Fisher vector places: Learning compact place specific descriptors

Anonymous Author(s)
Affiliation
Address
email

Abstract

The aim of this work is to localize a query photograph by finding other images depicting the same place in a large geotagged image database. The contribution of this paper is as follows. We represent each database image by a Fisher vector and for each vector we learn per-location SVM classifiers. We show that trained and re-normalized SVM weight can be used as an embedded descriptor that replaces original Fisher vector. Though the SVMs were learned independently no further calibration is necessary. We perform two experiments on state-of-the-art datasets for place recognition and we show that our method consistently improves the results for different dimensions of decorrelated Fischer vectors.

1 Introduction

Internet contains a huge collection of imagery that grows literally every second. For instance, in 2013 Facebook claimed [7] that every single minute users upload about 208,300 new images, for Instagrama this number was about 27,800. This makes about 340,000,000 new images per day. Another interesting factor is that every minute users will upload more than 100 hours of video on Youtube. Considering the fact that most of the videos are in HD make these numbers even more impressive.

Imagine that there was a system that takes a picture from the imagery as an input and it shows you a position on the map indicating where the input image was captured. What would be this system useful for? One could take a picture from his smartphone and use it as an alternative to GPS or to precise localization in urban areas when the GPS was noisy. Surely, it would find its place in security and surveillance applications where one could localize an unknown image or video sequence based on its appearance. An alternative application would be a path reconstruction from a first-person-view or UAV video ẏhich was cheaper then state-of-the-art SLAM. Interesting may be a localization of scenes in old movies or collecting images from one place and observe how the place has changed over time.

Indeed, many of these systems already exist [15, 24] and have been intensively developed in last decade. In place recognition problems, researchers use many diverse techniques and aim for slightly different goals but the basic idea of the place recognition is the same; There is a structured collection of imagery where each image contains information about its location in the environment. This information can be for instance a GPS location or an image position in the semantic graph. Such a structured image collection is called a database. Given an unknown query image, the goal is to find the most similar image in the database and use its structural information for query localization, e.g. plot its GPS position on the map.

The structured database may contain from few thousands to couple of millions of images. To find a similar image in the database is therefore a challenging task. As an example, the query and database images may depict the same 3D object (e.g. building) from a different camera viewpoint, under different illumination conditions or the object can be partially occluded. In addition, the database can be very large. For instance, we estimate that Google Street View of France contains more than 60 million panoramic images. Therefore, we want to represent the images by descriptors that are robust to viewpoint changes, occlusion, illumination conditions and at the same time these descriptors must be low dimensional (or very sparse) in order to perform efficient and cheap retrieval in a large database.

2 Related work

Unlike the other work in large scale place recognition [6, 4, 12, 21] and image retrieval [17, 20, 23], we do not build on the bag-of-visual-words representation [3, 23]. Instead, we use Fisher vectors (FV) [9] that are recently finding its place in image retrieval, place recognition [25] and other computer vision tasks [22, 13]. These high-dimensional image representations are well-suited for retrieval problems since they are very robust to view-point changes and are more discriminative than standard bag-of-visual-words (BOW). However, the main drawback of FV is that unlike BOW it is dense which makes it nearly unfeasible for usage because of the memory footprint and computational cost.

For the above-mentioned reasons researchers aim on FV optimization. As an example, a nice paper of Perronnin et al. [19] focuses on analysis and compression of FV for image retrieval. They explain how the FV is related to BOW and underline that power-normalization of FV is equivalent to a burstiness features suppression [8]. They benefit from the fact that as the power α goes to zero the α-normalized Fisher vector converges to a ternary representation which can be efficiently binarized. They compared their method with the Local Sensitive Hashing (LSH) and the Spectral Hashing (SH) and showed comparable results while achieving only 520 bits signature per image. This encouraging result have been outperformed by Jegou et al. [9] who achieved comparable results with just 128 bits signature by using product quantizers (PQ). Hence, the Fisher vectors can be significantly compressed by appropriate indexing. In general, these results have been achieved by optimizing:

(i) the aggregation of local image descriptors (FV)

(ii) the dimensionality reduction (e.g. PCA, Mahalanobis metric, whitening)

(iii) the indexing algorithm (eg. PQ, LSH, SH)

The lower the dimension before indexing step (iii) is the lower size of the signature is achieved after indexation. Therefore it is very important to project the FV into lower dimension. There are different methods for (ii) among which the very popular is PCA. This method, however, has a drawback. It discards uncorrelated 'noisy' dimensions that can actually contain discriminative information for image retrieval.This problem have been addressed in [22]. They use a labeled training data and utilize discriminative learning in order to learn a global low-rank metric which can be decomposed to a linear projection to low-dimensional space. Our method aim on improving the performance of the projected FV, hence it could be viewed as an intermediate step between (ii) and (iii) into the scheme above.

3 Approach overview

Our setup is similar to approach of [6] where authors learn a set of per-location classifiers for place recognition, one per each location in the map. In that paper they use a BOW representation and emphasize that subsequent classifier calibration is a critical issue. Due to the calibration procedure the method is less efficient in terms of computational cost and is hard to scale up. Need for calibration has been also reported in [16] where they learn per-exemplar SVM classifiers on HOG representation for object retrieval. We aim on embedding FVs in order to achieve better performance by learning. In contrast to above-mentioned papers, our method with FVs works per-se without need of any further calibration and can be therefore subsequently compressed by an indexing algorithm [9]. In addition, it does not increase the dimensionality of the problem as if it would be the case for sparse image representation.

Rather then global approach we propose to treat FVs locally. We aim on learning a SVM weight in turn for each FV in such a way that this weight can be subsequently used as a new descriptor that replaces the original FV. Our method could be plugged into the scheme above as an intermediate step before indexing. To achieve the goal, for each FV in turn we: (i) collect hard negatives for given FV, (ii) learn a linear SVM weight (iii) and re-normalize the learned weight. At query time we compute a FV for the query image and measure its similarity with database images by computing a dot-product between the FV and each of the re-normalized SVM weights. We observe that learned re-normalized weight does not differ to much from the original FV. Loosely speaking, we embedded the information about hard negative exemplars into the original FV. The details will be given later in the text.

The main contributions of this paper are as follows: We show that trained and re-normalized SVM weight can be used as an embedded descriptor that outperforms the original one without need for further calibration. We perform two experiments on state-of-the-art dataset for place recognition and we show that our method consistently improves the results for (i) different PCA dimensions and (ii) over all recall@K metric shortlist, i.e. the rate of relevant images that are ranked in top K positions.

The rest of the paper is organized as follows: In section 4 we first give a brief overview of FV implementation for image retrieval. Then, in section 5, we the explain how the per-location classifiers are trained and finally in section 7 we present our experimental results along with implementation details.

4 Fisher vector overview

(This is a minimalistic overview of what the FV is. This can be omited or shrinked but I think it is worth since many people are not familiar with FV)
The Fisher vector can be thought of as an extension of BOW [23] that is achieved by considering a high-order statistics of the distribution of the feature descriptors. It aggregates a large set of descriptors into a high-dimensional representation of the fixed size. In the following text we briefly overview some basic concepts of Fisher vectors, its normalization and standard dimensionality reduction.

4.1 Computing Fisher vectors

It first starts with learning a generative model of the local image descriptors x of the dimension d (in our case we use SIFT features, see details in Sec. 7). The model is assumed to follow a GMM with N components and diagonal covariances. This model can be understood as a probabilistic visual vocabulary. Having the model trained, the Fisher vector of the sample of the image descriptors is computed as a gradient of the sample's likelihood with respect to the learned parameters of the GMM, which is subsequently scaled by inverse square root of the Fisher information matrix [18]. Considering only the derivatives w.r.t. the mean and covariances, we a obtain representation which captures the average of the first and second order differences between the descriptors and each of the GMM center k. For the image containing T feature vectors it can be expressed as follows:

$$\Phi_k^{(1)} = \frac{1}{N\sqrt{(w_k)}} \sum_{j=1}^{T} \alpha_j(k) \left(\frac{x_j - \mu_k}{\sigma_k} \right) \tag{1}$$

$$\Phi_k^{(2)} = \frac{1}{N\sqrt{(2w_k)}} \sum_{j=1}^{T} \alpha_j(k) \left(\frac{(x_j - \mu_k)^2}{\sigma_k^2} - 1 \right) \tag{2}$$

where w_k, μ_k and σ_k are weight, mean and diagonal of covariance matrix for k-th component of the learned GMM and $\alpha_j(k)$ is a soft assignment of the j-th feature x_j to the Gaussian k. The resulting Fisher vector Φ is then concatenation of these gradients which forms into a $2Nd$ dimensional vector such that $\Phi = [\Phi_1^{(1)T}, \Phi_1^{(2)T}, ..., \Phi_N^{(1)T}, \Phi_N^{(2)T}]^T$.

While some works, e.g.[19], use only a partial derivatives w.r.t. the mean parameters (equation (1)), we use partial derivatives both w.r.t. the mean and variance because this approach have been implemented in a library that we used (details are provided in section 7). As reported in [10] both approaches provide comparable results.

4.2 Normalization

As in the case of standard BOW, Fisher vector is typically $L2$ normalized in order to measure similarity between two FVs by a dot-product. However it has been shown that direct $L2$ normalization is suboptimal and better performance can be achieved by power-normalization followed by $L2$ normalization. Thus the normalization operator can be written as follows:

$$L(z) = \frac{\text{sign}(z)\,|z|^\alpha}{||\text{sign}(z)\,|z|^\alpha\,||_2} \tag{3}$$

where $\alpha \in (0, 1)$ is an element-wise power and $z \in R^d$ is an arbitrary real vector. As mentioned in the introduction, the power-normalization has the effect of suppression of the descriptors that occur frequently in the image (bursty features), these can be for instance periodical structures such as skyscraper windows or bricks of the wall. In our setup we use $\alpha = 0.5$ as it has been reported by Jegou et al. [11] that this value is optimal for wide range of GMM components N.

4.3 Dimension reduction

The resulted Fisher vector is a high-dimensional representation of an image, which dimensionality depends only on number of N components of the GMM and dimension d of the local image descriptors which is fixed. In practice, the PCA is being used for decorrelation and projection to a lower dimensional space because it subsequently directly affects a size of the indexed signature (see Sec. 2 and [9]).

We observed that after the PCA decorrelation it is important to $L2$ re-normalize projected FV to achieve better performance. Our interpretation is that after performing the PCA the higher dimensions contain a noise that affects only a magnitude of the projected vectors but not its direction Figure or not?. Furthermore we observed that even slightly better results can be achieved by using the normalization operator (3) instead of $L2$.

Finally, it is worth noting that PCA projection into a 'not-too-low' dimensional space can actually slightly improve the performance of the full FV. What that 'not-too-low' actually means have been partially discussed in [11]. However, in general as the dimension decreases the performance of decorrelated FV decrease which is, indeed, what we observe in our experiments. In the next section we therefore aim on enhancing the performance of projected FVs.

5 Per-exemplar SVM and Fischer vectors

I have renamed this seciton and skipped per location classifiers at all. In place recognition there are different definitions of what *the place* means. The definition depends on the goal of the particular

task. Should it be a pose of the camera or is it an ID of a building in the image? In our problem, *the place* is defined as an associated GPS location of the database image that captures the same scene as does the query image. Hence, given the unknown query image, the goal is to find a database image depicting the same scene and retrieve its GPS location.

Each database image j is represented by its Fisher vector Φ_j. The goal is to learn a set of new vectors Ψ_j, one per each database image, such that at query time, given the Fisher vector Φ_q of an unknown query image, we can either retrieve the the correct database image as the image j^* with the highest score

$$j^* = \arg\max_j \Phi_q^T \Psi_j \tag{4}$$

or use these scores to rank candidate images and use geometric verification to try and identify the correct location in an n-best list. Rather then tackle the problem as multi class learning we follow the approach of [16] and [6] and we learn Ψ_j independently for each image in turn using per-exemplar SVM. We aim on replacing the original database Fisher vector Φ with some new vector Ψ that performs better in sense of separation from hard negative examples and that has unit $L2$-norm in order to measure the similarity by a dot-product (eq. (4)).

To achieve the goal we will solve the per-exemplar SVM problem and then relax the solution. The per-exemplar SVM can be written as

$$\arg\min_{w_j, b_j} ||w_j||^2 + C_1 \cdot h(w_j^T \Phi_j^+ + b_j) + C_2 \sum_{\Phi \in \mathcal{N}_j} h(-w_j^T \Phi - b_j) \tag{5}$$

where Φ_j^+ is a single positive exemplar, Φ are FVs from negative training data, $||w_j||$ is a regularization term, h is a penalty function and C_1, C_2 are penalty weights for positive or negative data $\mathcal{N}_j$.

Clearly, the w_j can be of an arbitrary norm which depends on penalty parameters C_1, C_2 and training data. To find the unit vector Ψ we will relax the SVM problem in two steps: (i) We use SVM for ranking problem which is independent on b_j. (ii) Rather than normalizing w_j by additional constraint in objective (5) we first solve the SVM and then re-normalize w_j. The resulting vector Ψ can be then written as

$$\Psi_j = \frac{w_j}{||w_j||} \tag{6}$$

.

5.1 Training data

The negative training data set contains hard negative examples of the image j. These hard negatives are database images that are spatially far-away from image j and, at the same time, have a high similarity score with image j measured by a dot-product of its FVs. This approach follows the simple idea of [12] that far-away images do not share the same visual content. A GPS information associated with each database image allows us to construct such a negative set for each image j in turn. Details are provided in experimental section 7.

The positive training data set is represented by the only image j itself. Why is that? It comes from the nature of the dataset. One could find additional positive examples by taking adjacent panorama images and building a graph where each node represents an image and each edge represents a visual overlap. However, due to the panorama sampling density during the capturing process of the Google Street View it is very rare to detect a visual overlap between two adjacent panoramas. Instead, very often the detected visual overlap is between the images that come from the same panorama and differ by a homography. We found that there is no benefit from taking these images into account.

6 Why does exemplar-SVM works

The convex objective (5) consist of regularization term a two penalty terms with weights C_1 and C_2. We will show that if $C_2 \to 0$ the resulting w_j is going to be parallel with Φ_j^+, hence $\Psi_j \to \Phi_j^+$.

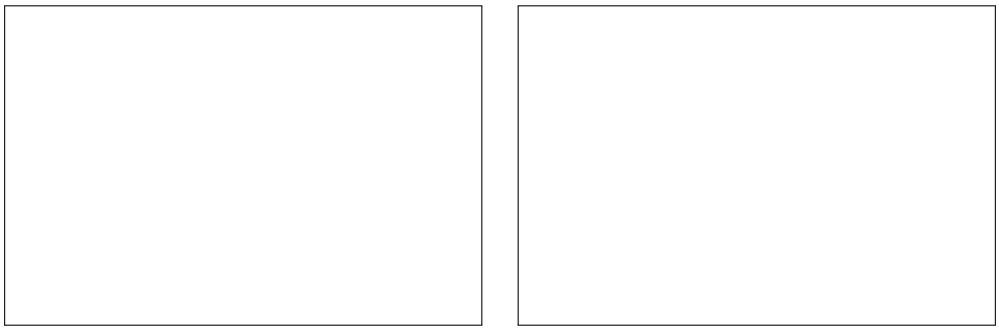

Figure 1: An illustration of the effect of decreasing a parameter C_2. As the paramterer C_2 is decreased the resulting vector w is being parallel to Φ^+.

Lets assume for the moment that the weight of the negative data penalty $C_2 \to 0$. Then the regularization term and the first penalty term dominates the equation (5). Any arbitrary vector w_j can be decomposed into parallel and orthogonal direction w.r.t. Φ_j^+ such that $w_j = w_j^\perp + w_j^{||}$, hecnce the convex objective (5) can be rewritten as

$$||w_j^\perp + w_j^{||}||^2 + C_1 \cdot h((w_j^\perp + w_j^{||})^T \Phi_j^+ + b_j) + C_2 \sum_{\Phi \in \mathcal{N}_j} h(-(w_j^\perp + w_j^{||})^T \Phi - b_j) \qquad (7)$$

If the orthogonal part $w_j^\perp$ was removed the equation (7) becomes

$$||w_j^{||}||^2 + C_1 \cdot h(w_j^{||T} \Phi_j^+ + b_j) + C_2 \sum_{\Phi \in \mathcal{N}_j} h(-w_j^{||T} \Phi - b_j) \qquad (8)$$

Notice that this operation has affected only two terms. The negative data penalty term has changed, however this change was negligible since $C_2 \to 0$. The regularization term always decreases, while the positive data penalty term does not change due to the orthogonality. This means that as we decrease the penalty weight C_2 the optimal w_j^* is close to being parallel with the original Fisher vector Φ_j^+ as shown in Figure 1. The re-normalized w_j^* is therefore almost identical with Φ^+. As we increase the weight C_2 the optimal w_j^* declines from the original FV and this declination depends on (i) distribution of the negative data and (ii) the value of C_2 w.r.t. C_1 Loosely speaking, the parameter C_2 designates how much information about hard negative examples wil be embedded in w_j^*.

As a conclusion, for certain setting of the parameters C_1 and C_2 the learned w_j^* is slightly declined from the original Fisher vector in such a way that it is better separated from the hard negative examples. The details of picking the parameters are given in the following section.

(TODO: In Results show (i) how much it differs e.g. dot product with original FV, (ii) how the performance changes with C2 e.g. x-validation plot or just plot.)

7 Experimental evaluation

(TODO: Show calibrated BOW, why we are not showing calibrated FV, argue for it. Emphasize that the goal was achieved. Highlight its benefits!) In this section we first describe the experimental datasets, then we give implementation details and finally compare performance of the proposed approach on two datasets with several baseline methods.

7.1 Image datasets

(Another image retrieval datasets? INRIA Holliday, Sun, etc., it could be doable, but we have to find a FV competitor)

We performed our experiments on a database of Google Street View images from the Internet. For evaluation we have used two datasets. The first one is identical to [6, 24] ($25k$ of images), the latter one is an extension extension of this dataset to achieve a higher spatial density ($55k$ of images).

We downloaded panoramas from Pittsburgh (U.S.) covering roughly an area of 1.3×1.2 km^2. Similar to [2], for each panorama we generate 12 overlapping perspective views corresponding to two different elevation angles to capture both the street-level scene and the building façades, resulting in a total of 24 perspective views each with 90° FOV and resolution of 960×720 pixels.

As a query set with known ground truth GPS positions, we use the 8999 panoramas from the Google Street View research dataset, which cover approximately the same area, but were captured at a different time, and typically depict the same places from different viewpoints and under different illumination conditions. For each test panorama, we generate perspective images as described above. Finally, we randomly select out of all generated perspective views a subset of 4k images, which is used as a test set to evaluate the performance of the proposed approach.

7.2 Implementation details

SIFT features: For all images in turn SIFT local descriptors [14] are extracted and subsequently these features are converted to a rootSIFT [1]. For extraction we use publicly available library `vlfeat` [**?**].

BOW baseline: A vocabulary of 100k visual words is learned by approximate k-means clustering [20] from a subset of features from 5,000 randomly selected images. A tf-idf [23] vector is computed for each image by assigning each descriptor to the nearest cluster center. Finally, all tf-idf vectors are normalized to have unit L_2 norm.

Fisher vector baseline: We first begin with decorrelation of local descriptors. We train a PCA matrix on a set of descriptors of 5,000 randomly selected database images and we project all root-SIFT descriptors into a $d = 64$ dimensional space. Then we learn a generative GMM that consist of $N = 256$ components. We train this model on a set of decorrelated descriptors from 5,000 randomly selected database images.

It follows by learning a PCA for FV decorrelation. The resulting dimension of full FV is $2Nd = 32,768$. To learn a reasonable non-singular PCA matrix we need at least this amount of data which we do not have for the first dataset which consists of only 25k images. For both databases we use all available data to train the PCA and we underline that for the first $25k$ dataset the projection to $d' > 25k$ is meaningless since the PCA matrix was singular.

Finally, to compute the FV of an image we (i) compute the rootSIFTs and perform the PCA decorrelation to $d = 64$ space, (ii) use the learned GMM and decorrelated descriptors to compute full FV (iii) project the full FV into lower dimensional spaces, namely to 2^β where $\beta = 7 \ldots 14$ and (iv) re-normalize projected FVs by operator (3).

Learning vectors: To learn the linear regressor (**??**) for database image j, the positive and negative training data is constructed as follows. The *negative training set* $\mathcal{N}_j$ is obtained by: (i) finding the set of images with geographical distance greater than $200\mathrm{m}$; (ii) sorting the images by decreasing value of similarity to image j measured by the dot product between their respective Fisher vectors; (iii) taking the top $N = 500$ ranked images as the negative set. In other words, the negative training data consists of the hard negative images, i.e. those that are very similar to image j but are far away from its geographical position, hence, cannot have the same visual content. The *positive training set* $\mathcal{P}_j$ consist of the only image j itself.

For SVM training we use `libsvm` [5] with $L2$ regularizer and $L2$-loss penaty fucntion $h(x)$ from equation (5). To obtain parameters C_1 and C_2 we perform a grid search and evaluate the performance on held out test set. We observe that for different FV target PCA dimensions the parameter C_1 is quite stable (typically $C_1 = 1$) while the optimal parameter for C_2 differs between 10^{-6} to 10^{-1}.

To learn the new image representation from FVs, for each database image j in turn we (i) learn SVM from $\mathcal{P}_j$ and N_j (ii) $L2$ normalize learned w_j (iii) use this vector as the new image representation. At query time we compute a FV Φ_q of the query image and measure its similarity score to each database image by equation (**??**).

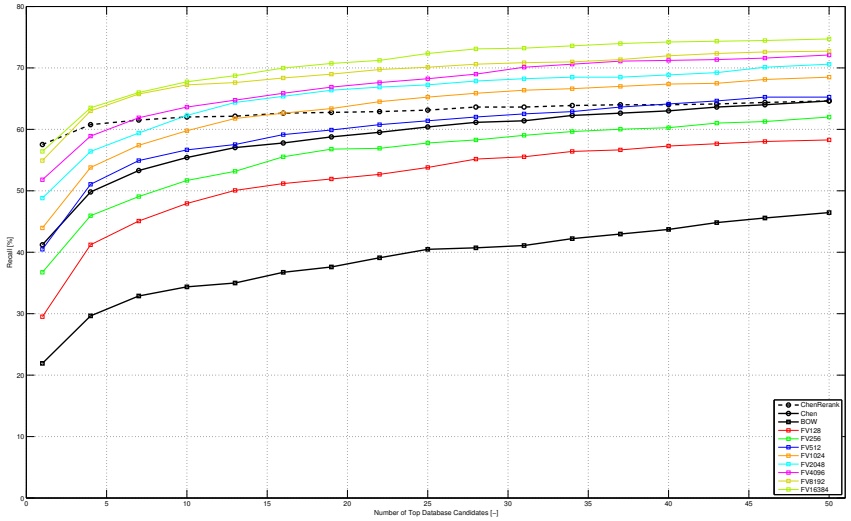

Figure 2: Fake results, actually SF, no embedded FV curves Recall curves for different methods

recall@K [%]	25k Pittsburgh					55k Pittsburgh				
Method:	1	2	5	10	20	1	2	5	10	20
BOW	22	24	31	34	38					
FV256	37	39	46	52	56					
svmFV256	40	42	55	59						
FV1024										
FV2048										
FV4096	etc....									

Table 1: Fake results! The percentage of correctly localized test queries for which the topK ranked database image is within 20 meters from the ground truth query position. The proposed method (svmFV) outperforms the baseline methods.

Ground truth: Since all the query images have associated GPS location we can compute the spatial distance from database image. We consider query image to be correctly localized if its retrieved database image lies within a perimeter of $20m$ from the query image.

7.3 Results

For each database we compare our results to two baselines: a standard BOW baseline and a Fisher vector baseline. We measure our performance by recall@K metric which is the portion of query images that have at least one relevant image inside its ranked shortlist of the length K. We perform our experiments on several target FV dimensions as shown in the figure 2.

It is noticeable that proposed method outperforms all baselines and consistently improves the performance of the original FV over all lengths K of the shortlist and all target FV dimensions. This particularly means that learned descriptors are more likely to attract relevant images into top$K$ shortlist. The results are summarized in table 1.

8 Conclusions

References

[1] Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: IEEE PAMI (2012)

[2] Chen, D., Baatz, G., Köser, Tsai, S., Vedantham, R., Pylvanainen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., Girod, B., Grzeszczuk, R.: City-scale landmark identification on mobile devices. In: CVPR (2011)

[3] Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)

[4] Cummins, M., Newman, P.: Highly scalable appearance-only SLAM - FAB-MAP 2.0. In: Proceedings of Robotics: Science and Systems. Seattle, USA (2009)

[5] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. J. Machine Learning Research **9**, 1871–1874 (2008)

[6] Gronat, P., Obozinski, G., Sivic, J., Pajdla, T.: Learning and calibrating per-location classifiers for visual place recognition. In: CVPR, pp. 907–914. IEEE (2013). URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#GronatOSP13

[7] Horaczek, S.: How many photos are uploaded to the internet every minute? (2014). URL http://www.popphoto.com/news/2013/05/how-many-photos-are-uploaded-to-internet-every-minute

[8] Jegou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR, pp. 1169–1176 (2009)

[9] Jégou, H., Douze, M., Schmid, C.: Product Quantization for Nearest Neighbor Search. IEEE PAMI **33**(1), 117–128 (2011). DOI 10.1109/TPAMI.2010.57. URL http://hal.inria.fr/inria-00514462/en/

[10] Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR 2010 - 23rd IEEE Conference on Computer Vision & Pattern Recognition, pp. 3304–3311. IEEE Computer Society, San Francisco, United States (2010). DOI 10.1109/CVPR.2010.5540039. URL http://hal.inria.fr/inria-00548637

[11] Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(9), 1704–1716 (2012). DOI 10.1109/TPAMI.2011.235. URL http://hal.inria.fr/inria-00633013. QUAERO

[12] Knopp, J., Sivic, J., Pajdla, T.: Avoidng confusing features in place recognition. In: ECCV (2010)

[13] Krapac, J., Verbeek, J., Jurie, F.: Modeling Spatial Layout with Fisher Vectors for Image Categorization. In: ICCV 2011 - International Conference on Computer Vision, pp. 1487–1494. IEEE, Barcelona, Spain (2011). DOI 10.1109/ICCV.2011.6126406. URL http://hal.inria.fr/inria-00612277

[14] Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)

[15] Majdik, A., Albers-Schoenberg, Y., Scaramuzza, D.: Mav urban localization from google street view data. In: IROS, pp. 3979–3986. IEEE (2013). URL http://dblp.uni-trier.de/db/conf/iros/iros2013.html#MajdikAS13

[16] Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV (2011)

[17] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)

[18] Perronnin, F., Dance, C.R.: Fisher kernels on visual vocabularies for image categorization. In: CVPR (2007)

[19] Perronnin, F., Liu, Y., Snchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: CVPR, pp. 3384–3391. IEEE (2010). URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#PerronninLSP10

[20] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)

[21] Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR (2007)

[22] Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A.: Fisher Vector Faces in the Wild. In: British Machine Vision Conference (2013)

[23] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003). URL http://www.robots.ox.ac.uk/~vgg

[24] Torii, A., Sivic, J., Pajdla, T.: Visual localization by linear combination of image descriptors. In: ICCV Workshops, pp. 102–109. IEEE (2011). URL http://dblp.uni-trier.de/db/conf/iccvw/iccvw2011.html#ToriiSP11

[25] Torii, A., Sivic, J., Pajdla, T., Okutomi, M.: Visual place recognition with repetitive structures. In: CVPR (2013)