# Generalized Max Pooling

Naila Murray and Florent Perronnin
Computer Vision Group, Xerox Research Centre Europe

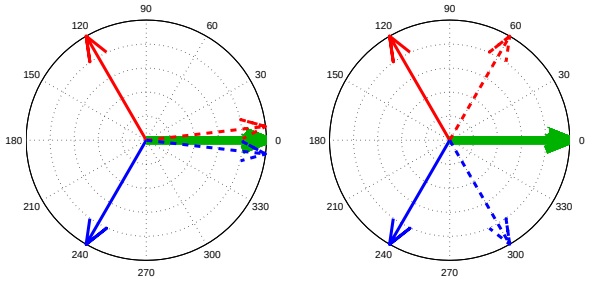
## Abstract

*State-of-the-art patch-based image representations involve a pooling operation that aggregates statistics computed from local descriptors. Standard pooling operations include sum- and max-pooling. Sum-pooling lacks discriminability because the resulting representation is strongly influenced by frequent yet often uninformative descriptors, but only weakly influenced by rare yet potentially highly-informative ones. Max-pooling equalizes the influence of frequent and rare descriptors but is only applicable to representations that rely on count statistics, such as the bag-of-visual-words (BOV) and its soft- and sparse-coding extensions. We propose a novel pooling mechanism that achieves the same effect as max-pooling but is applicable beyond the BOV and especially to the state-of-the-art Fisher Vector – hence the name Generalized Max Pooling (GMP). It involves equalizing the similarity between each patch and the pooled representation, which is shown to be equivalent to re-weighting the per-patch statistics. We show on five public image classification benchmarks that the proposed GMP can lead to significant performance gains with respect to heuristic alternatives.*

## 1. Introduction

This work is concerned with image classification. A state-of-the-art approach to representing an image from a collection of local descriptors consists of (i) encoding the descriptors using an embedding function $\varphi$ that maps the descriptors in a non-linear fashion into a higher-dimensional space; and (ii) aggregating the codes into a fixed-length vector using a pooling function. Successful representations that fall within this framework include the Bag-Of-Visual-words (BOV) [37, 11], the Fisher Vector (FV) [27], the VLAD [18], the Super Vector (SV) [42] and the Efficient Match Kernel (EMK) [4]. In this work, we focus on step (ii): the pooling step.

By far the most popular pooling mechanism involves summing (or averaging) the descriptor encodings [11, 27, 18, 42, 4]. An advantage of sum-pooling is its generality: it can be applied to any encoding. A major disadvantage however is that frequently-occurring descriptors will be more influential in the final representation than rarely-occurring ones (see Figure 1(a)). By "frequently-occurring descriptors" we mean descriptors which, though not necessarily identical, together form a mode in descriptor space. However, such frequent descriptors are not necessarily the most informative ones. Let us take the example of a fine-grained classification task where the goal is to distinguish bird species. In a typical bird image, most patches might correspond to background foliage or sky and therefore carry little information about the bird class. On the other hand, the most discriminative information might be highly localized and therefore correspond to only a handful of patches. Hence, it is crucial to ensure that even those rare patches contribute significantly to the final representation.




(a) Sum-pooling (b) Our GMP approach

Figure 1. We show the effect of pooling a single descriptor encoding (→ or →) with a set of tightly-clustered descriptor encodings (➔). Two pooled representations are shown: → + ➔ = ⇢ and → + ➔ = ⇢. With sum-pooling (a), the cluster of descriptors ➔ dominates the pooled representations ⇢ and ⇢, and as a result they are very similar to each other. With our GMP approach (b), both descriptors contribute meaningfully, resulting in highly distinguishable pooled representations.

Many approaches have been proposed in the computer vision literature to address the problem of frequent descriptors (see section 2). However, all of these solutions are heuristic in nature and/or limited to certain types of encodings. For instance, max-pooling [7] only makes sense in the context of the BOV or its soft-coding and sparse-coding extensions. However, it is not directly applicable to the FV, the VLAD, the SV or the EMK. This is because the

max-pooling operation treats each dimension independently while for such representations the encoding dimensions are strongly correlated and should be treated jointly.

In this work we propose a novel pooling mechanism that involves equalizing the similarity between each patch and the pooled representation. This can be viewed as a generalization of max-pooling to any encoding $\varphi$ – hence the name, *Generalized Max Pooling* (GMP). For instance, GMP is applicable to codebook-free representations such as the EMK and to higher-order representations such as the FV.

Our main contributions are the following ones. We first propose a matching criterion to compute the GMP representation (section 3). It is referred to as the primal formulation because it involves explicitly computing the final pooled representation. We then show that this criterion is equivalent to re-weighting the per-patch encodings before sum-pooling (section 4). We refer to this formulation as dual because the weights can be computed from the kernel of patch-to-patch similarities without the need to access the patch encodings. We show experimentally on five public benchmarks that the proposed GMP can provide significant performance gains with respect to heuristic alternatives such as power normalization (section 5).

## 2. Related Work

The problem of reducing the influence of frequent descriptors has received a great deal of attention in computer vision. This issue can be addressed at the pooling stage or *a posteriori* by performing some normalization on the image-level pooled descriptor. We therefore review related works on local descriptor pooling and image-level descriptor normalization.

**Local descriptor pooling.** Pooling is the operation which involves aggregating several local descriptor encodings into a single representation. On the one hand, pooling achieves some invariance to perturbations of the descriptors. On the other hand, it may lead to a loss of information. To reduce as much as possible this loss, only close descriptors should be pooled together [7, 5], where the notion of closeness can be understood in the geometric and/or descriptor domains. To enforce the pooling of close descriptors in the geometric space, it is possible to use spatial pyramids [21]. In the descriptor space, the closeness constraint is achieved through the choice of an appropriate embedding $\varphi$. Note that our focus in this work is not on the choice of $\varphi$ but on the pooling operation, which we now discuss.

Pooling is typically achieved by either summing/averaging or by taking the maximum response. Sum-pooling has been used in many biologically-inspired visual recognition systems to approximate the operation of receptive fields in early stages of the visual cortex [15, 31, 20]. It is also a standard component in convolutional neural networks [22]. A major disadvantage of sum-pooling is that it is based on the incorrect assumption that the descriptors in an image are independent and that their contributions can be summed [7, 9, 35].

Max-pooling was advocated by Riesenhuber and Poggio as a more appropriate pooling mechanism for higher-level visual processing such as object recognition [33]. It has subsequently been used in computer vision models of object recognition [36] and especially in neural networks [32, 23]. It has also recently found success in image classification tasks when used in conjunction with sparse coding techniques [41, 6, 7]. A major disadvantage of max-pooling is that it only makes sense when applied to embedding functions $\varphi$ which encode a strength of association between a descriptor and a codeword, as is the case of the BOV and its soft- and sparse-coding extensions. However, it is not directly applicable to those representations which compute higher-order statistics such as the FV, which has been shown to yield state-of-the-art results in classification [8] and retrieval [18].

Several extensions to the standard sum- and max-pooling frameworks have been proposed. For instance, one can transition smoothly from sum- to max-pooling using $\ell_p$- or softmax-pooling [7]. It is also possible to add weights to obtain a weighted pooling [12]. While our GMP can be viewed as an instance of weighted pooling (see section 4), a major difference with [12] is that their weights are computed using external information to cancel-out the influence of irrelevant descriptors while our weights are computed from the descriptors themselves and equalize the influence of frequent and rare descriptors.

**Image-level descriptor normalization.** Many works make use of sum-pooling and correct *a posteriori* for the incorrect independence assumption through normalization of the pooled representation. Jégou *et al.* proposed several re-weighting strategies for addressing visual burstiness in the context of image retrieval by matching [17]. These include penalizing multiple matches between a query descriptor and a database image and penalizing the matches of descriptors which are matched to multiple database images (*i.e*. IDF weighting applied at the descriptor level rather than the visual word level). Torii *et al.* proposed another re-weighting scheme for BOV-based representations, which soft-assigns to fewer visual words those descriptors which are extracted from a repetitive image structure [38]. Arandjelović *et al.* showed that, for the VLAD representation [18], applying $\ell_2$-normalization to the aggregated representation of each pooling region mitigates the burstiness effect [2]. Delhumeau *et al*. found that, for VLAD, $\ell_2$-normalizing the descriptor residuals and then applying PCA before pooling was beneficial [13]. Power normalization has also been shown to be an effective heuristic for treating frequent descriptors in BOV, FV or VLAD representations [29, 30, 18]. The main drawback of the previous works is that they are

heuristic and/or restricted to image representations based on a finite vocabulary. In the latter case, they are not applicable to codebook-free representations such as the EMK.

One of the rare works which considered the independence problem in a principled manner is that of Cinbis *et al*. which proposes a latent model to take into account inter-descriptor dependencies [9]. However, this work is specific to representations based on Gaussian Mixture Models. In contrast, our GMP is generic and applicable to all aggregation-based representations.

## 3. GMP as equalization of similarities

Let $X = \{x_1, \ldots, x_N\}$ be a set of $N$ patches extracted from an image and let $\varphi_n = \varphi(x_n)$ denote the $D$-dimensional encoding of the $n$-th patch.

Our goal is to propose a pooling mechanism that mimics the desirable properties of max-pooling in the BOV case and is extensible beyond the BOV. One such property is the fact that the dot-product similarity between the max-pooled representation $\varphi^{max}$ and a single patch encoding $\varphi_n$ is a constant value[1]. To see this, let $C$ denote the codebook cardinality ($C = D$ in the BOV case) and let $i_n$ be the index of the closest codeword to patch $x_n$. $\varphi_n$ is a binary vector with a single non-zero entry at index $i_n$. $\varphi^{max}$ is a binary representation where a 1 is indicative of the presence of the codeword in the image. Consequently, we have $\varphi_n^T \varphi^{max} = 1$ for all $\varphi_n$ and $\varphi^{max}$ is equally similar to frequent and rare patches. This occurs because frequent and rare patches contribute equally to the aggregated representation,

In contrast, the sum-pooled representation is overly influenced by frequent descriptors. Indeed, in the case of the unnormalized sum-pooled representation, we have $\varphi^{sum} = [\pi_1, \ldots, \pi_C]^T$ where $\pi_k$ is the number of occurrences of codeword $k$ and therefore $\varphi_n^T \varphi^{sum} = \pi_{i_n}$. Consequently, more frequent patches make a greater contribution to the aggregated representation. As a result, $\varphi^{sum}$ is more similar to frequent descriptors than to rare ones.

Figure 1 illustrates this property for our GMP approach: $\varphi^{sum}$ is more similar to the more frequent descriptor (Figure 1(a)) whereas $\varphi^{gmp}$ is equally similar to both the frequent and rare descriptor (Figure 1(b)), although the descriptors are not single-entry binary vectors as in the BOV case. We now describe how GMP generalizes this property to arbitrary descriptors (section 3.1). We also explain how to compute efficiently the GMP representation in practice (section 3.2).

[1]As mentioned in the introduction, we consider in this work the image classification problem. For efficiency purposes, we focus on the case where the pooled representations are classified using linear kernel machines, for instance linear SVMs. Therefore, the implicit metric which is used to measure the similarity between patch encodings is the dot-product.

### 3.1. Primal Formulation

Let $\varphi^{gmp}$ denote the GMP representation. We generalize the previous matching property and enforce the dot-product similarity between each patch encoding $\varphi_n$ and the GMP representation $\varphi^{gmp}$ to be a constant $c$:

$$\varphi_n^T \varphi^{gmp} = c, \text{ for } n = 1, \ldots, N. \tag{1}$$

Note that the value of the constant has no influence as we typically $\ell_2$-normalize the final representation. Therefore, we arbitrarily set this constant to $c = 1$. Let $\Phi$ denote the $D \times N$ matrix that contains the patch encodings: $\Phi = [\varphi_1, ..., \varphi_N]$. In matrix form, (1) can be rewritten as:

$$\Phi^T \varphi^{gmp} = \mathbb{1}_N, \tag{2}$$

where $\mathbb{1}_N$ denotes the $N$-dimensional vector of all ones. This is a linear system of $N$ equations with $D$ unknowns. In general, this system might not have a solution (*e.g.* when $D < N$) or might have an infinite number of solutions (*e.g.* when $D > N$). Therefore, we turn (2) into a least-squares regression problem and seek

$$\varphi^{gmp} = \arg\min_{\varphi} ||\Phi^T \varphi - \mathbb{1}_N||^2, \tag{3}$$

with the additional constraint that $\varphi^{gmp}$ has minimal norm in the case of an infinite number of solutions. The previous problem admits a simple closed-form solution:

$$\varphi^{gmp} = (\Phi^T)^+ \mathbb{1}_N = (\Phi\Phi^T)^+ \Phi \mathbb{1}_N, \tag{4}$$

where $^+$ denotes the pseudo-inverse and the second equality stems from the property $A^+ = (A^T A)^+ A^T$. We note that $\Phi\mathbb{1}_N = \sum_{n=1}^N \varphi(x_n) = \varphi^{sum}$ is the sum-pooled representation. Hence, the proposed GMP involves projecting the $\varphi^{sum}$ on $(\Phi\Phi^T)^+$. Note that this is different from recent works in computer vision advocating for the decorrelation of the data [16] since in our case the uncentered correlation matrix $\Phi\Phi^T$ is computed *from patches of the same image*.

It can be easily shown that, in the BOV case (hard coding), the GMP representation is strictly equivalent to the max-pooling representation. This is not a surprise given that GMP was designed to mimic the good properties of max-pooling. We show in the appendix a more general property.

### 3.2. Computing the GMP in practice

Since the pseudo-inverse is not a continuous operation it is beneficial to add a regularization term to obtain a stable solution. We introduce $\varphi_\lambda^{gmp}$, the regularized GMP:

$$\varphi_\lambda^{gmp} = \arg\min_{\varphi} ||\Phi^T \varphi - \mathbb{1}_N||^2 + \lambda||\varphi||^2. \tag{5}$$

This is a ridge regression problem whose solution is

$$\varphi_\lambda^{gmp} = (\Phi\Phi^T + \lambda I)^{-1} \Phi \mathbb{1}_N. \tag{6}$$

The regularization parameter $\lambda$ should be cross-validated. For $\lambda$ very large, we have $\varphi_\lambda^{gmp} \approx \Phi \mathbb{1}_N/\lambda$ and we are back to sum pooling. Therefore, $\lambda$ does not only play a regularization role. It also enables one to smoothly transition between the solution to (4) ($\lambda = 0$) and sum pooling ($\lambda \to \infty$).

We now turn to the problem of computing $\varphi_\lambda^{gmp}$. We can compute (6) using Conjugate Gradient Descent (CGD) which is designed for PSD matrices. This might be computationally intensive if the encoding dimensionality $D$ is large and the matrix $\Phi$ is full (cost in $O(D^2)$).

However, we can exploit the structure of certain patch encodings. Especially, the computation can be sped-up if the individual patch encodings $\varphi_n$ are block-sparse. By block-sparse we mean that the indices of the encoding can be partitioned into a set of groups where the activation of one entry in a group means the activation of all entries in the group. This is the case for instance of the VLAD and the SV where each group of indices corresponds to a given cluster centroid. This is also the case of the FV if we assume a hard assignment model where each group corresponds to the gradients with respect to the parameters of a given Gaussian. In such a case, the matrix $\Phi\Phi^T$ is block-diagonal. Consequently $\Phi\Phi^T + \lambda I$ is block diagonal and (6) can be solved block-by-block, which is significantly less demanding than solving the full problem directly (cost in $O(D^2/C)$).

# 4. GMP as weighted pooling

In what follows, we first provide a dual interpretation of GMP as a weighted pooling (section 4.1). We then visualize the computed weights (section 4.2).

## 4.1. Dual Formulation

We note that the regularized GMP $\varphi_\lambda^{gmp}$ is the solution to (5) and that, consequently, according to the representer theorem, $\varphi_\lambda^{gmp}$ can be written as a linear combination of the encodings:

$$\varphi_\lambda^{gmp} = \Phi \alpha_\lambda \tag{7}$$

where $\alpha_\lambda$ is the vector of weights. Therefore GMP can be viewed as an instance of weighted pooling [12]. By introducing $\varphi = \Phi\alpha$ in the GMP objective (5), we obtain:

$$\alpha_\lambda = \arg\min_\alpha ||\Phi^T\Phi\alpha - \mathbb{1}_N||^2 + \lambda||\Phi\alpha||^2. \tag{8}$$

If we denote by $K = \Phi^T\Phi$ the $N \times N$ kernel matrix of patch-to-patch similarities, we finally obtain:

$$\alpha_\lambda = \arg\min_\alpha ||K\alpha - \mathbb{1}_N||^2 + \lambda\alpha^T K\alpha \tag{9}$$

which admits the following simple solution:

$$\alpha_\lambda = (K + \lambda I_N)^{-1}\mathbb{1}_N \tag{10}$$

which only depends on the patch-to-patch similarity kernel, not on the encodings.

We note that, in the general case, computing the kernel matrix $K$ and solving (9) have a cost in $O(N^2D)$ and $O(N^2)$ respectively. This might be prohibitive for large values of $N$ and $D$. However, as was the case for the primal formulation, we can exploit the structure of certain encodings. This is the case of the VLAD or the FV (with hard assignment): since the encoding is block-sparse, the matrix $K$ is block-diagonal. Using an inverted file type of structure, one can reduce the cost of computing $K$ to $O(N^2D/C^2)$ by matching only the encodings $\varphi_n$ that correspond to patches assigned to the same codeword[2]. Also, one can solve for $\alpha_\lambda$ block-by-block which reduces the cost to $O(N^2/C)$.

Once weights have been computed, the GMP representation is obtained by linearly re-weighting the per-patch encodings – see equation (7). Note that in all our experiments we use the primal formulation to compute $\varphi_\lambda^{gmp}$, which is more efficient than first computing the weights and then re-weighting the encodings. However, we will see in the following section that the dual formulation is useful because it enables visualizing the effect of the GMP.

## 4.2. Visualizing weights

We use weights computed via the dual formulation to generate a topographic map whose value at pixel location $l$ is computed as

$$s_l = \sum_{x_i \in \mathcal{P}_l} \alpha_i, \tag{11}$$

where $\mathcal{P}_l$ is the set of patches $x_i$ which contain location $l$ and $\alpha_i$ is the weight of patch $x_i$. These maps give us a quantitative measure of the rarity, and potential discriminativeness, of the different regions in the image. Figure 2 shows several such maps for bird images. Clearly, they are reminiscent of saliency maps computed to predict fixations of the human gaze. One sees that the highly-weighted regions contain rare image patches, such as the breast of the bird in the third row. When the background is quite simple, as is the case of the first row, the learned weights tend to segment the foreground from the background. Note however that our maps are computed in a fully *unsupervised* manner and that there is no foreground/background segmentation guarantee in the general case.

# 5. Experimental Evaluation

We first describe the image classification datasets and image descriptors we use. We then report results.

[2]The $O(N^2D/C^2)$ complexity is based on the optimistic assumption that the same number of patches $N/C$ is assigned to each codeword.

Figure 2. Maps generated using weights computed from color descriptors using the EMK encoding, for a sample of images from the CUB-2011 dataset. Left: original images. Middle: weight maps. Right: weighted images. See sections 5.1 and 5.2 for details of the dataset and descriptors.

### 5.1. Datasets

As mentioned earlier, we expect the proposed GMP to be particularly beneficial on fine-grained tasks where the most discriminative information might be associated with a handful of patches. Therefore, we validated the proposed approach on four fine-grained image classification datasets: CUB-2010, CUB-2011, Oxford Pets and Oxford Flowers. We also include the PASCAL VOC 2007 dataset in our experiments since it is one of the most widely used benchmarks in the image classification literature. On all datasets, we use the standard training/validation/test protocols.

The **Pascal VOC 2007** (VOC-2007) dataset [14] contains 9,963 images of 20 classes. Performance on this dataset is measured with mean average precision (mAP). A recent benchmark of encoding methods [8] reported 61.7% using the FV descriptor with spatial pyramids [30].

The **CalTech UCSD birds 2010** (CUB-2010) dataset [40] contains 6,033 images of 200 bird categories. Performance is measured with top-1 accuracy. The best reported performance we are aware of for CUB-2010 is 17.5% [1]. This method uses sparse coding in combination with object detection and segmentation prior to classification. Without detection and segmentation, performance drops to 14.4% [1].

The **CalTech UCSD birds 2011** (CUB-2011) dataset [39] is an extension of CUB-2010 that contains 11,788 images of the same 200 bird categories. Performance is measured with top-1 accuracy. The best reported performance for CUB-2011 is, to our knowledge, 56.8%. This was obtained using ground-truth bounding boxes and part detection [3]. The best reported performance we are aware of that does not use ground-truth annotations or localization is 28.2% [34].

The **Oxford-IIIT-Pet** (Pets) dataset [26] contains 7,349 images of 37 categories of cats and dogs. Performance is measured with top-1 accuracy. The best reported performance for Pets is 54.3%, which was also obtained using the method of [1]. Without detection and segmentation, performance drops to 50.8% [1].

The **Oxford 102 Flowers** (Flowers) dataset [25] contains 8,189 images of 102 flower categories. Performance is measured with top-1 accuracy. The best reported performance for Flowers is 80.7%, and was also obtained using the method of [1]. Again, without detection and segmentation performance drops to 76.7% [1].

### 5.2. Descriptors

In our experiments, patches are extracted densely at multiple scales resulting in approximately 10K descriptors per image. We experimented with two types of low-level descriptors: 128-dim SIFT descriptors [24] and 96-dim color descriptors [10]. In both cases, we reduced their dimensionality to 64 dimensions with PCA. Late fusion results were obtained by evaluating an unweighted summation of the scores given by the SIFT and color-based classifiers.

As mentioned earlier, the proposed GMP is general and can be applied to any aggregated representation. Having shown in the appendix, and verified experimentally, the formal equivalence between GMP and standard max-pooling in the BOV hard-coding case, we do not report any result for the BOV. In our experiments, we focus on two aggregated representations: the EMK [4] and the FV [27].

### 5.3. Results with the EMK

To compute the EMK representations we follow [4]: we project the descriptors on random Gaussian directions, apply a cosine non-linearity and aggregate the responses. The EMK is a vocabulary-free approach which does not perform any quantization and as a result preserves minute and highly-localized image details. The EMK is thus especially relevant for fine-grained problems. However, since all embeddings are pooled together rather than within Voronoi regions as with vocabulary-based approaches, the EMK is particularly susceptible to the effect of frequent descriptors. Therefore we expect GMP to have a significant positive impact on the EMK performance. To the best of our knowledge, there is no previous transformation that may be applied to the EMK to counteract frequent descriptors. In particular, power normalization heuristics which are used for vocabulary-based approaches such as the BOV [29] or the FV [30] are not suitable.

The EMK representation has two parameters: the number of output dimensions $D$ (*i.e.* the number of random projections) and the bandwidth $\sigma$ of the Gaussian kernel from which the random directions are drawn. The dimension $D$ was set to 2,048 for all experiments as there was negligible

improvement in performance for larger values. $\sigma$ was chosen through cross-validation. The choice of $\lambda$ (the regularization parameter of the GMP) has a significant impact on the final performance and was chosen by cross-validation from the set $\{10^1, 10^2, 10^3, 10^4, 10^5\}$. We do not use spatial pyramids.

Results with the EMK are shown in Table 1. We report results for the baseline EMK (sum-pooling *i.e.* no mitigation of frequent descriptors) and the EMK with the proposed GMP. A significant improvement in performance, between 3% and 20%, is achieved for all datasets when using GMP. This indicates that suppressing frequent descriptors is indeed beneficial when using EMKs. On the fine-grained datasets, the improvements are particularly impressive – 16% on average.

### 5.4. Results with the FV

We now turn to the state-of-the-art FV representation [27, 30]. To construct the FV we compute for each descriptor the gradient of the log-likelihood with respect to the parameters of a Gaussian Mixture Model (GMM) and pool the gradients. For the FV, increasing the number of Gaussians $G$ counteracts the negative effects of frequent descriptors as fewer and fewer descriptors are assigned to the same Gaussian. Therefore we expect GMP to have a smaller impact than for the EMK, particularly as $G$ increases. By default we do not use spatial pyramids, but have included a discussion on its effect for the VOC-2007 dataset.

Experiments were conducted for FVs with $G$ set to either 16 or 256, leading to 2,048-dim and 32,768-dim vectors respectively. Values of $G$ of 16 and 256 were chosen in order to have a comparable dimensionality to that of the EMK representation in the former case, and to have a state-of-the-art FV representation in the latter case. The value of the GMP regularization parameter $\lambda$ was once again chosen by cross-validation from the set $\{10^1, 10^2, 10^3, 10^4, 10^5\}$.

**Power normalization baseline.** Our baseline method uses power normalization, the state-of-the-art and post-hoc approach for improving the pooled FV representation [30]. In the literature, the power is usually set to 0.5 [30, 8, 18, 35]. Indeed, we found this value to be optimal for VOC-2007 for SIFT descriptors. However it has been shown, in the context of image retrieval, that a lower value often can achieve significant performance gains [28]. We observed the same effect for classification. Therefore, we cross-validated the value of the power parameter. We tested the following set of 8 values: $\{1.0, 0.7, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0\}$. Note that for a value of 0, we follow [28] and apply the power normalization only to non-zero entries. Results with the best-performing power (*i.e.* the value that led to the best results on the validation set) are denoted by sum+p in Table 2. The optimal power was determined on a per-descriptor and per-dataset basis. Hence, *our power baseline is a very strong one*. For instance, for CUB-2011, performance with late fusion and $G = 256$ increases from 25.4% with the default $0.5$ value to 29.7% with the cross-validated power normalization.

| Descriptor | VOC-2007 | | CUB-2010 | | CUB-2011 | | Pets | | Flowers | |
|---|---|---|---|---|---|---|---|---|---|---|
| | sum | GMP | sum | GMP | sum | GMP | sum | GMP | sum | GMP |
| SIFT | 40.0 | **46.0** | 2.9 | **6.5** | 5.0 | **10.6** | 21.7 | **35.6** | 41.3 | **52.2** |
| Color | 30.1 | **34.6** | 3.0 | **11.8** | 4.0 | **22.0** | 13.2 | **28.5** | 41.8 | **60.2** |
| Fusion | 43.1 | **49.5** | 3.7 | **13.5** | 6.0 | **24.8** | 22.8 | **42.9** | 55.8 | **69.5** |

Table 1. EMK results for SIFT descriptors, color descriptors, and late fusion of SIFT + color. Results are shown for $D = 2,048$-dim features for sum-pooling (sum) and for our GMP approach.

**GMP *vs*. no power normalization.** Results are shown in Table 2. Our GMP consistently performs significantly better – 10% better on average for late fusion and $G = 256$ – than when no power normalization is applied. The improvement is particularly impressive for several fine-grained datasets. For instance, for CUB-2011 GMP obtains a top-1 accuracy of 30.8% compared to 13.2% with sum-pooling.

**GMP *vs*. power normalization.** GMP always outperforms power normalization for all datasets for $G = 16$. The average improvement for late fusion is 2.8%. As expected, as $G$ increases to 256 GMP has less of an impact, but still outperforms power normalization by 0.4% on average, with late fusion. Note that, on the Flowers dataset with late fusion and $G = 256$, we obtain 83.5% and 82.2% respectively for the power normalization and GMP. These outperform the previous state-of-the-art (80.7% [1]). Also, on the Pets dataset with late fusion and $G = 256$, GMP obtains top-1 accuracy of 56.1%, compared to 54.2% with power normalization – an increase in performance of 1.9%. This is to our knowledge the best-reported result for this dataset, out-performing the previous state-of-the-art (54.3% [1]). Therefore GMP achieves or exceeds the performance of the ad-hoc power normalization technique, while being more principled and more general. Note that GMP may also be combined with power normalization (GMP+p in Table 2). This combination results in average improvements over power normalization of 3.8% for $G = 16$ and 2.5% for $G = 256$, showing that GMP and power normalization are somewhat complementary.

**Effect of spatial pyramids.** We ran additional experiments on VOC-2007 to investigate the effect of our method when using Spatial Pyramids (SPs). We used a coarse pyramid and extracted 4 FVs per image: one FV for the whole image and one FV each for three horizontal strips corresponding to the top, middle and bottom regions of the image. With SPs, GMP again afforded improvements with respect to power normalization. For instance, with late fusion and $G = 256$, GMP obtains 62.0% compared to 60.2% for the power baseline – a 1.8% increase in performance.

| Descriptor | | VOC-2007 | | | | CUB-2010 | | | | CUB-2011 | | | | Pets | | | | Flowers | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | sum | sum+p | GMP | GMP+p | sum | sum+p | GMP | GMP+p | sum | sum+p | GMP | GMP+p | sum | sum+p | GMP | GMP+p | sum | sum+p | GMP | GMP+p |
| $G$=16 | SIFT | 49.1 | 51.6 | 52.8 | **53.4** | 4.1 | 6.2 | 6.4 | **6.9** | 7.9 | 11.1 | 11.5 | **12.7** | 29.4 | 32.1 | 35.1 | **35.7** | 58.3 | 62.8 | 63.8 | **65.3** |
| | Color | 40.2 | 43.8 | 45.3 | **45.8** | 4.9 | 8.7 | 12.5 | **13.1** | 7.2 | 16.8 | 21.6 | **22.8** | 22.5 | 28.6 | 32.5 | **33.5** | 55.3 | 65.6 | 65.9 | **67.2** |
| | Fusion | 52.2 | 55.1 | 57.0 | **57.1** | 5.6 | 10.1 | 13.0 | **13.9** | 10.0 | 18.0 | 23.4 | **25.6** | 33.5 | 40.5 | 42.9 | **44.4** | 69.9 | 77.6 | 79.0 | **79.3** |
| $G$=256 | SIFT | 52.6 | 57.8 | 58.1 | **58.9** | 5.3 | 8.1 | 7.7 | **9.6** | 10.2 | 16.3 | 16.4 | **17.0** | 38.1 | 47.1 | 47.9 | **49.2** | 67.7 | 73.0 | 72.8 | **73.3** |
| | Color | 39.4 | 49.5 | 50.0 | **50.4** | 4.6 | 13.0 | 14.2 | **14.6** | 9.0 | 27.4 | 27.0 | **29.3** | 23.6 | 41.1 | 41.6 | **43.0** | 63.9 | 74.0 | 72.8 | **75.1** |
| | Fusion | 54.7 | 60.6 | **61.8** | 61.7 | 7.1 | 14.2 | 13.3 | **17.2** | 13.2 | 29.7 | 30.8 | **33.3** | 40.5 | 54.2 | 56.1 | **56.8** | 77.2 | 83.5 | 82.2 | **84.6** |

Table 2. FV results for SIFT descriptors, color descriptors, and late fusion of SIFT + color. Results are shown for $G = 16$ and $G = 256$ Gaussians for sum-pooling (sum), sum-pooling + power normalization (sum+p), our GMP approach (GMP), and GMP + power normalization (GMP+p).

**Effect of the number of Gaussians $G$.** As expected, there is a consistent and significant positive impact on performance when $G$ is increased from 16 to 256. Our GMP approach is complementary to increasing $G$, as performance is generally improved when more Gaussians are used and GMP is applied. Furthermore, GMP is particularly attractive when low-dimensional FVs must be used.

**FV *vs*. EMK.** The baseline EMK results are quite poor in comparison with the baseline FV results. However, for CUB-2010, CUB-2011, and Pets, GMP improves the EMK performance to the point that EMK results with GMP are comparable to FV results with GMP when $G = 16$ (with $G = 16$, the FV and EMK representations are both 2,048-dimensional). In fact, for CUB-2011, EMK with GMP is superior to FV with GMP for $G = 16$ (24.8% *vs*. 23.4%).

## 6. Conclusions

We have proposed a principled and general method for pooling patch-level descriptors which equalizes the influence of frequent and rare descriptors, preserving discriminating information in the resulting aggregated representation. Our generalized max-pooling (GMP) is applicable to any encoding technique and can thus be seen as an extension of max pooling, which can only be applied to count-based representations such as BOV and its soft-coding and hard-coding extensions. Extensive experiments on several public datasets show that GMP performs on par with, and sometimes significantly better than, heuristic alternatives.

We note that, in the same proceedings, Jégou and Zisserman [19] propose a democratic aggregation which bears some similarity to our GMP. Especially, it involves re-weighting the patch encodings to balance the influence of descriptors. One potential benefit of the GMP is that it is can be efficiently computed in the primal while the computation of the democratic aggregation can only be performed in the dual. However, it remains to be seen how these two aggregation mechanisms compare in practice.

## A. GMP and Max-Pooling

In this appendix, we relate the GMP to max-pooling. We denote by $\mathcal{E} = \{\varphi_1, \ldots, \varphi_N\}$ the set of descriptor encodings of a given image. We assume that these encodings are drawn from a finite codebook of possible encodings, *i.e.* $\varphi(x_i) \in \{q_1, \ldots, q_C\}$. Note that the codewords $q_k$ might be binary or real-valued. We denote by $Q$ the $D \times C$ codebook matrix of possible embeddings where we recall that $D$ is the output encoding dimensionality. We assume $Q = [q_1, \ldots, q_C]$ is orthonormal, *i.e.* $Q^T Q = I_C$ where $I_C$ is the $C \times C$ identity matrix. For instance, in the case of the BOV with hard-coding, $D = C$ and the $q_k$'s are binary with only the $k$-th entry equal to 1, so that $Q = I_C$. We finally denote by $\pi_k$ the proportion of occurrences of $q_k$ in $\mathcal{E}$.

**Theorem.** $\varphi^{gmp}$ does not depend on the proportions $\pi_k$, but only on the presence or absence of the $q_k$'s in $\mathcal{E}$.

**Proof.** We denote by $\Pi$ the $C \times C$ diagonal matrix that contains the values $\pi_1$, ..., $\pi_C$ on the diagonal. We rewrite $\Phi \mathbb{1}_M = Q\Pi\mathbb{1}_C$ and $\Phi\Phi^T = Q\Pi Q^T$. The latter quantity is the SVD decomposition of $\Phi\Phi^T$ and therefore we have $(\Phi\Phi^T)^+ = Q\Pi^+ Q^T$. Hence (4) becomes $\varphi^{gmp} = Q\Pi^+ Q^T Q\Pi\mathbb{1}_C = Q(\Pi^+\Pi)\mathbb{1}_C$. Since $\Pi$ is diagonal, its pseudo-inverse is diagonal and the values on the diagonal are equal to $1/\pi_k$ if $\pi_k \neq 0$ and 0 if $\pi_k = 0$. Therefore, $\Pi^+\Pi$ is a diagonal matrix with element $k$ on the diagonal equal to 1 if $\pi_k \neq 0$ and 0 otherwise. Therefore we have

$$\varphi^{gmp} = \sum_{k:\pi_k \neq 0} q_k, \tag{12}$$

which does not depend on the proportions $\pi_k$, just on the presence or absence of the $q_k$'s in $\mathcal{E}$.

For the BOV hard-coding case, equation (12) shows that $\varphi^{gmp}$ is a binary representation where each dimension informs on the presence/absence of each codeword in the image. This is *exactly the max-pooled representation*. Therefore, our pooling mechanism can be understood as a *generalization of max-pooling*.

Note that there is no equivalence between the standard max-pooling and the GMP in the soft- or sparse-coding cases. One benefit of the GMP however is that it is independent of a rotation of the encodings. This is not the case of the standard max-pooling which operates on a per-dimension basis.

## Acknowledgments

The authors wish to warmly thank H. Jégou and T. Furon for fruitful discussions as well as P. Torr, A. Vedaldi and A. Zisserman for useful comments. This work was done in the context of the Project Fire-ID, supported by the Agence Nationale de la Recherche (ANR-12-CORD-0016).

## References

[1] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. *CVPR*, 2013.

[2] R. Arandjelovic and A. Zisserman. All about VLAD. *CVPR*, 2013.

[3] T. Berg and P. N. Belhumeur. POOF: Part-Based One-vs-One Features for fine-grained categorization, face verification, and attribute estimation. *CVPR*, 2013.

[4] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. *NIPS*, 2009.

[5] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: multi-way local pooling for image recognition. *ICCV*, 2011.

[6] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. *CVPR*, 2010.

[7] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. *ICML*, 2010.

[8] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. *BMVC*, 2011.

[9] R. G. Cinbis, J. Verbeek, and C. Schmid. Image categorization using Fisher kernels of non-iid image models. *CVPR*, 2012.

[10] S. Clinchant, G. Csurka, F. Perronnin, and J.-M. Renders. XRCEs participation to imageval. *ImageEval @ CVIR*, 2007.

[11] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *ECCV SLCV workshop*, 2004.

[12] T. de Campos, G. Csurka, and F. Perronnin. Images as sets of locally weighted features. *CVIU*, 2012.

[13] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the vlad image representation. *ACM MM*, 2013.

[14] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 2010.

[15] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern recognition*, 1982.

[16] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. *ECCV*, 2012.

[17] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. *CVPR*, 2009.

[18] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 2012.

[19] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. *CVPR*, 2014.

[20] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. *CVPR*, 2012.

[21] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.

[22] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Handwritten digit recognition with a back-propagation network. *NIPS*, 1989.

[23] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *ICML*, 2009.

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[25] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. *ICCVGIP*, 2008.

[26] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. *CVPR*, 2012.

[27] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. *CVPR*, 2007.

[28] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. *CVPR*, 2010.

[29] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. *CVPR*, 2010.

[30] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. *ECCV*, 2010.

[31] N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 2008.

[32] M. Ranzato, Y.-l. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. *NIPS*, 2007.

[33] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 1999.

[34] J. A. Rodriguez and D. Larlus. Predicting an object location using a global image representation. *ICCV*, 2013.

[35] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013.

[36] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. *CVPR*, 2005.

[37] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. *ICCV*, 2003.

[38] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. *CVPR*, 2013.

[39] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, CalTech, 2011.

[40] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, CalTech, 2010.

[41] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *CVPR*, 2009.

[42] Z. Zhou, K. Yu, T. Zhang, and T. Huang. Image classification using super-vector coding of local image descriptors. *ECCV*, 2010.