Exemplar SVMs as Visual Feature Encoders

Joaquin Zepeda and Patrick Pérez
Technicolor

Abstract

In this work, we investigate the use of exemplar SVMs (linear SVMs trained with one positive example only and a vast collection of negative examples) as encoders that turn generic image features into new, task-tailored features. The proposed feature encoding leverages the ability of the exemplar-SVM (E-SVM) classifier to extract, from the original representation of the exemplar image, what is unique about it. While existing image description pipelines rely on the intuition of the designer to encode uniqueness into the feature encoding process, our proposed approach does it explicitly relative to a "universe" of features represented by the generic negatives. We show that such a post-processing enhances the performance of state-of-the art image retrieval methods based on aggregated image features, as well as the performance of nearest class mean and K-nearest neighbor image classification methods. We establish these advantages for several features, including "traditional" features as well as features derived from deep convolutional neural nets. As an additional contribution, we also propose a recursive extension of this E-SVM encoding scheme (RE-SVM) that provides further performance gains.

1. Introduction

Exemplar SVMs (E-SVMs), proposed by Malisiewicz and Efros [21], are linear classifiers learned from a single positive example, referred to as the *exemplar*, and a pool $\mathcal{N}$ of generic examples that is used as the set of negative examples. Despite several shortcomings of the approach (see Section 2), exemplar SVMs have given good results in a wide range of tasks requiring generalization in data-constrained scenarios, e.g., [2, 20, 30]. This results from the ability of the approach to extract, in the form of a linear classifier, what is unique about a specific image (given a generic representation of it), relative to a universe of features stemming from the task-tailored pool of negative examples. This is the property that we wish to leverage in our work.

In this paper we propose to use E-SVM as an encoding mechanism that turns generic image features into better task-tailored representations (Fig. 1). In particular, our approach can be used to enhance the performance of state-of-the art image retrieval methods such as those of [7, 9].

Extracting distinctive signatures from images, with subsequent use for image comparison, retrieval or classification, is the aim of feature encoder in existing image representation pipelines. Such encoders, however, often rely on the intuition of the designer. In contrast, our approach performs this extraction explicitly relative to a universe of features consistent with the targeted application.

Besides the utilization of basic E-SVMs as an encoding mechanism, we also propose using E-SVM learning in a recursive framework: An initial set of E-SVM features is extracted for the pool of generic negatives, and the process is repeated a certain number of times by using the resulting E-SVM features as input features in the subsequent recursion. This method shares some lineage with the now widely popular deep approaches that appeared following the success of [17]. As we shall demonstrate, encoding image features with recursive E-SVM (RE-SVM) further improves image search performance.

The rest of the paper is organized as follows: in Section 2, related work on image representation and exemplar-SVMs is discussed into more depth; We introduce proposed E-SVM encoding method in Section 3, along with its use in the context of image search and its recursive variant; Section 4 is devoted to implementation details and to experiments in the context of image retrieval; We outline several perspective to this work in Section 5, before concluding.

2. Related work

2.1. Exemplar SVMs (E-SVMs)

Exemplar SVM have been introduced by Malisiewicz and Efros [21] to address the task of learning a classifier from a single positive exemplar and a set $\mathcal{N}$ of negative examples. For a given exemplar image, the approach produces a linear classifier that captures distinctive aspects of the exemplar relative to the "universe" represented by the generic negatives pool.

Training good exemplar SVMs requires that the generic negative set $\mathcal{N}$ be very large, consisting of as many as

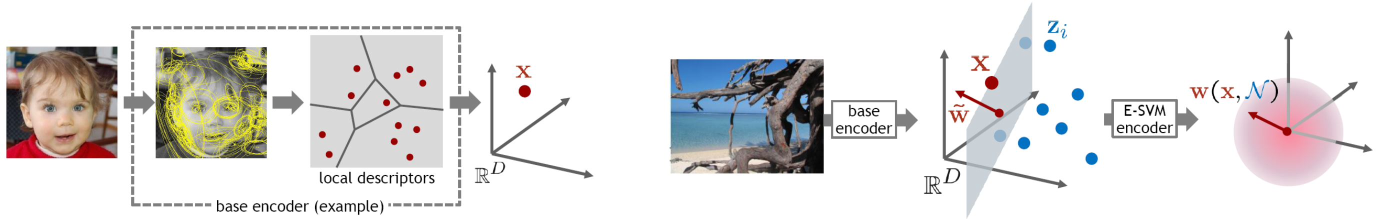

Figure 1. **Principle of E-SVM visual encoder**. (Left) Given a generic visual encoder, like BoW, Fisher vector or VLAD, an image is described as a fixed size feature vector $\mathbf{x} \in \mathbb{R}^D$; (Right) Using a pool of generic negative image features $\mathcal{N} = \{\mathbf{z}_i\}_{i=1}^N$, an E-SVM $\tilde{\mathbf{w}}$ is learned for each input image. The ℓ_2-normalized E-SVM $\mathbf{w}$ is the new encoding of the image for subsequent analysis.

one million items [20]. Besides enhancing the discriminative power of the classifier, larger negative sets make the training process stable relative to the choice of regularization weights, as well as robust to the presence of eventual false negatives in $\mathcal{N}$. In order to make the training process tractable, employing hard-negative mining [6, 8] as part of the SVM learning process, which has been shown to converge to the true solution [8], is a must. The process consists of keeping a hard-negative cache that is a subset of $\mathcal{N}$, and alternately *i)* training a classifier using this hard-negatives cache in place of $\mathcal{N}$, and *ii)* growing the cache using examples from $\mathcal{N}$ having a classification score inside the margin (*i.e.* greater than -1). Despite the use of hard negative mining, the complexity required to train E-SVMs with the requisite size of generic negative set is a shortcoming of the approach.

A second shortcoming of E-SVMs is that, despite the size of the negative set, there is only so much generalization that can be extracted from a single positive exemplar [2]. Indeed E-SVM-based approaches dealing with object detection address this issue by producing extra positives from each exemplar image patch by applying small transformations (*e.g.* shifts, scaling) to it [30], effectively using multiple positive examples.

Despite these shortcomings, exemplar SVMs have given good results in a wide range of tasks requiring generalization in data-constrained scenarios. Malisiewicz *et al.* [20] proposed using E-SVMs to transfer meta-data (*e.g.* object segmentation or pose information) from a set of annotated exemplars to an unannotated set. Generic object detection has also been addressed using ensembles of E-SVMS, where each E-SVM in the ensemble corresponds to one positive example of a given class. Using logistic regression-based calibration makes it possible to compare the scores of the various E-SVMs in the ensemble. The regression is carried using as positives those items from the search database for which the E-SVM returns the highest positive score, similarly to approaches used in unsupervised mid-level feature discovery [32]. Shrivastava *et al.* [30] proposed using E-SVMs for cross-domain search, where the exemplar is an image in a given domain (*e.g.* a hand-drawn sketch, a painting or a photograph), and the targeted search images are representations in a different domain. One of the applications considered in that work is image retrieval in the sense of [14], where both the query image and the targeted search images are photographs. This is the application considered herein, but in their approach, an E-SVM is computed only for the query image, and their method underperforms relative to approaches based on features tailored for the image retrieval task. Another E-SVM based method [2] addresses the decreased generalization power resulting from using a single positive exemplar by constraining the learned E-SVM classifier to be close (under the ℓ_2 norm) to a linear combination of generic SVM classifiers. The resulting approach gives improved performance relative to standard E-SVMs in the task of pose-specific object retrieval.

In this paper we propose turning exemplar SVMs into feature maps that can be used to post-process generic image features. In the context of image search, this is unlike previous E-SVM-based approaches which are asymmetric in the sense that an exemplar SVM is learned only for the query image and subsequently applied as a classifier to the features extracted from the search database. For image classification, our approach also differs from previous attempts to exploit E-SVMs in its way to deal with the lack of generalization power [2] that is a consequence of single-exemplar learning: When applied to classification, our symmetric E-SVM approach overcomes this drawback by leaving the task of generalization to a standard classifier based on multiple positive and negative examples.

2.2. Ad-hoc and learned image representations

Common to both search and classification tasks is the need to encode the image into a single, fixed-dimensional feature vector. Many successful image feature encoders operate on ad-hoc, fixed-dimensional local descriptor vectors extracted from densely [5, 1] or sparsely [19, 23] sampled local regions of the image. The feature encoder aggregates these local descriptors to produce a higher dimension image feature vector (Fig.1, left). Examples of such feature encoders include the bag-of-words encoder [33], the Fisher encoder [26] and the VLAD encoder [13, 14]. All these aggregation methods depend on specific models of the data distribution in the local-descriptor space that are learned in an unsupervised task-independent manner. For bag-of-words and VLAD, the model is a codebook obtained using K-means, while the Fisher encoding is based on a Gaussian Mixture Model (GMM). These pipelines have proved very effective for a variety of image analysis tasks.

The approach we propose herein is a task-dependent post-processing mechanism that can be applied to any of the aggregated image features described above (Fig. 1, right). In this respect, our method is similar to that of Tolias *et al.* [35], wherein a kernel similar to the popular Hellinger kernel is shown to provide a large improvement in the image retrieval task. Yet their method is only established to perform well on very high dimensional base features that are impractical in complexity constrained or large scale scenarios.

Another interesting manner to make image encoding task-dependent is to adapt, through appropriate learning, some parts of the whole pipeline. This idea can be leveraged to learn local descriptors [4, 31] or the model used for aggregation, e.g., GMM used in the Fisher encoding [34]. Approaches based on deep Convolutional Neural Networks (CNNs)[17, 24] can also be interpreted as feature learning methods, and these now define the new state-of-the art baseline in semantic search. Our approach is different, and complementary, in the sense that E-SVM encoder can be used on top of any initial feature of interest, whether generically engineered or already optimized according to one of the approaches mentioned above. We shall see, in particular, that our approach can operate on CNN-based image representations. Another methodological difference lies in the fact that, in our proposed approach, each individual image encoding must resort to its own learning routine. While this obviously comes at a certain computational cost, it is trivially parallelizable and is a source of great flexibility. Furthermore, we show that the computational overhead is lower than the computational cost of standard feature extractors built by aggregating local descriptors.

Our approach is also somewhat related to so-called explicit feature maps [36] that embed original image feature vector into another space where dot product similarity provides a good approximation of a given kernel of interest. Such an explicit embedding is nonetheless task-independent, being only driven by the choice of the kernel and, it usually yields an increase of the original feature dimension. By contrast, our approach maps discriminatively the input image feature to a new feature of identical dimension.

2.3. Methods based on deep Convolutional Neural Networks (CNNs)

Starting with the eye-opening results of Krizhevsky, Sutskever, and Hinton [17], deep Convolutional Neural Networks (CNNs) have become an important tool of the computer vision researcher's arsenal. CNN architectures can be used as feature extractors by using the activation coefficients at the output of the first fully-connected layer directly as a feature [28] or by combining CNNs with pyramid methods [10, 11] or other traditional approaches such as bag-of-words or Fisher encoding [18]. One drawback with these types of approaches is their large complexity, as a CNN architecture consists of many tens of millions of coefficients. Approaches such as that in [10] that further use the CNN pipeline as a local feature extractor over a dense grid result in astronomical complexity, restricting their applicability to large scale settings considered herein. Yet their simplicty of construction and performance make them pertinent research methods.

3. Proposed approach

3.1. Feature encoding with E-SVMs

We assume that a generic, D-dimensional image feature encoder is given. This base encoder can be global, based on aggregated local features, or derived from CNNs-based features. We shall denote by vectors in $\mathbb{R}^D$ such features. An exemplar SVM can be computed from the exemplar feature vector $\mathbf{x}$ and a large set of generic feature vectors $\mathcal{N} = \{\mathbf{z}_i\}_{i=1}^N$ by solving the following optimization problem:

$$\tilde{\mathbf{w}}(\mathbf{x},\mathcal{N}) = \underset{\mathbf{w}\in\mathbb{R}^D}{\operatorname{argmin}} \Big[\frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \alpha_1 \max(0, 1-\mathbf{x}^\top\mathbf{w}) + \alpha_{-1}\sum_{i=1}^N \max(0, 1+\mathbf{z}_i^\top\mathbf{w})\Big], \tag{1}$$

where λ, α_1 and α_{-1} are positive parameters that control the level of regularization and the relative weight of negative examples. For convenience, throughout we will we refer to E-SVMs as the ℓ_2-normalized version of the solution to the above problem:

$$\mathbf{w}(\mathbf{x},\mathcal{N}) = \frac{\tilde{\mathbf{w}}(\mathbf{x},\mathcal{N})}{\|\tilde{\mathbf{w}}(\mathbf{x},\mathcal{N})\|_2}. \tag{2}$$

When dependence on $\mathbf{x}$ and $\mathcal{N}$ is clear from the context, we shall simply denote $\mathbf{w}$ this E-SVM.

Optimization problem (1) is a classic linear SVM problem relying on hinge loss, with the notable particularity that positive and negative sets are extremely unbalanced, one positive for up to, say, one million negatives. In [21], the property of hinge loss to yield dual solutions dependent only on a small number of (negative) support vectors is leveraged through hard negative mining. As an alternative efficient solver, we shall rely on stochastic gradient descent (see details in Section 4.1).

We propose using E-SVMs thus computed as new features. Hence we assume that we are given a first feature encoder, task-dependent or not, that produces feature vector $\mathbf{x}$ from a given image, but we instead use $\mathbf{w}(\mathbf{x}, \mathcal{N})$ as the task-dependent feature representation for said image. Two particular aspects of this encoding are worth emphasizing:

- While E-SVM is a linear SVM, the resulting encoding, even before normalization, is obviously not linear relative to base feature $\mathbf{x}$;
- This is a dimension preserving encoding since the new image representation still lives in $\mathbb{R}^D$. This is in stark contrast with high-dimensional encoding using for instance Fisher vectors [16] or explicit feature maps that approximate infinite-dimensional kernel maps [36].

The proposed visual encoding approach is illustrated in Fig. 1.

3.2. Symmetric encoding for image search

As demonstrated in [21], the E-SVM $\mathbf{w}^\circ = \mathbf{w}(\mathbf{x}^\circ, \mathcal{N})$ attached to a given image $\mathbf{x}^\circ$ can be used on its own to retrieve images with very similar content in a dataset $\mathcal{D} = \{\mathbf{x}_j\}_{j=1}^M$, using scores $\mathbf{x}_j^\top \mathbf{w}^\circ$. We propose instead a symmetric approach where each image $\mathbf{x}_j$ in the dataset is also equipped with its E-SVM feature $\mathbf{w}_j = \mathbf{w}(\mathbf{x}_j, \mathcal{N})$. Our approach then consists in sorting all these according to their similarity

$$s_j = \mathbf{w}_j^\top \mathbf{w}^\circ \tag{3}$$

with the E-SVM of the query image.

3.3. Recursive E-SVMs encoding

The above proposition of post-processing the output $\mathbf{x}$ of any generic feature encoder to produce E-SVM features $\mathbf{w}(\mathbf{x}, \mathcal{N})$ suggests applying this procedure recursively. We can formalize this approach by first defining $\mathbf{w}^0 \triangleq \mathbf{x}$ and $\mathcal{N}^0 \triangleq \mathcal{N}$. The k-th recursion of E-SVM feature computation can then be written as follows for $k \geq 1$:

$$\mathbf{w}^k = \mathbf{w}(\mathbf{w}^{k-1}, \mathcal{N}^{k-1}), \tag{4}$$

$$\text{where } \mathcal{N}^k = \{\mathbf{w}(\mathbf{z}, \mathcal{N}^{k-1}), \mathbf{z} \in \mathcal{N}^{k-1}\}. \tag{5}$$

Features built using the k-th recursive E-SVM (RE-SVM) procedure specified in (4) can be used in a manner analogous to (3) to carry out image retrieval.

The recursive E-SVM feature construction approach in (4) is reminiscent of deep architectures, popularized following the success of [17], that use the output of a given layer as the input to the subsequent layer. Unlike those approaches, however, the feature in (4) is learned on a per-image basis and in a completely un-supervised manner. Furthermore, the computation of each $\mathbf{w}^k$ is done by means of a single, non-linear, convex problem, as opposed to the standard tandem linear/non-linear arrangement used in each layer in approaches derived from [17].

4. Experiments

4.1. Implementation Details

Base visual encoding As our base image features, we use a recent variant of the VLAD encoder [7] which is computed by power-normalizing (element-wise $\operatorname{sign}(x)|x|^{0.2}$ operation) and ℓ_2 normalizing the following concatenated vector:

$$\left[\mathbf{\Phi}_k^\top \sum_{\mathbf{s} \in \mathcal{S} \cap \mathcal{C}_k} \frac{\mathbf{s} - \mathbf{c}_k}{\|\mathbf{s} - \mathbf{c}_k\|} \right]_k, \tag{6}$$

where $\mathcal{S}$ is the set of local descriptors extracted from the image, the $\mathbf{c}_k$'s are codewords obtained using K-means and $\mathbf{\Phi}_k$ is the local PCA basis obtained from the set $\mathcal{S} \bigcap \mathcal{C}_k$ of image local descriptors that lie in cell $\mathcal{C}_k$ associated to k-th codeword. As in [7], we use a training set randomly chosen from Flickr images and use local SIFT descriptors densely extracted at three scales.

E-SVM computation We use the PEGASOS stochastic gradient descent primal SVM solver [29, 3] to compute exemplar SVMs, using a re-sampling strategy to implicitly choose the penalty weights α_1 and α_{-1} for the exemplar and the negative pool. To illustrate the approach, we can rewrite the objective in (1) as follows, where $y_i = -1, \forall i = 1, \ldots, N$, $y_{N+1} = 1$, and we let $\mathbf{z}_{N+1} \triangleq \mathbf{x}$:

$$\frac{1}{\alpha_1 + N\alpha_{-1}} \sum_{i=1}^{N+1} \alpha_{y_i} \big(\frac{\lambda}{2} \|\mathbf{w}\|^2 + \max(0, 1 - y_i \mathbf{z}_i^\top \mathbf{w}) \big). \tag{7}$$

The expectation over i of the gradient of the term inside the summation can be controled by the α_1, α_{-1} parameters, or by using the exemplar every f_p random draws from the negative pool during the SGD optimization, which we found to converge faster. In order to add stability to the RE-SVM representation, we use the same random ordering of the negative pool to compute all RE-SVM features. The resulting implementation allows us to compute E-SVM features in close to 600 ms for the longest SGD runtimes considered ($100,000$ iterations).

The synopsis of the algorithm is provided in Alg.1, where we let $\mathbf{1}_{a<b} = 1$ if $a < b$ and 0 otherwise.

Algorithm 1. E-SVM feature encoding with PEGASOS.

```
Input: $\mathbf{x}$, $\mathcal{N} = \{\mathbf{z}_i\}_{i=1}^{N}$, $\lambda$, $T$, $f_p$
Initialize: set $\mathbf{w}_1 = \mathbf{x}$
for $t = 1, \ldots, T$ do
    if $t \bmod f_p \neq 0$ then
        Choose random $\mathbf{z}$ from $\mathcal{N}$, without repetition
        Set $y = -1$
    else
        Set $\mathbf{z} = \mathbf{x}$, $y = 1$
    end if
    Set $\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{1}{\lambda t}(\lambda \mathbf{w}_t - \mathbf{1}_{y\mathbf{z}^\top \mathbf{w} < 1} y \mathbf{z})$
end for
Output: $\mathbf{w} = \frac{\mathbf{w}_{T+1}}{\|\mathbf{w}_{T+1}\|}$
```

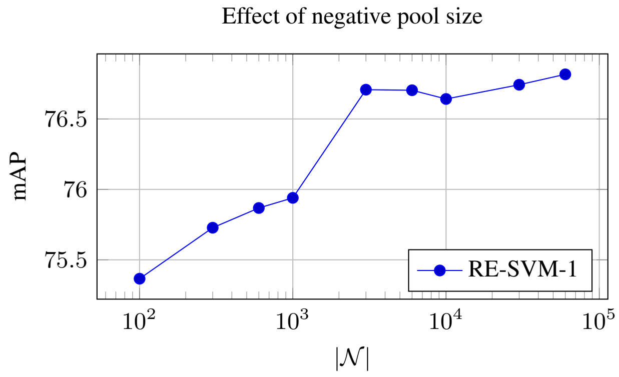

Figure 2. Plot of mean Average Precision (mAP) on Holidays dataset as a function of $|\mathcal{N}|$ when using $T = 1e5$ SGD iterations, $\lambda = 1$, and $f_p = 10$.

4.2. Image retrieval

Datasets and protocol We evaluate our algoritm on two publicly available datasets, *Holidays* [12] and *Oxford* [27]. The *Holidays* dataset consists of 1491 images of vacation shots divided into 500 groups of matching images. The second dataset, the Oxford dataset, consists of close to 5000 images of buildings from the city of Oxford. The images are divided into 55 groups of matching images, and a query image is specified for each group. We use the full image as the query image instead of the cropped region. Both datasets include a specfic evaluation protol based on mean Average Precision (mAP) that we use throughout.

Effect of parameters In Figs. 2-5 we evaluate the effect of the RE-SVM encoding parameters on mAP performance on the Holidays dataset.

In Fig. 2, we evaluate the effect of the negative pool size N on the performance of the system and observe that the performance increases with larger negative pools. In latter experiments we fix the pool size to $N = 60e3$. Using larger pools could indeed increase the system performance, but this is at the expense of a larger encoder memory footprint and exploiting larger pools further requires longer SGD runtimes.

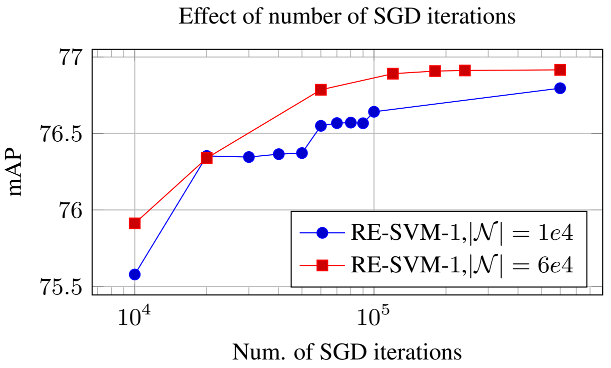

Figure 3. mean Average Precision (mAP) on Holidays dataset as a function of the number T of SGD iterations when using $|\mathcal{N}| = 1e4$ or $|\mathcal{N}| = 6e4$, $\lambda = 1$ and $f_p = 10$.

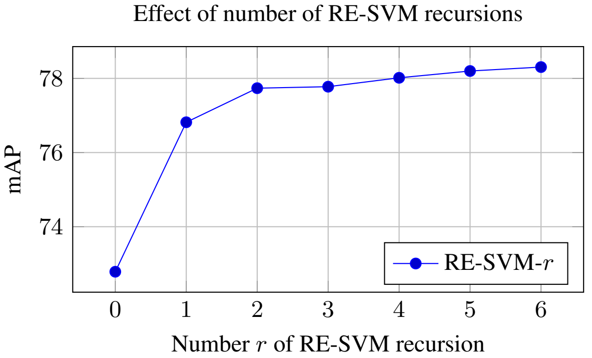

Figure 4. mean Average Precision (mAP) on Holidays dataset as a function of the number r of RE-SVM recursions when using $N = 60e3$, $T = 1e5$ SGD iterations, $\lambda = 1$, and $f_p = 10$. The point for $r = 0$ corresponds to the baseline using VLAD-64.

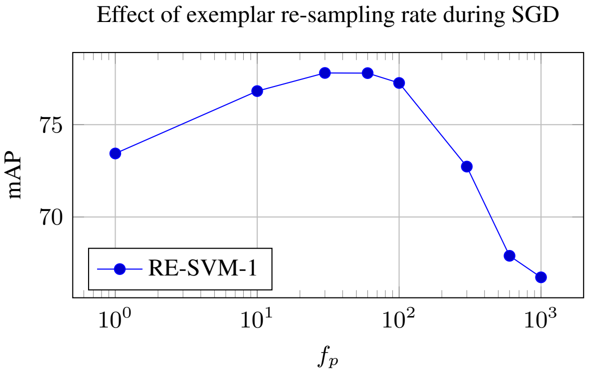

Figure 5. Mean Average Precision (mAP) on Holidays dataset as a function of the exemplar re-sampling rate f_p when using $N = 60e3$, $T = 1e5$ SGD iterations and $\lambda = 1$.

Iterations	1e3	10e3	100e3
Run-time	10 ms	70 ms	620 ms

Table 1. E-SVM encoding runtime when using a single core running at 2.6 GHz.

In Fig. 3, we evaluate the effect of the number of SGD iterations for two different pool sizes ($N = 10e3$ and $60e3$). For our pool size of $60e3$, the benefit of increasing the number T of iterations saturates after $100e3$ iterations, and hence we use $T = 100e3$ iterations in latter experiments. In Table 1 we provide runtimes for E-SVM learning and show that using $100e3$ iterations takes 620 ms with our non-optimized C implementation.

In Fig. 4, we evaluate the merit of recursively training RE-SVMs, as discussed in Section 3. We plot RE-SVM results when using r recursions (referred to as RE-SVM-r), for $r = 0, ..., 6$, where $r = 0$ refers to the baseline results obtained with the VLAD encoder. A single RE-SVM recursion produces a gain of close to 4 mAP points relative to the VLAD encoder, and using 6 recursions produces a gain of close to 5 points. Since most of the gain is obtained by using only 2 recursions, we will use $r = 2$ in latter experiments.

In Fig. 5 we evaluate the effect of varying the exemplar sampling rate during the SGD optimization process. Note that any value between $f_p = 1$ and $f_p = 400$ results in an improvement relative to the baseline VLAD encoder. This is consistent with the findings of [21] concerning the robustness of E-SVMs to choice of balancing weights. In latter experiments we will use a value of $f_p = 30$.

Large scale experiments In Fig. 6 and Fig. 7 we evaluate the robustness of our method to the addition of a large number of distractor images from Flickr different from those used for the negative pool. The distractor images are encoded in the same manner as the benchmark images using VLAD + RE-SVM-2. The parameters used for both RE-SVM recursions (see Alg.1) are

$$N = 60e3,\ T = 100e3,\ \text{ and } f_p = 30, \tag{8}$$

according to the discussion that followed evaluations on the Holidays dataset. Note that the same parameters selected on Holidays also give an important improvement in the Oxford dataset. Moreover, this improvement is constant over the entire range of distractor images considered. For the Holidays dataset, the improvement is in excess of 5 mAP points for the entire range of distractor images. For the Oxford dataset, the improvement is in excess of 10 mAP points likewise for the entire range of distractor images.

Applicability of RE-SVMs to other base features In order to test the applicability of our method to generic features, we also carry out experiments using bag-of-words [33] and the Fisher vector [25], both computed over densely extracted local SIFT descriptors.

The Bag-of-Words (BoW) feature is based on a codebook $\{\mathbf{c}_k\}_k$ and is obtained by ℓ_2 normalizing the following histogram of quantized local descriptors,

$$\left[|\mathcal{S} \cap \mathcal{C}_k|\right]_k, \tag{9}$$

where $\mathcal{S}$ represents the set of local descriptors extracted from the image and $\mathcal{C}_k$ is the Voronoi cell associated to codeword $\mathbf{c}_k$. We build BoW features using a codebook size of 1000.

The Fisher encoding is based on a Gaussian mixture model of the local descriptor space. We use the ℓ_2 normalized version of the first order variant given by

$$\left[\sum_{\mathbf{s}\in\mathcal{S}} \frac{p(k|\mathbf{s})}{\sqrt{\beta_k}} \mathbf{\Sigma}_k^{-1} (\mathbf{s} - \mathbf{c}_k)\right]_k, \tag{10}$$

where β_k, $\mathbf{c}_k$ and $\mathbf{\Sigma}_k$ denote, respectively, the k-th mixture component prior weight, mean vector and correlation matrix (constrained to be diagonal). We use 64 mixture components in our experiments.

In Table 2, we illustrate the performance of the base VLAD-64, BoW-1000 and Fisher-64 encodings on the Holidays and Oxford benchmarks, along with the performance of the RE-SVM-1 and RE-SVM-2 features derived from each encoding. As illustrated in the table, even a single RE-SVM recursion gives a large boost to all three encodings. The Fisher vector, in particular, performs poorly initially, but gains as many as 35 mAP points (on the Holidays dataset) after two RE-SVM recursions to outperform BoW.

We also compare against the CNN-based method proposed by [28], as well as our own, better-performing implementation of their system based on CAFFE [15]. Their approach consists of using the activation coefficients from a fully connected layer of a deep CNN architecture as an image feature for retrieval. In order to focus on the discerning power of the feature, we neglect voting and augmentation mechanisms [28] that are orthogonal to the specific feature construction method, and which have the adverse effect of increasing system complexity and feature dimensionality. As shown in the table, our system also gives an important advantage (3.6 mAP points) when using such CNN-based features as base features.

In Fig. 8 and Fig. 9 we show, respectively, example queries for which our proposed approach improves and worsens the rank of a matching image. Note that the examples of worsened performance in Fig. 9 contain mostly image pairs in different vertical/horizontal disposition. We believe that such cases could be easily addressed using a positive set obtained by applying to the exemplar, simple transformations including rotations, mirroring, displacement, cropping, and potentially others.

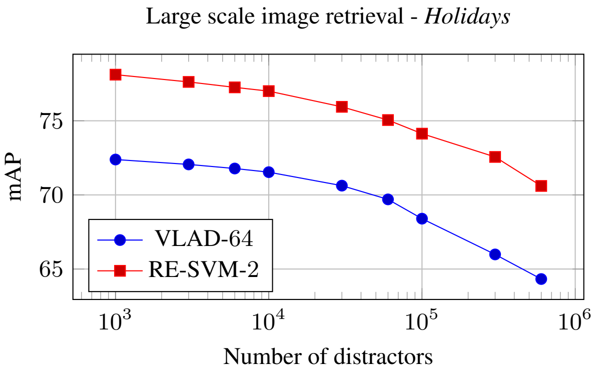

Figure 6. Mean Average Precision (mAP) on Holidays dataset as a function of the number of distractors when using $N = 60e3$, $T = 1e5$ SGD iterations, $\lambda = 1$, and $f_p = 30$.

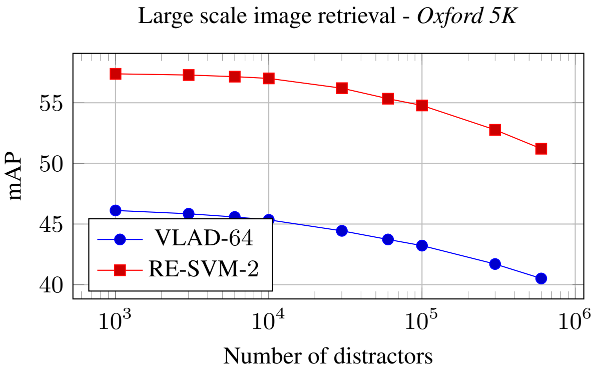

Figure 7. Mean Average Precision (mAP) on Oxford dataset as a function of the number of distractors when $N = 60e3$, $T = 1e5$, $\lambda = 1$, and $f_p = 30$.

	Holidays	Oxford 5K
VLAD-64	72.7	46.3
VLAD-64 + RE-SVM-1	77.5	55.5
VLAD-64 + RE-SVM-2	**78.3**	**57.5**
Fisher-64	18.2	9.27
Fisher-64 + RE-SVM-1	59.8	27.7
Fisher-64 + RE-SVM-2	63.6	31.5
BoW-1000	38.6	17.0
BoW-1000 + RE-SVM-1	44.5	20.5
BoW-1000 + RE-SVM-2	49.1	25.5
CNN [28]	64.2	32.2
CNN [15]	68.2	40.6
CNN [15] + RE-SVM-1	71.3	43.9
CNN [15] + RE-SVM-2	71.8	44.6

Table 2. Results for VLAD, BoW, Fisher and CNN encodings and their RE-SVM-1 and RE-SVM-2 variants.

Classifier	CNN [15]	+ RE-SVM-1
NCM	51.8	55.5
K-NN 3	60.7	62.2
K-NN 5	65.7	66.5
K-NN 10	68.9	69.8

Table 3. Results (mAP) on the Pascal VOC image classification task when using CNN [15] as a base feature.

4.3. Image Classification

In Table 3 we test our method in the Pascal VOC image classification task when using either Nearest Class Mean (NCM) or K-Nearest Neighbors (K-NN) classifiers, showing improvements of close to 4 mAP points for the NCM classifier and up to 2 mAP points for the K-NN classifier. NCM and K-NN classifiers have the important advantage that new classifiers can be added at near zero cost [22], contrary to one-vs-rest approaches using linear classifiers that require that the classifiers for all classes be updated periodically when adding new classes. NCM in particular further enjoys a very low testing cost. We also tested our approach using linear classifiers and found that the RE-SVM-1 variant resulted in a negligible drop in performance of < 1 mAP point.

5. Conclusion

In this work we proposed using Exemplar Support Vector Machines (E-SVMs) as an image feature encoder applicable to generic image features such as VLAD, Fisher, Bag-of-Words and CNN-derived features. Our approach is in contrast to existing approaches that compute E-SVMs only from one image and use the resulting E-SVM as a classifier applied to features of the original representation. We further propose computing E-SVMs recursively from E-SVM encoded features, an approach we refer to as Recursive Exemplar SVMs (RE-SVMs).

We test our method on the image retrieval task using a variety of features and show that it can give an improvement of as much as 5 points in mean Average Precision (mAP) relative to high-performing VLAD encodings. We further carry out large scale tests with large numbers of distractor images equally represented using RE-SVMs and show that our performance gain is robust to distractor images. We further show that our proposed method has wider applications in the image classification task, and we believe wider applications are possible, including image-related tasks such as registration but also generic tasks that require fixed-dimensional feature representations.

Aknowledgements This work was partly supported by European integrated project AXES

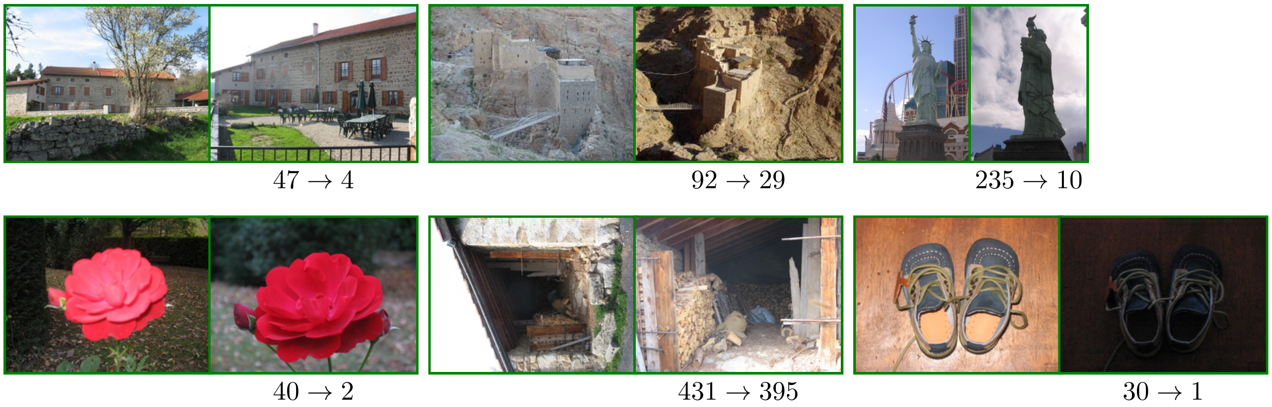

Figure 8. Select images that get ranked better when using the RE-SVM-2 encoding than when using VLAD-64. For each pair, the left image is the query and the right image is a match, with the change in rank indicated below the match.

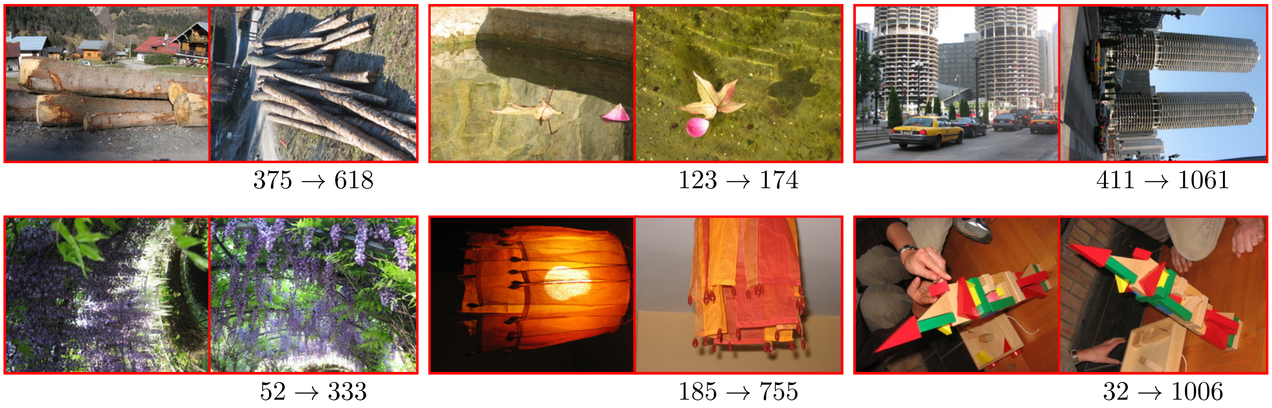

Figure 9. Select images that get ranked worse when using the RE-SVM-2 encoding than when using VLAD-64. For each pair, the left image is the query and the right image is a match, with the change in rank indicated below the match.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. *Computer Vision and Pattern Recognition*, 2012. 2

[2] Y. Aytar and A. Zisserman. Enhancing Exemplar SVMs using Part Level Transfer Regularization. *British Machine Vision Conference*, 2012. 1, 2

[3] L. Bottou. Stochastic gradient descent tricks. In G. Montavon, G. Orr, and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1. Springer, 2 edition, 2012. 4

[4] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):43–57, 2011. 3

[5] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. *British Machine Vision Conference*, 2011. 2

[6] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *Computer Vision and Pattern Recognition*, 2005. 2

[7] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the VLAD image representation. *ACM International Conference on Multimedia*, 2013. 1, 4

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–45, 2010. 2

[9] T. Ge and K. He. Product Sparse Coding. *Computer Vision and Pattern Recognition*, 2014. 1

[10] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multiscale Orderless Pooling of Deep Convolutional Activation Features. *European Conference on Computer Vision*, 2014. 3

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual

Recognition. *European Conference on Computer Vision*, 2014. 3

[12] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010. 5

[13] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. *Computer Vision and Pattern Recognition*, 2010. 2

[14] H. Jegou, F. Perronnin, M. Douze, S. Jorge, P. Patrick, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2011. 2

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, and U. C. B. Eecs. Caffe : Convolutional Architecture for Fast Feature Embedding. *ACM International Conference on Multimedia*, 2014. 6, 7

[16] S. Jorge, F. Perronnin, and Z. Akata. Fisher Vectors for Fine-Grained Visual Categorization. *Computer Vision and Pattern Recognition*, 2011. 4

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 2012. 1, 3, 4

[18] P. Kulkarni, J. Zepeda, F. Jurie, P. Perez, and L. Chevallier. Hybrid Multi-Layer Deep CNN / Aggregator Feature for Image Classification. *IEEE Int. Conf. Audio Acoustics and Speech Processing*, 2015. 3

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

[20] T. Malisiewicz, A. Gupta, and A. a. Efros. Ensemble of exemplar-SVMs for object detection and beyond. *International Conference on Computer Vision*, 2011. 1, 2

[21] T. Malisiewicz, A. Shrivastava, A. Gupta, and A. A. Efros. Exemplar-SVMs for Visual Object Detection, Label Transfer and Image Retrieval. *International Conference of Machine Learning*, 2012. 1, 3, 4, 6

[22] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric Learning for Large Scale Image Classification : Generalizing to New Classes at Near-Zero Cost. *European Confernce on Computer Vision*, 2012. 7

[23] K. Mikolajczyk, T. Tuytelaars, C. Schmid, a. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. 2

[24] J. Oquab, M. and Bottou, L. and Laptev, I. and Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. *Computer Vision and Pattern Recognition*, 2014. 3

[25] F. Perronnin, C. Dance, and D. Maupertuis. Fisher Kernels on Visual Vocabularies for Image Categorization. *Computer Vision and Pattern Recognition*, 2007. 6

[26] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. *European Conference on Computer Vision*, 2010. 2

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. *Computer Vision and Pattern Recognition*, 2007. 5

[28] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN Features off-the-shelf : an Astounding Baseline for Recognition. *Computer Vision and Pattern Recognition Workshops*, 2014. 3, 6, 7

[29] S. Shalev-shwartz and N. Srebro. Pegasos : Primal Estimated sub-GrAdient SOlver for SVM. *Mathematical programming*, 127.1:3-30, 2011. 4

[30] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. a. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics*, 30(6):154, 2011. 1, 2

[31] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. *European Conference on Computer Vision*, 2012. 3

[32] S. Singh, A. Gupta, and A. A. Efros. Unsupervised Discovery of Mid-Level Discriminative Patches. *European Conference on Computer Vision*, 2012. 2

[33] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. *International Conference on Computer Vision*, 2003. 2, 6

[34] V. Sydorov, M. Sakurada, and C. Lampert. Deep Fisher KernelsEnd to End Learning of the Fisher Kernel GMM Parameters. *Computer Vision and Pattern Recognition*, 2014. 3

[35] G. Tolias, Y. Avrithis, and H. Jegou. To Aggregate or Not to aggregate: Selective Match Kernels for Image Search. *IEEE International Conference on Computer Vision*, 2013. 3

[36] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):480–92, 2012. 3, 4