Inria+Xerox@FGcomp: Boosting the Fisher vector for fine-grained classification

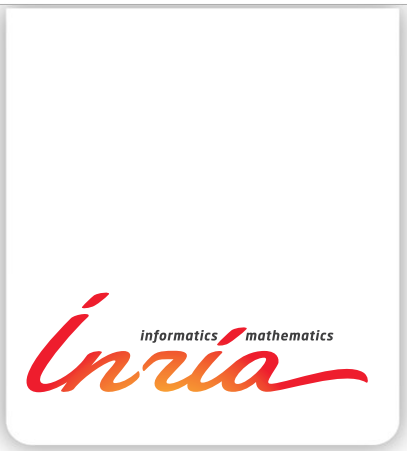
Philippe-Henri Gosselin, Naila Murray, Hervé Jégou, Florent Perronnin

Inria+Xerox@FGcomp: Boosting the Fisher vector for fine-grained classification

Philippe-Henri Gosselin,
Naila Murray,
Hervé Jégou,
Florent Perronnin

RESEARCH REPORT
N° 8431
December 2013
Project-Team Texmex

ISSN 0249-6399 ISRN INRIA/RR--8431--FR+ENG



Inria+Xerox@FGcomp: Boosting the Fisher vector for fine-grained classification

Philippe-Henri Gosselin*†,
Naila Murray‡,
Hervé Jégou†,
Florent Perronnin‡

Project-Team Texmex

Research Report n° 8431 — December 2013 — 16 pages

Abstract: This report describes the joint submission of Inria and Xerox to their participation to the FGCOMP 2013 challenge. Although the proposed system follows most of the standard Fisher classification pipeline, we describe several key features and good practices that improve the accuracy when specifically considering fine-grained classification tasks. In particular, we consider the late fusion of two systems both based on Fisher vectors, but that employ drastically different design choices that make them very complementary. Moreover, we show that a simple yet effective filtering strategy significantly boosts the performance for several class domains.

Key-words: image classification, fine-grained classification, FGCOMP, evaluation campaign

This work was done in the context of the Project Fire-ID, supported by the Agence Nationale de la Recherche (ANR-12-CORD-0016). This technical report is a pre-print of a letter submitted to Pattern Recognition Letters.

* ENSEA
† Inria
‡ Xerox Research Center Europe

RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Inria+Xerox@FGcomp: Classification à grain fin avec le vecteur de Fisher

Résumé : Ce rapport décrit la soumission jointe d'Inria et Xerox à la campagne d'évaluation FGCOMP 2013. Bien que notre système soit fondé sur une chaîne de traitement classique utilisant des vecteurs de Fisher, nous décrivons des éléments essentiels et des bonnes pratiques qui améliorent significativement la pertinence dans le contexte spécifique de la classification d'image à grain fin. En particulier, nous montrons que la fusion tardive de deux systèmes utilisant tous deux des vecteurs de Fisher, mais qui diffèrent radicalement dans certains choix de design, est très complémentaire. De plus, nous montrons qu'une simple stratégie de filtrage des descripteurs locaux améliore considérablement la pertinence pour plusieurs domaines de reconnaissance.

Mots-clés : classification d'image, classification à grain fin, FGCOMP, campagne d'évaluation

Contents

1 Introduction

Given an input image, image classification aims at determining what is the category of the objects depicted in the image. For instance, typical visual classes are 'person', 'bird', 'cat', 'aircraft', 'chair', etc. Recently, we have witnessed a shift in the interest of the computer vision community towards *Fine-grained classification* (FGC). Although a precise definition of the problem is not formally given, the objective is here to determine the class at a finer level of granularity. For instance, assuming that we know that all considered images contain birds, FGC requests the system to decide what kind of bird is depicted. Another example is to indicate what model of car is present in the image, as opposed to classification that would simply ask to determine whether a car appears in the image.

Compared to traditional image classification, FGC is assumed more challenging because many classes are very similar and prone to confusion: the difference between two classes of cars or birds may be visually distinguishable only based on tiny details that are difficult to learn automatically.

Another point is that, due to significant differences with standard image classification, it is still unclear which approaches perform best, and how they should be adapted to better address the specific aspects underlying fine-grained recognition. The best performing approaches for image classification are expected to give good performance in FGC: The Fisher vector [18, 19, 4], deformable part models [8] and deep learning approaches [15].

The goal of this paper is to evaluate the suitability of the Fisher vector in this context. Starting from a system similar to the Fisher pipeline used in regular image classification, we derive a first subsystem (SA) that closely follows the one used in previous image classification evaluation campaigns. We then show how tuning certain important parameters gives a significant boost in performance. Then we consider a second subsystem (SB) also based on Fisher, but designed 1) to be as complementary as possible with subsystem SA and 2) to incorporate some strategies that are specifically adapted to particular domains, such as cars and aircraft. These systems have been designed and optimized in the context of the joint participation of Inria and

Table 1: FGCOMP 2013 challenge: statistics on number of classes, minimum/maximum and average numbers of labels per class.

		train examples				test
domain	classes	min	avg	max	total	size
aircraft	100	66	66	67	6667	3333
bird	83	50	50	50	4150	4105
car	196	24	41	68	8144	8041
dog	120	198	221	302	26580	12000
shoes	70	23	50	195	3511	1002

Xerox in the FGCOMP 2013 fine-grained challenge. For this purpose, we split the training set into two sets (75% for learning and 25% for validation).

This paper is organized as follows. Section 2 describes the fine-grained challenge and its evaluation protocol. Section 3 describes the vanilla Fisher classification pipeline, as used in particular in previous evaluation campaigns such as Pascal VOC'07 [7] and Imagenet [6]. Section 4 describes how we have adapted this method to the context of fine-grained classification, and gives a few good practices that may help potential participants of further campaigns. Section 5 analyzes the official results obtained in the FGCOMP 2013 challenge, where our joint participation has obtained the best performance among all participants.

2 Description of FGCOMP evaluation campaign

The FGComp challenge aims at evaluating current fine-grained classification systems when targeting a specific domain. In this context, the system has to predict a class in an image given its domain, and thus there is no need to determine with which domain an image is associated. The domains considered in the 2013 campaign were: aircrafts, birds, cars, dogs and shoes. In each case the class to predict is the brand or model of an object, or the species of an animal.

Table 1 summarizes the statistics per domain. For each domain, the number of classes is between 70 and 196, which is relatively large compared to most classification datasets. The average number of annotated samples per class is about 50 labels for most domains except dogs (221 examples in training set). This is significantly smaller than the number of labels one usually finds in the context of image categorization, where thousand of labels are available for each class, typically.

The organizers defined two tracks. The first track assumes that object locations are determined by an external procedure, for instance a user draws a bounding box around the object. As a result, images in both the training and testing sets are provided with a bounding box. The second track only expects that the bounding boxes are provided during the training stage. During the testing stage, it is up to the classification system to find the location of the object inside the image, if necessary.

3 Fisher standard pipeline

This section briefly describes the "standard" classification pipeline based on Fisher vector (FV) [20], as used by Xerox in prior competitions, *e.g.* in Pascal VOC [7] and Imagenet challenges [6]. A detailed comparison of this method with other techniques of the state-of-the-art is given by Chat-

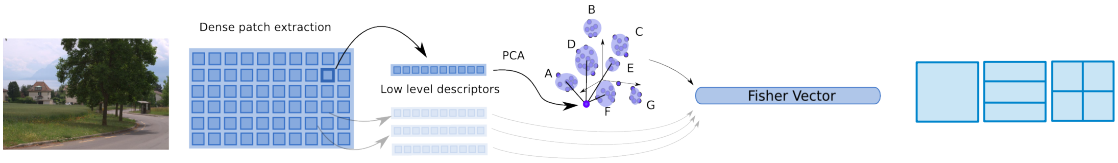


Figure 1: "Standard" Fisher pipeline used as reference in our paper.

field *et al.* [4], who conclude that it outperforms other coding techniques (such as bag-of-words or local linear coding) for classification tasks. However, in the latest Imagenet classification challenges, the FV was outperformed by a system based on deep learning [15]

The Fisher image classification pipeline consists of the following steps:

1. Down-sampling of the images to a fixed size of S pixels, keeping the aspect ratio from the original image. The images smaller than S pixels are not modified. This step drastically reduces the number of descriptors, and avoids extracting descriptors at small resolutions.

2. Extraction of SIFT on a dense multi-resolution grid. The number of resolutions is typically set to 5 and the step size (number of pixels between each sample) on x- and y-axis set to $s_x = s_y = 3$ pixels.

3. Post-processing of SIFT descriptors. First, the descriptor dimensions are reduced with PCA, typically to 64 or 80 components. This reduction is important for the next stage, as it ensures that the diagonal covariance matrix assumption is better satisfied. Second, a component-wise processing is applied to the raw SIFT descriptors: we consider both the non-linear processing known as RootSIFT [1, 9] and the similar $sign(x)\log(1+|x|)$ function used at Xerox in previous challenges.

4. Encoding with the FV. This step converts the set of local descriptors into a single vector representing the image. It relies on a Gaussian Mixture Model (GMM) formed of k Gaussians, assuming a diagonal covariance matrix. This Gaussian mixture is learned on the training set.

5. Spatial pyramid pooling [16] is also applied when using the FV: the image is partitioned into regions, each of which is represented by a FV obtained for the region descriptors. Several partitions are usually considered: $1 \times 1 + 3 \times 1 + 2 \times 2$ is a typical setting.

6. We post-process the FV with signed power-law normalization [19]. This step is parametrized by a parameter α, which is the exponent involved in the non-linear processing of each component x_i of the initial FV, as $x_i := \text{sign}(x_i) * |x_i|^\alpha$.

7. The resulting vector is ℓ_2-normalized and the cosine similarity is used as the similarity metric.

8. A 1-*vs*-rest support vector machine (SVM) linear classifier is trained and used to determine if the image belongs to a given class.

Color Descriptor. In addition to SIFT and as in previous participations of Xerox in image classification challenges, we additionally used a color descriptor, referred to as *X-color* in the rest of this report [5]. It encodes the mean and variance of R,G, and B color channels in each cell of a 4×4 grid partition of the patch, resulting in a $2 \times 3 \times 16 = 96$-dimensional descriptor. Apart from

Table 2: Comparison of our two sub-systems. Subsystem A (SA) is close to the Fisher classification pipeline described in Section 3. Subsystem B (SB), while also relying on Fisher vector, has been designed with the objective of being complementary with SA. A range of parameters indicate that we have cross-validated the parameter on a validation set (subset of training set).

Subsystem	SA	SB
Image (re-)sizing	100k pixels	100k–300k pixels
dense sampling	every 3 pixels (x & y)	every 3 pixels
input descriptor	SIFT+X-color	SIFT
post-processing	$x_i := sign(x_i)\log(1+\|x_i\|)$P	RootSIFT [1]
	PCA to d=96 for SIFT, to d=48 for X-color	PCA to 80 components
	X-color: no post-processing	filter low-energy patches ($\tau = 0..700$)
vocabulary size k	1,024	1,024 – 4,096
spatial coding	spatial pyramid [16]: $1 \times 1 + 3 \times 1$	spatial coordinate coding [14]
classifier	Stochastic Gradient Descent [3]	LASVM [2], $C = 100$

descriptor computation and local descriptor post-processing (none is applied), all other steps are identical with X-color. The corresponding FV is complementary to that produced with SIFT descriptors.

4 Adapting the Fisher vector to FGC

We have designed a fine-grained image classification system, which consists of two subsystems, both of them based on FV. All parameters have been optimized on a *per-domain* basis.

The subsystem SA implements the Fisher processing pipeline described in Section 3. The main differences are 1) the optimization of a few parameters assumed to be important for FGC and 2) the choice of a $1 \times 1 + 3 \times 1$ grid for the spatial pyramid (we have not used the 2×2 grid to limit the dimensionality of the vector when considering large vocabularies).

The subsystem SB is constructed such that:

- It is as complementary as possible with SA, so that the late fusion of the two subsystems is likely to give a significant boost compared with SA used alone. In order to achieve such a complementary system, we have made different choices in several steps of the processing pipeline, particularly when post-processing local descriptors, and when exploiting spatial information.
- It focuses more on the optimization of some domains (namely aircraft, cars and shoes) that can be considered as instance classification. These visual objects correspond to manufactured, man-made objects. Unlike dogs and shoes, we expect little intra-class variability. We also observe less texture on the object itself and in the background.

This section first focuses on demonstrating the importance of the parameters involved in our system and strategies that are specifically adapted to specific domains. Then, we discuss different design choices, which are summarized in Table 2. All the results we present are obtained by cross-validation on the training set, because the annotation of test images is currently not available. We split this set into *learn* (75% of training set) and *val* (remaining 25% images).

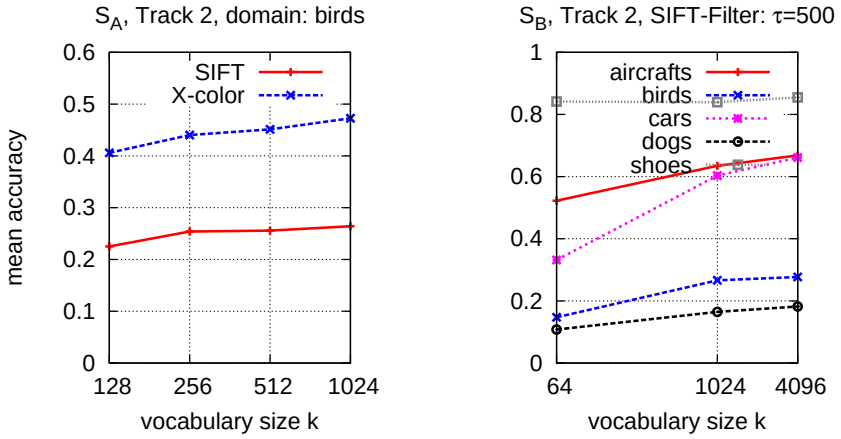


Figure 2: Impact of vocabulary size on performance. *Left,* for the SA standard pipeline with spatial pyramid coding (shown only for the 'bird' domain). *Right,* for SB (in this case, without spatial coding, images down-sampled to 100k pixels).

4.1 Large vocabularies

In fine-grained classification, several domains deal with man-made objects. Therefore, one would expect a low intra-class variability, although this has to be compared with the inter-class variability. One of the key parameters reflecting the variability of the classes is the vocabulary size. In large-scale particular object retrieval, the vocabulary size in bag-of-words is chosen to be very large, comprising up to 1 million visual words. If we consider that fine-grained classification is a visual recognition task that is in between image classification and particular object recognition, it is worth evaluating the impact of the vocabulary size on the performance.

For the system SA, we have arbitrarily used $k = 1,024$ Gaussians for both types of features and for all domains. This choice was mostly guided by previous results in regular image classification, where this particular choice is large enough to saturate the performance [20]. Note that, even without the spatial pyramid, this choice already gives a high-dimensionality to the vector ($D = 2 \times 64 \times 1024 = 131,072$). As shown later for SB, this choice is actually suboptimal and can be improved by setting these parameters on a per-domain basis.

Figure 2 shows the impact of the vocabulary sizes in both our subsystems. As one can see, the performance increases for most subdomains. Apart from the shoes domain, we have actually not reached a point of saturation. This suggests that better performance could be further increased by increasing the parameter k, with the caveat that we have to deal with very high-dimensional vectors. This problem is partially addressed in SB by an alternative choice for the spatial coding strategy, see later in this section.

However, to keep complexity at a reasonable level, we have set k=4,096 for aircrafts/birds/cars, k=2,048 for dogs (for computational reasons as dogs is the largest domain) and k=1,024 for shoes.

Notice the much better performance of X-color descriptor compared to that achieved with SIFT for the bird domain. We note that this conclusion is specific to the bird domain, for which color is a very discriminative clue. The subsystem SB does not use any color and is therefore expected to be inferior for this domain.

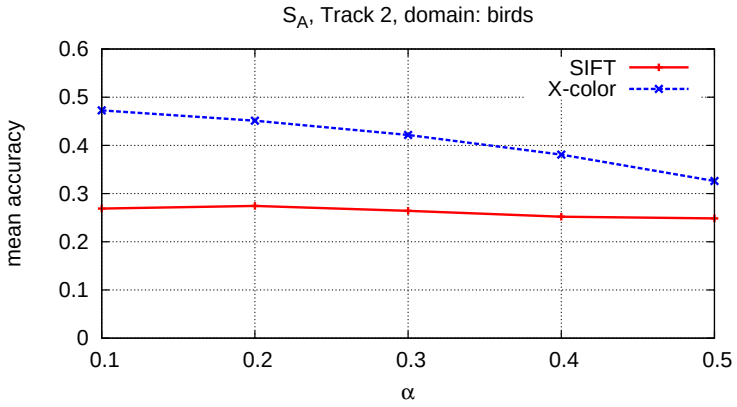


Figure 3: Cross-validation (in SA) of the α parameter involved in the power-law normalization, for the bird domain. In general, $\alpha = 0.3$ was best for SIFT and $\alpha = 0.1$ was best for X-color. Consequently, we used these values in SA. For SB, we set $\alpha = 0.1$.

4.2 Power-law

Power-law normalization has become a *de facto* post-processing stage applied after coding schemes such as bag-of-words [11] or Fisher vectors [19]. Its positive effect is related [13] to the non-*iid* behavior of the descriptors, more specifically the burstiness effect. As mentioned in Section 3, it is parametrized by a single parameter α, which is often fixed in the literature. In our case, we have cross-validated this parameter for both SIFT and X-color descriptors. The results are shown in Figure 3, where it can be observed that small values provides much better performance with X-color. The performance is more stable for SIFT in the interval [0.1,0.3]. Therefore, in SA we set $\alpha = 0.3$ and $\alpha = 0.1$ for SIFT and X-color, respectively, while in SB we complementarily set $\alpha = 0.1$. Slightly better results are obtained on the validation set by setting these parameters on a per-domain basis.

4.3 Resolution

In systems relying on dense sampling, it is often considered necessary to down-sample the images whose resolution is too large. While reducing the image size is mostly considered for computational reasons, i.e., to limit the number of descriptors to a tractable number (otherwise, this number could be as large as hundreds of thousands), Table 3 reports the relationship between performance and image size. As one can observe, the largest resolution generally offers the best performance.

We set S=300k pixels for aircrafts, birds and shoes, and S=100k pixels for dogs and shoes.

4.4 Filtering strategy

We introduce a specific filtering of low-energy descriptors, based on the observation that these patches are not discriminant in a fine-grained classification context. The method is simple: before ℓ_2-normalizing the SIFT descriptor, we compute the ℓ_2 of the patch and compare it to a threshold τ. This strategy can be seen as an extension of a filtering stage used by Jain *et al.* [9], who filter the patches whose quantized values of the gradients are strictly equal to 0. In our case, we apply this strategy in a more extreme manner by setting a threshold τ that filter significantly more

Table 3: Performance per domain as a function of the image resolution (down-sampling). Evaluation is done for subsystem SB in Track 2: $k = 64$, $\tau = 500$, $\alpha = 0.1$. We do not include the dog domain in this comparison, as the corresponding number of images is large (see Table 1) and we agnostically set the resolution to 100k to limit the computational overhead.

domain	100k	300k
aircrafts	0.635	**0.668**
birds	0.266	**0.293**
cars	**0.603**	0.565
shoes	0.839	**0.862**

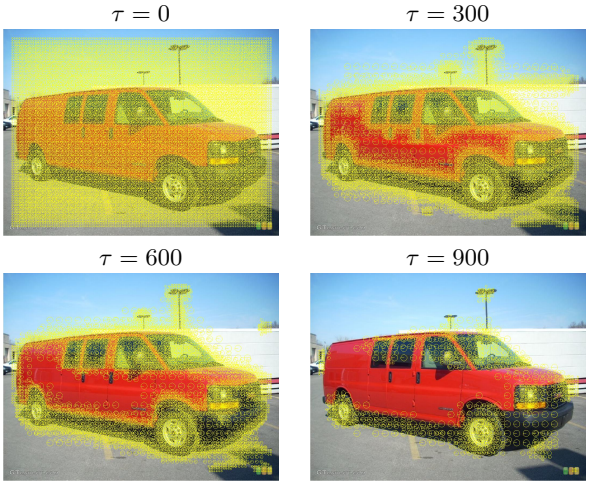


Figure 4: Impact of the filtering step on the selected dense patches.

patches. Note that even in the case $\tau = 0$, we remove some patches (those whose gradients are 0, similar to Jain *et al.*).

The consequence is that we remove uniform patches, which are quite common in some domains such as aircraft where the objects are often depicted in the sky. This is also the case for smooth objects like cars, whose interior regions are uniform. Furthermore, with τ large enough, blurry patches are discarded and generally only corners and edges are preserved. Considering the scale of patches, smaller patches are more likely to be removed than larger patches, and thus this increases the weight of higher scales. An example of filtering is shown in Figure 4, which shows the effect of filtering for different values of the threshold τ.

The filtering is consistently applied to descriptors used to train the Gaussian Mixture Model, which focuses more on high-energy patches. The expected benefit is to remove the weights of uninformative patches in the Fisher vector. This result in some cases in an increasing of classification accuracy. For instance in the competition all domains but shoes benefit from this filtering, as shown in Figure 5.

Finally and as shown in Figure 6, this filtering step significantly reduces the number of extracted descriptors, and lowers the computational complexity without penalty on performance. In most domains, τ gives comparable values of accuracy for a relatively large range of values. We favor a stricter filtering (larger value of τ) in order to reduces the computational cost of the

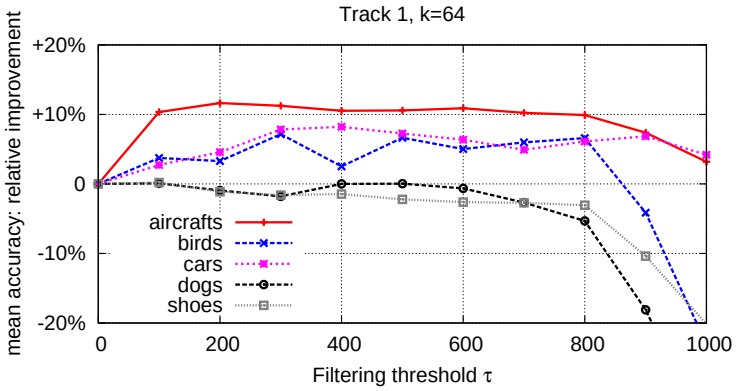


Figure 5: SB: Impact of the dense-SIFT filtering strategy for the different domains (Track 1, final submission setup, except for $k = 64$).

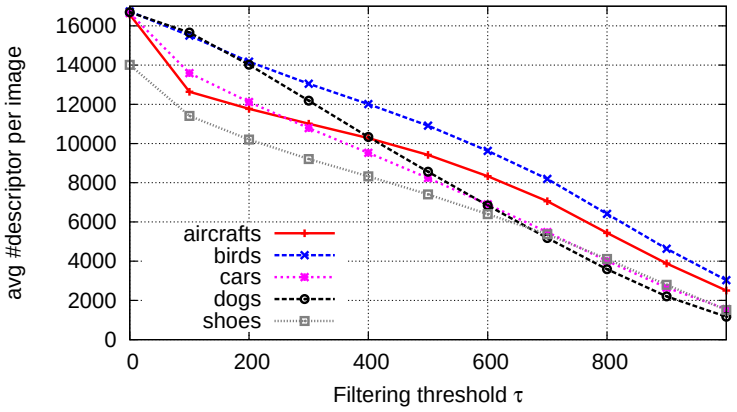


Figure 6: SB: Impact of the dense-SIFT filtering strategy on the average number of patches kept per image (images down-sampled to 100k pixels).

subsequent Fisher vector computation, which linearly depends on the number of descriptors.

4.5 Dealing with unbalanced training set

In SB, we do not rely on SGD but on a more conventional linear SVM solver, namely the LASVM package [2]. For each class of a domain, the naive way of training is to use all images in the class as positive samples, and the remaining images as negative samples. However, due to the very small number of positive samples per class (about 50), this lead to extremely unbalanced training sets, and thus to suboptimal classifiers.

In order to mitigate this problem, we build for each class a training set consisting of all the positive samples, and a selection of negative samples. The selection is performed by computing for each negative sample the average similarity to all positive samples. In the case of Fisher vectors, the similarity is the dot product between a negative sample and the mean of positive samples. Then, we rank all negative samples according to this average similarity, and select the ones that maximize it. The number of selected negative samples is a ratio of the number of positive samples. For the challenge, we use a ratio 40:1. This lead to an average of 2000 negative samples for each class.

In addition to increasing the accuracy of classifiers, this strategy is also very appealing for large scale training, since the size of class-specific training sets no longer depends on the size of the whole training set. In order to get a fully scalable training procedure, the selection of negative samples have to be performed with a scalable similarity measure, for instance LSH [17], Hamming Embedding [10] or compact feature vectors [12].

Remark: Note that our SA system uses a SGD solver, which uses its own strategy (randomly sampling a given number of negatives for each positive sample) to address this issue [20].

4.6 Overview of optimization strategy

It is unfeasible to test all the possible combinations of the parameters, given that we rely on limited computational power. The number of parameters tested is bounded by the resources required to make this optimization. We performed a first set of preliminary experiments aimed at determining the typical range of interesting parameters, which were not too costly to compute. In particular, we selected $k = 64$ to limit the dimensionality of the vectors. Then, in order to reduce the cost of performing the whole cross-validation of all parameters jointly, we adopted the following order for the subsystem SB:

- Resolution to which the images are down-sampled ;
- Spatial coding (σ_x and σ_y) jointly with filtering strategy (threshold τ) ;
- Filtering threshold τ ;
- Vocabulary size k.

The cross-validation of these parameters is not done class-wise, due to large risk of overfitting and of obtaining inconsistent scores across classes. Instead, we carried out the cross-validation *per-domain*.

Note that our first pass of cross-validation demonstrated the need to cross-validate the parameters σ_x and σ_y jointly with the filtering threshold τ. The parameters τ and k have a strong impact on complexity: large τ filters more descriptors and therefore reduces the complexity, while large k increases the complexity. Considering both accuracy and these computational constraints, we finally fixed the parameters shown in Table 4 for Track 1 and in Table 5 for Track 2. Note, for the domain shoes, $\sigma_x = \sigma_y = 0$ means that the spatial coordinate coding is not useful and was not used.

Table 4: Track 1: Parameters fixed from cross-validation results and complexity constraints for SB.

domain	τ	σ_x	σ_y	k
aircraft	700	100	500	4096
birds	700	10	50	4096
cars	700	10	50	4096
dogs	700	10	50	2048
shoes	0	-	-	1024

Table 5: Track 2: Parameters fixed from cross-validation results and complexity constraints for SB.

domain	τ	σ_x	σ_y	k
aircraft	800	100	500	4096
birds	900	75	100	4096
cars	900	-	-	4096
dogs	600	-	-	2048
shoes	0	-	-	1024

4.7 Late fusion strategy

The proposed system implements two fusion stages:

- The late fusion of the classification scores from the SIFT-based representation and the X-color-based representation to give the final scores for SA;
- The late fusion of the scores from SA and SB.

In both cases, the fusion score s_f is a linear combination of the score provided by the input systems, as $s_f = ws_{c1} + (1-w)s_{c2}$, where w is a value in in the range $[0, 1]$, s_{c1} is the score from the first classifier and s_{c2} is the score from the second. Values of w were chosen via cross-validation on a per-domain basis. The resultant values for both tracks are shown in Figure 7. Note that the classification scores were not calibrated prior to late fusion so that w does not exactly correspond to the relative accuracy of each source of scores. However the weights are broadly consistent with the relative accuracy of each source of scores for a given domain. For example, X-color is highly weighted for animal categories where the color of an object (a bird or dog) is very consistent, but has low weights for man-made objects with very variable colors (cars, shoes and aircraft).

5 Results

This section presents the official results obtained by our system compared to those of all other participants[1]. For the submission, we have used the whole training set to train the SVM classifiers. For SB, we augment it by mirroring the images, as we assume that a mirrored image is also a valid instance of the target class. On our validation set, we validate that this choice increases the classification accuracy.

[1]Official results: https://sites.google.com/site/fgcomp2013

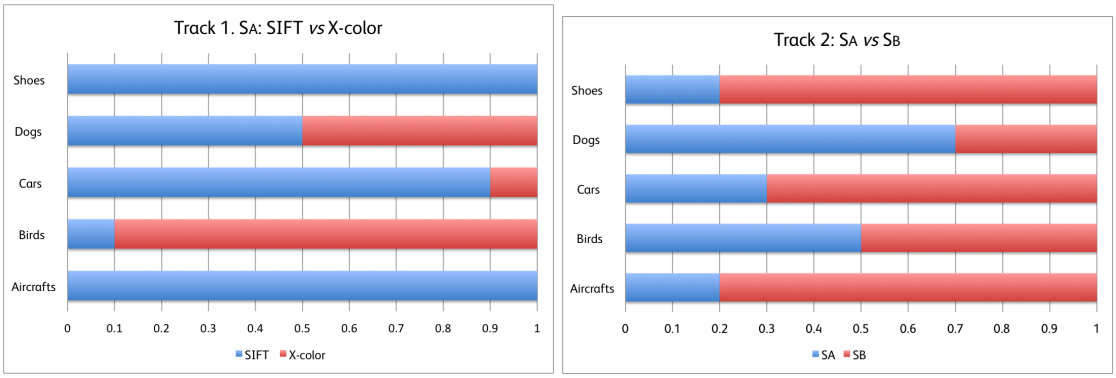


Figure 7: Cross-validated late fusion weights for Track 1 (per domain). The results obtained in Track 2 marginally differ.

The results for Track 1 and Track 2 are shown in Tables 6 and 7, respectively. As one can see, our whole system outperforms all others, including the methods that have used external data for training. We have also submitted separately SA and SB, in order to measure the individual performance of each subsystem, as well as the benefit of our choice to seek complementary methods.

The subsystem SA is better than SB for the domains birds and dogs. This is expected, as color is important for these domains, as already suggested by the cross-validated weights in the late fusion step 7. The inverse conclusion holds for the domain cars and shoes. This is also consistent with our cross-validated weights. This is, in our opinion, mainly due to the use of larger vocabularies and the use of our filtering strategy in SB.

6 Conclusion

In this paper, we have described several adaptations of the Fisher vector which improve its performance in a context of fine-grained classification. The main conclusions that we draw are as follows. First, large vocabulary sizes are important for fine-grained recognition. For most domains, the best performance is obtained with the largest mixture model and we did not observe any saturation. This suggests that better performance could be achieved by further increasing this size, although this would also raise computational issues. We partially alleviate the size problem by using spatial coordinate coding instead of the usual spatial pyramid, enabling our mixture model to be defined with up to 4,096 Gaussians.

Second, we have shown it is possible to obtain a large gain by using several Fisher vector-based systems, given that complementary design choices are made. In particular, complementary low level descriptors translate to complementary Fisher vectors, as shown in particular by considering the SIFT and X-color descriptors. Third, we have shown the interest, for several visual recognition domains, of using a simple filtering strategy that both boosts the accuracy of the Fisher vector and reduces its processing cost.

Overall, these simple ingredients and insights led us to establish a new state-of-the-art in fine-grained classification, as demonstrated through our participation in the FGCOMP fine-grained classification challenge, where we obtained the best results over all the participants.

Table 6: FGCOMP's Official results in Track 1. The asterisk * indicates that external data was used for learning. These runs are therefore not directly comparable.

Team	Aircrafts	Birds	Cars	Dogs	Shoes	Overall
Ours: SA +SB	81.46	71.69	87.79	52.90	91.52	77.07
CafeNet*	78.85	73.01	79.58	57.53	90.12	75.82
Ours: SA	75.88	66.28	84.70	50.42	88.63	73.18
VisionMetric*	75.49	63.90	74.33	55.87	89.02	71.72
Symbiotic	75.85	69.06	81.03	44.89	87.33	71.63
Ours: SB	80.59	58.54	84.67	35.62	90.92	70.07
CognitiveVision*	67.42	72.79	64.39	60.56	84.83	70.00
DPD_Berkeley*	68.47	69.58	67.40	50.84	89.52	69.16
VisionMetric	73.93	51.35	69.31	38.63	87.33	64.11
CognitiveVision	58.81	51.69	52.37	47.37	78.14	57.68
MPG	9.45	54.57	69.27	42.92	88.42	52.93
MPG	9.45	56.47	63.77	0.97	88.42	43.82
Infor_FG*	30.39	9.06	4.45	0.82	35.23	15.99
InterfAIce	5.79	2.56	1.12	6.96	5.99	4.48

Table 7: FGCOMP's official results in Track 2.

Team	Aircrafts	Birds	Cars	Dogs	Shoes	Overall
Ours: SA +SB	80.74	49.82	82.71	45.71	88.12	69.42
Symbiotic	72.49	46.02	77.99	37.14	89.12	64.55
Ours: SA	66.40	44.51	76.35	43.96	86.33	63.51
Ours: SB	80.74	34.45	76.89	24.40	87.33	60.76
DPD_Berkeley*	45.51	42.70	43.38	41.91	59.98	46.70
Infor_FG*	9.66	5.75	3.71	32.71	4.69	11.30
InterfAIce	5.43	2.58	1.17	6.94	5.29	4.28

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.

[2] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619, September 2005.

[3] L. Bottou. Sgd. http://leon.bottou.org/projects/sgd.

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

[5] S. Clinchant, G. Csurka, F. Perronnin, and J.-M. Renders. XRCE's participation to ImageEval. In *ImageEval Workshop at CVIR*, 2007.

[6] W. Dong, R. Socher, L. Li-Jia, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International journal of Computer Vision*, 88(2):303–338, June 2010.

[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, September 2010.

[9] M. Jain, R. Benmokhtar, P. Gros, and H. Jégou. Hamming embedding similarity-based image classification. In *ICMR*, June 2012.

[10] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision*, October 2008.

[11] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[12] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, January 2011.

[13] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local descriptors into compact codes. In IEEE *Transactions on Pattern Analysis and Machine Intelligence*, September 2012.

[14] P. Koniusz, F. Yan, and K. Mikolajczyk. Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer Vision and Image Understanding*, 17(5):479–492, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Image classification with deep convolutional neural networks. In *NIPS*, 2012.

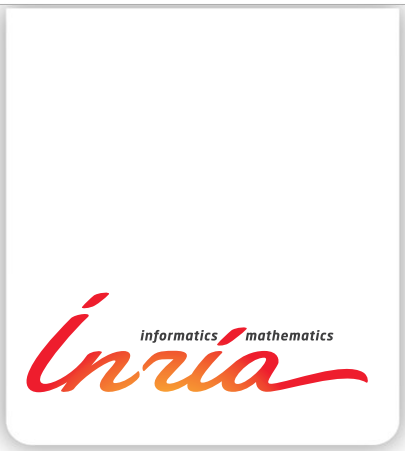
[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2006.

[17] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In *Proceedings of the International Conference on Very Large DataBases*, pages 950–961, 2007.

[18] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

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

[20] Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of Computer Vision*, 2013.



RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE

Campus universitaire de Beaulieu
35042 Rennes Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399