Revisiting the Fisher vector for fine-grained classification

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

Pattern Recognition Letters
journal homepage: www.elsevier.com

Revisiting the Fisher vector for fine-grained classification

Philippe-Henri Gosselin[a,b,**], Naila Murray[c], Hervé Jégou[b], Florent Perronnin[c]

[a]*ETIS / ENSEA - UCP - CNRS, Cergy, France*
[b]*Inria, Rennes, France*
[c]*Xerox Research Center Europe, Grenoble, France*

ARTICLE INFO

Article history:
Received ?
Received in final form ?
Accepted ?
Available online ?

Keywords:
Computer Vision
Fine-grain Classification
Challenge

ABSTRACT

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

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.

As noted in (Chai et al., 2013), FGC differs from standard coarse-grained classification (CGC) in two significant ways:

- Property 1: while in CGC classes exhibit global differences, in FGC classes often share the same global appearance. Therefore, two classes may be visually distinguishable only based on subtle localized details. To better illustrate this challenge, we present in Fig. 1 examples of classes for the airplane domain. For example, if we focus on aircrafts, the FGC system has to distinguish between the different version of Boeing 747 aircrafts. Note that it is possible, for instance if one counts the windows.

- Property 2: while in CGC the background provides valuable context for categorization, in FGC it is rarely discriminative and consequently acts as a source of noise.

For these two reasons, FGC is perceived as significantly more challening than CGC. While the best performing approaches to image classification are fairly well-identified – and include Fisher vectors (Perronnin and Dance, 2007; Perronnin et al., 2010; Chatfield et al., 2011), deformable part models (Felzenszwalb et al., 2010) and deep learning approaches (Krizhevsky et al., 2012) – it is still unclear which approaches perform best for FGC and how they should be adapted to better address the specificities of the problem.

The goal of this paper is to evaluate the suitability of the Fisher Vector (FV) in this context. Starting from the standard FV pipeline used in CGC, we derive two subsystems (SA and SB) with different focus, and then combine them in a final solution. These systems have been designed and optimized in the context of the joint participation of Inria and Xerox in the FGComp 2013 fine-grained challenge. This challenge evaluates systems for 5 different FGC problems thanks to 5 independent

**Corresponding author: Tel.: +33 1 30 73 66 11; fax: +33 1 30 73 66 27;
e-mail: gosselin@ensea.fr (Philippe-Henri Gosselin)

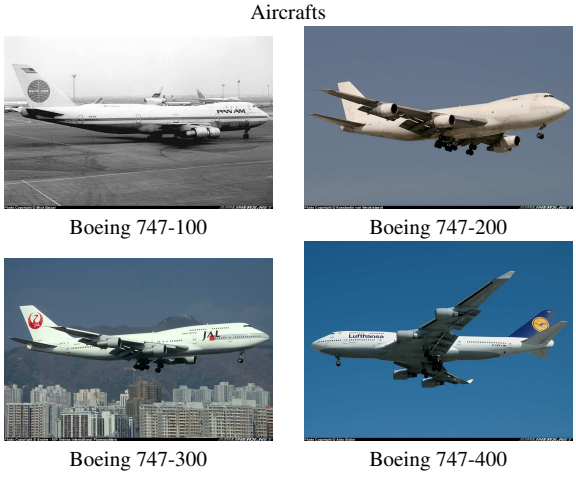

Fig. 1. Examples of aiplane classes. The system has to find the visual differences between these images, using only few training samples.

datasets. For this purpose, we split the training set into two sets (75% for learning and 25% for validation).

Our main contribution is to reveal the parameters which are crucial for high-accuracy FV-based FGC. For instance, we show that large vocabularies enable to model subtle visual details, and that a properly cross-validated power-normalization ensures that these details are not overpowered by frequent patches (see property 1). We also show that techniques such as SIFT filtering help in removing non-discriminative visual content in image backgrounds (see property 2). Another important contribution of this work is to show that, despite the recent success of deep-learning-based techniques (Krizhevsky et al., 2012) in challenges such as ImageNet, the shallow pipeline that involves coding and pooling patch-based descriptors is still very much relevant in the FGC context.

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 (Everingham et al., 2010) and Imagenet (Dong et al., 2009). 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 to. To evaluate different FGC scenarios, FGComp considers 5 datasets, each dataset evaluating a specific problem of FGC:

1. aircrafts. This dataset is compound of photographs of aircrafts in the sky or in airports (Maji et al., 2013). Classes are very specific models of aircrafts, as illustrated in Fig. 1.

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

Table 2. FGComp 2013 challenge: image properties on train set.

	image resolution (pixels)			
domain	min	average	std. dev.	max
aircraft	330k	840k	270k	2M
bird	100k	180k	20k	250k
car	5k	480k	980k	21M
dog	9k	190k	200k	8M
shoes	307k	307k	0	307k

2. birds. This is a subset of the CCUB NABirds 700 dataset, a.k.a. CUB-200[1]. This is a collection of photographs in natural environment of birds species that are commonly observed in North America. Birds species can share very similar visual characteristics as well as very different colors and shapes.
3. cars. This dataset focuses on the detection of car models, including different releases of the same car, as illustrated in Fig. 1. Most photographs were taken in urban environment.
4. dogs. This is a collection of dogs species. This dataset has many variability in terms of photographic conditions, like view angles, image resolutions and object counts.
5. shoes. This dataset is very different from the other ones, since this is photographs of shoes in very specific conditions: always a pure white background, same image resolution and well-chosen view angle.

Tables 1 and 2 summarize 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.

[1] http://www.birds.cornell.edu/nabirds/

3. Fisher standard pipeline

This section briefly describes the "standard" classification pipeline based on Fisher vector (FV) (Sánchez et al., 2013), as used by Xerox in prior competitions, *e.g.* in Pascal VOC (Everingham et al., 2010) and Imagenet challenges (Dong et al., 2009). A detailed comparison of this method with other techniques of the state-of-the-art is given by (Chatfield et al., 2011; Huang et al., 2014), 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 (Krizhevsky et al., 2012)

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 (Arandjelovic and Zisserman, 2012; Jain et al., 2012) 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 (Lazebnik et al., 2006) 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: $1\times1+3\times1+2\times2$ is a typical setting.
6. We post-process the FV with signed power-law normalization (Perronnin et al., 2010). 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 (Clinchant et al., 2007). 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 $2\times3\times16=96$-dimensional descriptor. Apart from descriptor computation, 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 several parameters assumed to be important for FGC and 2) the choice of a $1\times1+3\times1$ 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 3. 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). Performance values presented in curves are mean over 5 runs. Standard deviation over these 5 runs are always negligible (about 0.2%–1%), and thus are never plotted.

4.1. Large vocabularies

The visual vocabulary used to generate our Fisher vectors must be granular enough to to ensure discriminability between classes, but not so granular as to over-segment the feature space. For coarse-grained visual recognition, a small number of Gaussians (between 256 and 1024) are usually sufficient. However, when inter-class differences classes are subtle, for instance in large-scale particular object retrieval, the vocabulary size is chosen to be very large, comprising up to 1 million visual words. As fine-grained classification may be thought of as in between image classification and particular object recognition,

Table 3. 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		every 3 pixels
input descriptor	SIFT	X-color	SIFT
desc. filtering	no	no	filter low-energy patches
desc. post-processing	$sign(x)\log(1+\|x\|)$	no	RootSIFT (Arandjelovic and Zisserman, 2012)
desc. PCA	96	48	80
vocabulary size k	1,024		1,024 – 4,096
spatial coding	spatial pyramid (Lazebnik et al., 2006): $1\times1+3\times1$		spatial coordinate coding (Koniusz et al., 2013)
classifier	Stochastic Gradient Descent (Bottou)		LASVM (Bordes et al., 2005), $C=100$

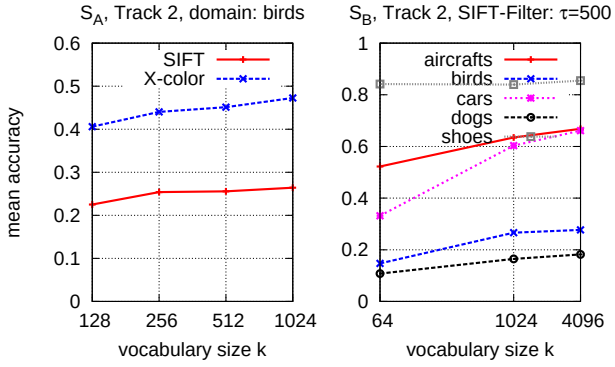

Fig. 2. Impact of vocabulary size on performance. *Left,* for the SA standard pipeline with spatial pyramid coding (shown only for the 'bird' domain). *Right,* for SB (in this case, without spatial coding, images down-sampled to 100k pixels).

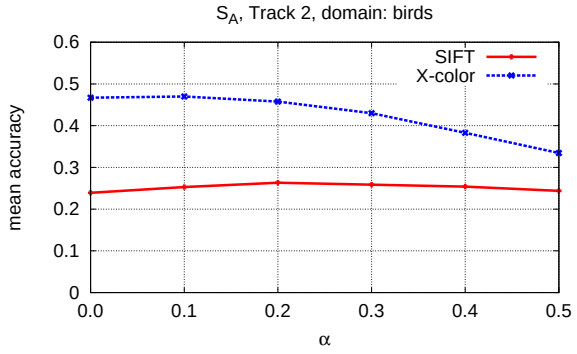

Fig. 3. Cross-validation (in SA) of the α parameter involved in the power-law normalization, for the bird domain. In general, $\alpha=0.3$ was best for SIFT and $\alpha=0.1$ was best for X-color. Consequently, we used these values in SA. For SB, we set $\alpha=0.1$.

in terms of class granularity, it is worthwhile to evaluate the impact of the vocabulary size on performance.

For the system SA, we have used $k=1,024$ Gaussians for both types of features and for all domains. This choice was mostly guided by storage and memory requirements, Indeed, even with a simple $1\times1+3\times1$ pyramid, $k=1,024$ already gives high-dimensional vectors ($D=2\times64\times1024\times4=524,288$). As shown later for SB, $k=1,024$ might not be optimal and more Gaussians can improve accuracy.

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 coordinate coding strategy (SCC) (Koniusz et al., 2013). SCC consists in augmenting each descriptors with values $\sigma_x x, \sigma_y y$, where $x,y\in[0,1]^2$ are the descriptor coordinates in the image, and $\sigma_x,\sigma_y\in\mathbb{R}$ weights tuned though cross validation. 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.

4.2. Power-law

Power-law normalization has become a *de facto* post-processing stage applied after coding schemes such as bag-of-words (Jégou et al., 2009) or Fisher vectors (Perronnin et al., 2010). Its positive effect is related (Jégou et al., 2012) to the non-*iid* behavior of the descriptors, more specifically the burstiness effect. This is all the more important for FGC as this ensures that infrequent (yet potentially highly informative patches) are not overpowered by frequent (yet not necessarily informative patches) such as uniform background patches. As mentioned in Section 3, this normalization 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

Table 4. 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**

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 4 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

While sophisticated techniques can be employed to focus the FGC process on the object and its most discriminant parts Chai et al. (2013), we introduce a simple technique which involves filtering of low-energy descriptors. It is based on the observation that these patches are not discriminant in a FGC context. 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.* (Jain et al., 2012), 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 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 results in an increase of classification accuracy for aircrafts, birds and cars domains. However, for dogs we don't see any improvement, which might be explained by high-frequencies textures of these objects. For shoes, since objects are always on a white background, filtering has no effect. For instance in the competition

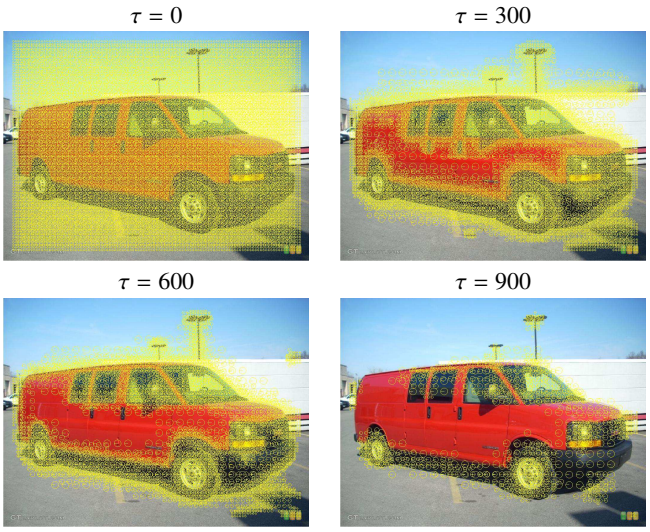

Fig. 4. Impact of the filtering step on the selected dense patches.

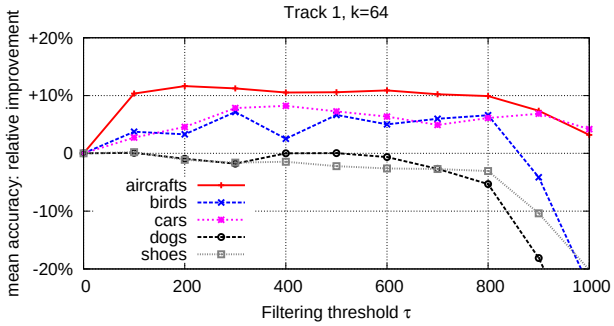

Fig. 5. SB: Impact of the dense-SIFT filtering strategy for the different domains (Track 1, final submission setup, except for $k = 64$).

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

4.5. Classifiers training

SA and SB employ different strategies to train SVM hyperplane classifiers. SA uses a Stochastic Gradient Descent (SGD) solver and employs a resampling strategy that involves randomly sampling a given number of negatives for each positive sample (Akata et al., 2013). SB relies on the LASVM package (Bordes et al., 2005). In order to speed up training, 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. 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

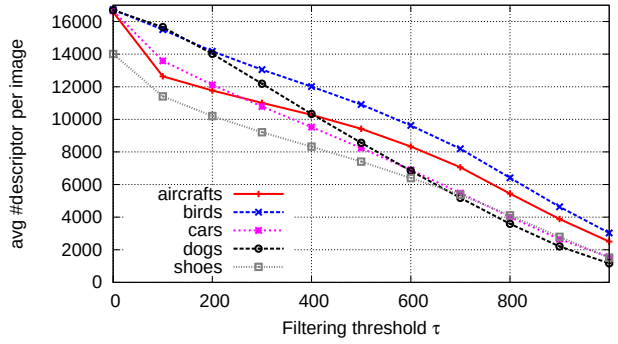

Fig. 6. SB: Impact of the dense-SIFT filtering strategy on the average number of patches kept per image (images down-sampled to 100k pixels).

Table 5. Track 1: Parameters fixed from cross-validation results and complexity constraints for SB.

	Track 1				Track 2			
domain	τ	σ_x	σ_y	k	τ	σ_x	σ_y	k
aircraft	700	100	500	4k	800	100	500	4k
birds	700	10	50	4k	900	75	100	4k
cars	700	10	50	4k	900	-	-	4k
dogs	700	10	50	2k	600	-	-	2k
shoes	0	-	-	1k	0	-	-	1k

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.

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 5. Note, for the domain shoes, $\sigma_x = \sigma_y = 0$ means that the spatial coordinate coding is not useful and was not used.

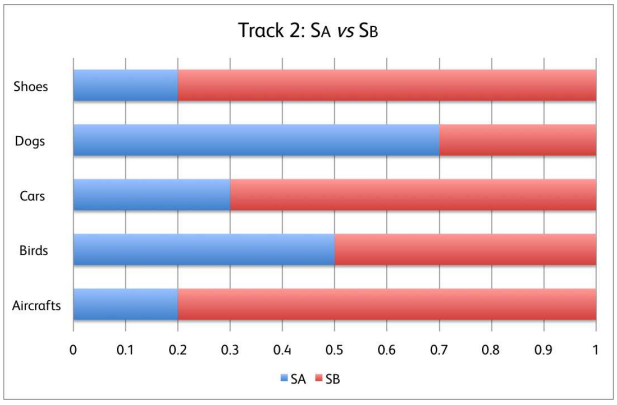

Fig. 7. Track 2: Cross-validated late fusion weights for systems SA (blue) *vs* SB (red).

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.

5. Results

This section presents the official results obtained by our system compared to those of all other participants[2]. 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.

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,

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

in order to measure the individual performance of each subsystem, as well as the benefit of our choice to seek complementary methods.

The subsystem S_A is better than S_B 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 S_B.

Other submissions to FGComp '13

Of the 9 participating teams, 5 had submissions which used deeply-learned features, including the CafeNet, VisionMetric, CognitiveVision, DPD_Berkeley and Infor_FG submissions. Each of these submissions required additional training data (for example the ImageNet dataset (Deng et al., 2009)) in order to adequately learn the feature representations. As such, these methods have a nominal training advantage. Of the remaining 4 submissions, 3 used Fisher-based feature-representations (Inria-Xerox, Symbiotic and MPG) and one used an information graph building algorithm (InterfAIce).

Top-performing methods include that of the CafeNet team, whose submission was an implementation of the convolutional-neural-net-based system of Krizhevsky *et al.* (Krizhevsky et al., 2012). The network was pre-trained with ImageNet 2012 data, and fine-tuned with FGComp data. The VisionMetric team used HOG features with LLC coding and combined these features with those from a pre-trained CNN. Distance metric learning provided a low-dimensional embedding. These two deep-learning methods achieved high performance but require very large amounts of training data, which was unavailable in this challenge, as is typical in fine-grained classification scenarios. This limited training data may have hampered their performance.

The Symbiotic team's submission was based on their state-of-the-art FGC system (Chai et al., 2013), which jointly trains part detection and segmentation models using training images and training bounding boxes. Fisher-encoded SIFT and color histograms were extracted from the foreground and each detected part and combined to form an image representation. Vertically-mirrored training images augmented the original training set. A key difference is the use of a small number of 256 Gaussians in (Chai et al., 2013).

The MPG submission was most similar to ours. However they used more low-level descriptors than we did, namely SIFT, RGB-SIFT, Opponent-SIFT, and C-SIFT. Their spatial pyramid was also more detailed (whole image + 4 quadrants + 3 horizontal strips). However, their visual vocabulary consisted of only 256 words, which in our experiments were not sufficient to encode subtle inter-class differences.

6. Conclusion

In this paper, we have described several adaptations to the Fisher vector which improve its performance in the FGC context. Our main recommendations for high-accuracy FGC are as follows. First, large vocabularies are important: dor 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. Second, properly cross-validating the power-normalization parameter is crucial in the FGC context. Third, patch filtering is a simple alternative to more complex object and part localization strategies Chai et al. (2013). Overall, these insights led us to establish a new state-of-the-art in FGC, as demonstrated by our winning participation in the FGComp challenge.

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: S_A +S_B	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: S_A	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: S_B	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: S_A +S_B	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: S_A	66.40	44.51	76.35	43.96	86.33	63.51
Ours: S_B	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

Acknowledgments

This work was supported by the Fire-ID project, funded by the French National Research Agency.

References

Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C., 2013. Label-embedding for attribute-based classification, in: CVPR.

Arandjelovic, R., Zisserman, A., 2012. Three things everyone should know to improve object retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

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

Bottou, L., . Stochastic gradient descent. http://leon.bottou.org/projects/sgd.

Chai, Y., Lempitsky, V., Zisserman, A., 2013. Symbiotic segmentation and part localization for fine-grained categorization, in: Proceedings of the International Conference on Computer Vision.

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

Clinchant, S., Csurka, G., Perronnin, F., Renders, J.M., 2007. XRCE's participation to ImageEval, in: ImageEval Workshop at CVIR.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

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

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

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32.

Huang, Y., Wu, Z., Wang, L., Tan, T., 2014. Feature coding in image classification: A comprehensive study. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 493–506.

Jain, M., Benmokhtar, R., Gros, P., Jégou, H., 2012. Hamming embedding similarity-based image classification, in: ICMR.

Jégou, H., Douze, M., Schmid, C., 2009. On the burstiness of visual elements, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C., 2012. Aggregating local descriptors into compact codes, in: IEEE Transactions on Pattern Analysis and Machine Intelligence.

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

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Image classification with deep convolutional neural networks, in: NIPS.

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

Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A., 2013. Fine-Grained Visual Classification of Aircraft. Technical Report. arXiv.

Perronnin, F., Dance, C.R., 2007. Fisher kernels on visual vocabularies for image categorization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Perronnin, F., J.Sánchez, Mensink, T., 2010. Improving the fisher kernel for large-scale image classification, in: Proceedings of the European Conference on Computer Vision.

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