposed Recursive Visual Patch Selection algorithm allows to learn highly discriminant and interpretable representations that maps the visual content of the images onto a vocabulary of uniquely defined and spatially localized visual atoms. In further works, we will attempt to devise innovative active learning methods allowing the experts to directly interact with the vocabulary.

6. REFERENCES

[1] A. Krizhevsky, et al. Imagenet classification with deep convolutional neural networks. In *NIPS'12*.

[2] J. Philbin et al. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR'08*.

[3] S. Romberg et al. Scalable logo recognition in real-world images. In *ICMR'11*.

[4] I. Guyon et al. Gene Selection for Cancer Classification Using SVM. In *Machine Learning Journal'02*.

[5] T. Urruty et al, Iterative Random Visual Word Selection. In *ICMR'14* .

[6] J. Krapac et al. Instance classification with prototype selection. In *ICMR'14*.

[7] G. Csurka et al. Visual categorization with bags of keypoints. In *ECCV04*.

[8] M-E. Nilsback and A. Zisserman. Automated Flower Classification over a Large Number of Classes. In *Indian Conference on Computer Vision, Graphics and Image'08*.

[9] J. Mairal et al. Supervised Dictionary Learning. In *NIPS'08*

[10] V. Leveau et al. Kernelizing Spatially Consistent Visual Matches for Fine-Grained Classification. In *ICMR 2015*.

[11] C. Szegedy et al. Going Deeper with Convolutions. In *CVPR'15*.

[12] O. Chapelle et al. Multi-class feature selection with svm. In Proceedings of the ASA.

[13] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR'07*.

[14] H. Jegou et al. Aggregating local descriptors into a compact image representation. In *CVPR"10*.